Context-Aware Question Answering in Urdu

Samreen Kazi and **Shakeel Khoja**
School of Mathematics and Computer Science
Institute of Business Administration (IBA)
Karachi, Pakistan
{sakazi, skhoja}@iba.edu.pk

Abstract

Answer sentence identification and extractive answer span identification are crucial components in the development of robust question-answering (QA) systems. Despite advancements in natural language processing (NLP), there remains a gap in applying these technologies to Urdu due to the scarcity of annotated datasets and linguistic tools. This paper addresses this gap by introducing a three-stage unified framework aimed at improving both tasks.The framework consists of three key components: key sentence identification, extractive answer span identification, and a unified scoring model. For sentence identification, the framework employs a sliding window approach for text alignment, using noun term frequency for relevance scoring and vector similarity from pre-trained word embeddings to capture deeper semantics. For extractive answer span identification, the model uses a fine-tuned multilingual BERT (mBERT) model trained on the Universal Dependencies (UD) Treebank for Urdu to identify noun chunks for linguistic relevance. The unified model integrates probabilities from sentence identification and span extraction to derive a composite score for selecting the most relevant answer span.Experimental results show the proposed approach significantly outperforms traditional methods, demonstrating its potential for broader application in other low-resource languages like Urdu.

1 Introduction

Question Answering (QA) systems are essential tools for extracting precise information from large text corpora in response to user queries (Kazi et al., 2023). Developing such systems for low-resource languages like Urdu is particularly challenging due to the lack of extensive annotated datasets and specialized linguistic tools (Daud et al., 2017). Standard QA approaches, which often rely on syntactic and semantic similarities typical of high-resource languages, struggle to capture the linguistic nuances and rich morphology characteristic of Urdu. This gap highlights the need for methodologies tailored specifically to the unique challenges posed by such languages (Otegi et al., 2020). Answer sentence identification and answer extraction are critical components of QA systems. Answer sentence identification involves identifying sentences based on their likelihood of containing the correct answer, while answer extraction focuses on identifying the specific text segment within these sentences that directly answers the question (Allam and Haggag, 2012). Traditional models for these tasks often fall short as they rely predominantly on surface-level syntactic and semantic similarities, which are insufficient for capturing the complex linguistic features of Urdu (Chang et al., 2024). This paper introduces a comprehensive three-stage unified framework that integrates Key sentence identification, extractive answer extraction to enhance performance for Urdu text. The proposed model employs combination of traditional and advanced text processing techniques to address the challenges posed by the Urdu language. The first stage utilizes a custom-designed weighted sliding window algorithm (Richardson et al., 2013) for precise text alignment, enhancing relevance scoring through the term frequency of nouns. The second stage leverages a fine-tuned multilingual BERT (mBERT) model (Devlin et al., 2018), trained on the Universal Dependencies (UD) Treebank for Urdu (Bhat et al., 2017), to identify noun chunks within the text. These chunks are evaluated and grouped based on semantic similarity, with the best chunk being

selected based on aggregated scores. The final stage combines the probabilities from both the identification and extraction stages into a unified score, ensuring the identification of the most relevant answer chunk from the top-ranked sentences by leveraging both sentence-level and phrase-level evidence. The research contributions of this work are as follows:

1. Development of a three-stage unified framework that integrates key sentence identification and extractive answer span identification, specifically tailored for the Urdu language.

2. Introduction of a customized sliding window algorithm for question-passage alignment, enhancing relevance scoring through the term frequency of nouns.

3. Demonstration of significant performance improvements over traditional methods on Urdu datasets, highlighting the model's potential for broader application in other low-resource languages.

The rest of this paper is structured as follows. Section 2 provides an overview of the relevant background. Section 3 details our methodology, focusing on the stages of answer sentence identification, answer extraction, and the unified model for QA. Section 4 outlines the experimental setup, and Section 5 presents the results, followed by a discussion of their significance.

2 Literature Review

Question Answering (QA) systems have advanced significantly in high-resource languages like English, Chinese, and European languages. Early multi-stage QA methods relied on feature engineering and traditional machine learning. Yao et al. (Yao et al., 2013) used syntactic features and logistic regression for answer ranking, highlighting linguistic structure's role.With deep learning, neural models became prominent. Severyn and Moschitti (Severyn and Moschitti, 2015) introduced a CNN for sentence pair modeling, outperforming previous methods. The advent of transformer models, notably BERT (Devlin et al., 2018), revolutionized QA. Nogueira and Cho (Nogueira and Cho, 2019) fine-tuned BERT for passage ranking, setting new benchmarks.Answer extraction has evolved from rule-based systems like TextRunner (Banko et al., 2007) to neural models. Named Entity Recognition (NER) significantly aids this process, with Lample et al. (Lample et al., 2016) combining LSTMs and CRFs. Span-based extraction models, like SpanBERT (Joshi et al., 2020), further improved extractive QA tasks.End-to-end QA systems like DrQA (Chen et al., 2017) have shown strong performance, supported by datasets like SQuAD (Rajpurkar et al., 2016), which have become standard benchmarks. The introduction of datasets such as Natural Questions (Kwiatkowski et al., 2019) has further pushed open-domain QA research.These advancements have inspired research in low-resource languages like Urdu, Arabic, and Hindi (Kazi and Khoja, 2021) (Arif et al., 2024) (Shaheen and Ezzeldin, 2014) (Gupta et al., 2018). While transformer models like T5 (Raffel et al., 2020) have been adapted, challenges remain in effectively applying these to Urdu due to linguistic nuances and resource constraints. Our work introduces a lightweight, interpretable multi-stage framework leveraging traditional techniques alongside fine-tuned multilingual BERT, addressing Urdu-specific challenges. Although it may not match the accuracy of models like mT5, it provides a foundation for advanced hybrid systems.

3 Methodology

This section presents the two-stage approach used to integrate key sentence identification and extractive answer span identification into a unified learning model, as illustrated in Figure 1. The methodology employs a sliding window technique for measuring text overlap between the passage and the question, incorporates term frequency of nouns for relevance scoring, and computes semantic vector similarity using word embeddings. Additionally, a fine-tuned mBERT model, trained on the UD Treebank for Urdu, is used for high-quality chunk identification. The framework consists of the following three components:

(i) Key sentence identification: A probabilistic model is used to identify sentences in

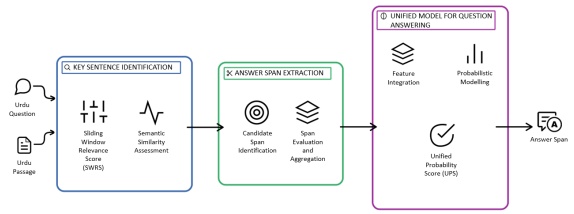

Figure 1: Overall Architecure of Context-Aware QA for Urdu Language

the passage that are most relevant to the question.

(ii) Extractive Answer Span Identification: Another probabilistic model is used to extract answer spans from the identified key sentences.

(iii) Unified model: The outputs from the key sentence identification and answer extraction stages are integrated into a unified model. The probabilities from both task-specific models are combined to improve the overall performance of the system.

3.1 Key Sentence Identification

In this section, we describe the methods used for key sentence identification, which involves determining which sentences are most likely to contain the correct answer to a given question. This process is divided into two main parts: Sliding Window Relevance Score Feature(SWRS) and Semantic Similarity Feature. The SWRS feature identifies the most relevant segment of the passage using a sliding window approach, while the semantic vector similarity measures the similarity between the question and candidate sentences using word embeddings.After extracting both features logistic regression model predicts the probability that each sentence contains an answer.

3.1.1 Sliding Window Relevance Score Feature(SWRS)

The SWRS feature begin by tokenizing the question and passage into individual words using UrduHack (ALi, 2020). This allows for a detailed comparison at the word level. Next, the term frequency (TF) for nouns in the corpus is calculated, as nouns often carry significant meaning in sentences. Using a sliding window approach, the passage is segmented into overlapping windows of a fixed size, and the relevance score for each window is calculated based on the overlap with the question words and the term frequency of nouns within the window. Additionally, a word co-occurrence matrix is used to enhance the relevance scoring by considering the contextual relationships between words. The window with the highest relevance score is then selected as the most aligned segment, providing a focused area of the passage that is most likely to contain the answer , as described in Algorithm 1.

Algorithm 1 Algorithm of SWRS

Abbreviations:
- Q_{text}: Question string
- P_{text}: Passage string
- W: Window size
- s: Step count
- C_m: Co-occurrence matrix
- TF_{noun}: Term Frequency of nouns
- $I_{\text{C}}(n)$: Inverse Count of nouns

Input:
- Q_{text}: Question string
- P_{text}: Passage string
- W: Window size
- s: Step count

Tokenization:
- Split Q_{text} and P_{text} into words
- Output: Q_{tokens}, P_{tokens}

Calculate TF for Nouns:
- Identify nouns in P_{tokens} and calculate TF_{noun}

Calculate Co-occurrence Matrix:
- Compute C_m using P_{tokens} and Q_{tokens}

Overlap Score (O_{s}):
- For each window W in P_{tokens}:
  - $O_{\text{s}} = \sum_{n \in W \cap Q_{\text{tokens}}} TF_{\text{noun}}(n) \times I_{\text{C}}(n)$

Co-occurrence Score (Co_{s}):
- For each window W in P_{tokens}:
  - $Co_{\text{s}} = \sum_{w \in W} \sum_{q \in Q_{\text{tokens}}} C\_m(w, q)$

Relevance Score (R_{s}):
- For each window W in P_{tokens}:
  - $R_{\text{s}} = O_{\text{s}} + Co_{\text{s}}$

Sliding Window:
- Slide W across P_{tokens} with size w and step s.
- Calculate scores and find $j^* = \arg\max_j R_{\text{s}}(j)$

Output:
- $best\_window = W_{j^*}$

Return:
- $best\_window$

3.1.2 Semantic Similarity Features

The initial phase involves utilizing pre-trained word embeddings, specifically fastText embeddings (Bojanowski et al., 2016) trained on a

large corpus of question-answer pairs. FastText embeddings are preferred here as they incorporate subword information, capturing morphological nuances and effectively handling out-of-vocabulary words. Training on a QA-specific corpus ensures that the embeddings are tailored to the domain, enhancing their representation of relevant semantic relationships. Word embeddings map words into a continuous vector space, where semantically similar words are situated closer together. Each word w in the question and candidate sentences is transformed into a high-dimensional vector $\mathbf{v}_w$ that encapsulates its semantic nuances. This transformation captures the contextual meaning of words, facilitating a sophisticated comparison between the question and candidate sentences beyond mere lexical similarity.Subsequently, a single vector representation for the entire question and each candidate sentence is constructed by aggregating the vectors of content words, such as nouns, verbs, and adjectives. This aggregation, achieved through vector summation:

$$\mathbf{V}_{\text{sentence}} = \sum_{w \in \text{content words}} \mathbf{v}_w$$

integrates the semantic information of all content words, resulting in a composite vector that represents the overall meaning of the sentence. This method enhances the capacity to perform meaningful comparisons between the question and the candidate sentences. The final step involves computing semantic similarity by calculating the cosine similarity between the vector representation of the question $\mathbf{V}_Q$ and each candidate sentence $\mathbf{V}_C$:

$$\text{Cosine Similarity} = \frac{\mathbf{V}_Q \cdot \mathbf{V}_C}{\|\mathbf{V}_Q\| \|\mathbf{V}_C\|}$$

3.2 Extractive Answer Span Identification

In this section, we focus on the process of extracting the specific span of the sentence that answers the given question. The extraction process ensures that the most relevant and precise text span is identified and selected, providing an accurate response to the user's query.In the end, another logistic regression model evaluates the likelihood of each candidate span being the correct answer.

3.2.1 Candidate span extraction

In the answer extraction process, the initial step involves identifying candidate spans within the sentences that are likely to contain the correct answer. To achieve this, we utilize a fine-tuned mBERT (multilingual BERT) model, specifically trained on the Universal Dependencies (UD) Treebank for Urdu. This model is proficient in identifying high-quality noun phrases and other relevant text segments, ensuring that the candidate span are both linguistically coherent and contextually appropriate for further evaluation. Let C represent the set of candidate span identified as:

$$C = \{c_1, c_2, \ldots, c_n\}$$

where each c_i represents an individual candidate span. Once the candidate spans are identified, the next step is to evaluate the quality of each span based on several features. These features include the length of the span, its position within the sentence, and its relevance to the question posed. The evaluation process can be represented by scoring each span c_i as follows:

$$S(c_i) = \alpha \cdot \text{len}(c_i) + \beta \cdot \text{pos}(c_i) + \gamma \cdot \text{rel}(c_i, Q)$$

where α, β, γ are weighting factors that adjust the importance of each feature, and Q is the vector representation of the question. This scoring helps determine the likelihood that a given span contains the correct answer, allowing us to filter and retain only the most promising candidates for further consideration. After evaluating the individual span, the subsequent step is to group semantically equivalent span. This grouping is based on both word overlap and semantic similarity, ensuring that span conveying the same or similar information are clustered together. Let G represent these groups:

$$G = \{g_1, g_2, \ldots, g_m\}$$

where each group g_j contains semantically similar chunks. Semantic similarity between span c_i and c_j can be computed using:

$$\text{sim}(c_i, c_j) = \frac{\mathbf{v}_{c_i} \cdot \mathbf{v}_{c_j}}{\|\mathbf{v}_{c_i}\| \|\mathbf{v}_{c_j}\|}$$

where $\mathbf{v}_{c_i}$ and $\mathbf{v}_{c_j}$ are the vector representations of the spans. This step ensures that

we consolidate the information across similar span, which aids in aggregating their scores.The final step in the span extraction process involves selecting the best span from each group of equivalent span. This is achieved by aggregating the scores within each group and selecting the span with the highest score:

$$c^* = \arg\max_{c_i \in g_j} S(c_i)$$

This selection process ensures that the chosen span not only aligns well with the question but also represents the most reliable and precise part of the text. By considering aggregated scores, we enhance the robustness of our selection, ensuring the extracted answer is both relevant and accurate. This systematic approach, encompassing candidate span identification, span evaluation, grouping of equivalent chunks, and the final selection of the best span, guarantees that the extracted span is contextually appropriate and precise. This enhances the overall effectiveness of the question-answering system by ensuring that semantically similar sentences, even without shared lexical items with the question, are considered relevant, thereby significantly improving the accuracy of the answer retrieval.

3.3 Unified Model for Question Answering

In this section, we introduce the methodology for combining probabilities derived from the key sentence identification and span extraction processes. The objective is to unify these probabilities into a single score that can identify the most relevant answer span from the top-ranked sentences. By integrating both identification and extraction stages, we ensure that the selected span is contextually appropriate and precisely extracted. Our unified model leverages advanced feature engineering and probabilistic modeling to enhance the accuracy and relevance of extracted answers from textual data. This sophisticated approach combines the strengths of key sentence identification and extractive answer span identification, tailored specifically to the nuances of different question types in Urdu. Here's an overview of the methodology:

3.3.1 Feature Extraction

The model leverages features extracted from various modules, including those specifically designed for key sentence identification and extractive answer span identification. Additionally, it incorporates a diverse set of features tailored to capture both the lexical and semantic nuances of the text, further enhancing its ability to identify the most relevant answer:

Question Type Specific Features:

- **Question-word Features**: Extracts and utilizes the POS, DEP, and NER tags of the main question word (e.g., کیا, کون/کس), appending these to the question type to refine feature sensitivity.
- **Question Focus**: Determines the focus noun phrase within the question, crucial for aligning the model's attention to the most relevant part of the quer.

Query Ques:

Pairs the headword and question focus features, creating compound indicators such as question-type|question-focus-word|headword-pos-tag.

Span Tags:

checking for the presence of significant noun phrase that match expected answers based on the question type.

3.3.2 Conditional Probability Learning

During training, the model learns the conditional probabilities $P(c \mid s, f)$:

- c: Candidate answer span.
- s: Sentence containing the answer span.
- f: Feature vector encompassing all extracted features for the answer span and the sentence.

This probabilistic framework captures the complex interdependencies between sentence relevance, answer span quality, and the rich contextual features derived from the text.

3.3.3 Unified Probability Score

In the prediction stage, the model calculates a unified probability score(UPS) for each candidate answer span:

$$\text{UPS}(c) = P(c \mid f) \times P(s \mid c, f)$$

where $P(c \mid f)$ represents the probability of the answer span being correct, based solely on its features, and $P(s \mid c, f)$ assesses the conditional probability that the sentence is key, given the span and its features.

3.3.4 Final Answer Selection

The model selects the candidate answer span with the highest unified probability score. This selection process prioritizes spans that are not only plausible based on their intrinsic features but also originate from sentences that are contextually aligned with the question. This dual consideration ensures that the chosen answers are both accurate and contextually relevant, thereby significantly enhancing the performance of the question-answering system.

4 Experiments

4.1 Data

For the experimental validation of our unified model, we employed two significant datasets tailored for Urdu question-answering systems: UQuAD and UQA. These datasets are selected and adapted to rigorously test both the answer sentence identification and extraction capabilities of our model. The Urdu Question Answering Dataset (UQuAD1.0) includes 46,481 Stanford Question Answering Dataset(SQuAD 2.0) (Rajpurkar et al., 2016) questions translated using google translation API covering various domains such as history, science, and general knowledge. It also contains 4000 crowdsource question annotated by humans based on Question types. The UQA corpus on the other hand features 136,211 questions, focusing on domain-specific topics, created using the "Enclose to Anchor, Translate, Seek" (EATS) technique from the Stanford Question Answering Dataset (SQuAD 2.0). This technique ensures that answer spans are preserved in the translated context paragraphs, making it suitable for training and evaluating Urdu QA models. It consists of 83,018 answerable and 41,727 unanswerable questions, providing a balanced setup for models to not only retrieve accurate answers but also to discern when no plausible answer is present in the text.Table**??**shows distrbution of dataset.

Dataset	QA Pairs	Question Types	EM
UQuAD (MT)	45,000	No	0.66
UQuAD (CS)	4,000	Yes	0.50
UQA (MT)	124,745	No	0.85

Table 1: Distribution of UQA and UQuAD Datasets.
'MT' = Machine Translation, 'CS' = Crowd-Sourced.

4.1.1 Dataset Adaptation for Sentence identification Evaluation:

To assess the identification capabilities of our model adequately, we adapted both UQuAD and UQA by an approach that includes:

1. Extraction of Candidate Answer Sentences: We analyzed each paragraph within the datasets to identify all sentences that could potentially contain the answer, based on their content overlap with the gold-standard answer provided (Charras and Lecroq, 2004).

2. Annotation of Candidate Sentences: Each identified sentence was subsequently labeled as either '1' (containing the answer) or '0' (not containing the answer). This binary annotation serves as the definitive ground truth for the key sentence identification task.

The adapted dataset was divided into training, and test sets as shown in Table 2, ensuring that no question-paragraph pair appeared in multiple subsets.

Dataset	Train	Test
UQuAD(MT)	36,000	9,000
UQA	99,796	24,949

Table 2: Train/Test Split for Training Model

4.1.2 Evaluation Metrics

We adhere to standard evaluation procedures and metrics for QA rankers as outlined in prior research (Rajpurkar et al., 2016). Our evaluation metrics for assessing the performance of question answering systems include:

- **Exact Match (EM)**: Measures the percentage of predictions that exactly match any one of the ground truth answers.

- **F1 Score**: Computes the harmonic mean of precision and recall at the individual token level, considering both the partial correctness of the answers.
- **Average Precision at K** : Defined as the average of correct answer sentences within the top K results to evaluate Key sentence identification.

4.2 Baseline Models for Comparison

4.2.1 Word N-grams - Sliding Window Baseline

To establish a comparative baseline, we used the Word N-gram overlap method, a traditional technique used to determine textual similarity (Richardson et al., 2013). This method involves segmenting texts into fixed-length N-grams and calculating similarity scores based on the overlap of these N-grams. This approach has been validated in various applications such as plagiarism detection and text reuse (Daud et al., 2017). For our purposes, we adapt it to extract answer spans by tokenizing the text into N-grams and selecting spans based on their overlap with the query, calculated as follows:

$$\text{overlap} = \frac{|S(P_1, n) \cap S(P_2, n)|}{\min(|S(P_1, n)|, |S(P_2, n)|)} \quad (1)$$

4.2.2 TF-IDF - Feature-Based Baseline

Additionally, we employ the traditional TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique, which represents text using term frequency-inverse document frequency metrics. This method is enhanced with N-gram frequencies ranging from unigrams to trigrams to capture local word order, crucial for understanding contextual relevance. The TF-IDF value for a term t in a document d within a document set D is calculated as follows:

$$\text{TF-IDF}(t, d, D) = \text{TF}(t, d) \times \text{IDF}(t, D) \quad (2)$$

where:

- $\text{TF}(t, d)$ is the term frequency of term t in document d, and
- $\text{IDF}(t, D)$ is the inverse document frequency of term t across the document set D, defined as:

$$\text{IDF}(t, D) = \log\left(\frac{|D|}{1 + |\{d \in D : t \in d\}|}\right) \quad (3)$$

For N-grams, the terms t are extended to include unigrams, bigrams, and trigrams, thereby enhancing the textual representation by capturing contiguous sequences of up to three words. This enhancement allows for a more nuanced understanding of the text's structure and semantics.

5 Results and Discussion

We employed various baseline approaches as mentioned in section 4.2, which include traditional text representation techniques to retrieve answer.In evaluating the sentence identification capabilities of our model, we observed differentiated performance across various question types, as shown in Table 3. The model exhibited high precision in identifying sentences relevant to 'Who', 'When', and 'Where' questions, achieving AP@K scores of 0.55, 0.58, and 0.68, respectively. These question types typically involve extracting specific entities or temporal and locational information, which are well-captured by our model's feature set. Conversely, 'What' and 'Why' questions, which often require understanding broader contexts or causal relationships, posed greater challenges, reflected in lower AP@K scores of 0.35 and 0.40. 'How' questions, diverse in their structure and intent, showed moderate performance with an AP@K score of 0.44. Overall, the model achieved an average precision across all question types of 0.44, indicating a robust capability to identify relevant answer-containing sentences but also highlighting potential areas for enhancement in handling complex question contexts and reason behind lower accuracy of unified model shown in tables 5 and 4. overall our unified model achieved better results in answer extraction as shown in 5 and 4 showcasing the effectiveness of our unified model compared to the traditional approaches.

5.1 Discussion

The results indicate that our unified Model significantly outperforms the baseline models,

Question Type	Average Precision at K
What (کیا)	0.35
Who (کون)	0.55
When (کب)	0.58
Where (کہاں)	0.68
Why (کیوں)	0.40
How (کیسے)	0.40

Table 3: Performance of the sentence identification model across various question types using Average Precision at K metric.

Dataset	N-gram	TF-IDF	Unified Model
UQuAD	0.15	0.25	0.48
UQA	0.20	0.22	0.55

Table 4: F1: Performance comparison of different models on UQuAD and UQA datasets.

Dataset	N-gram	TF-IDF	Unified Model
UQuAD	0.12	0.28	0.60
UQA	0.10	0.31	0.50

Table 5: EM: Performance comparison of different models on UQuAD and UQA datasets.

demonstrating its efficacy in leveraging complex feature interdependencies to accurately identify and extract answers. This superior performance underscores the advantage of integrating sentence identification with extraction capabilities in a unified model, particularly in the nuanced context of Urdu language question answering. Our approach to integrating sentence identification and span extraction through unified probabilistic modeling has demonstrated promising results. For example, in the UQuAD and UQA datasets, we observed marked improvements in precision over traditional models, as evidenced by the scores illustrated in our performance tables. To better understand the nuances of the model's performance, let's consider practical examples using Urdu question-answer pairs. Imagine a question in Urdu like "محمد علی نے ہونڈا میں کتنے سال کام کیا؟" (How many years did Mohammad Ali work at Honda?). Our model might identify a sentence such as "محمد علی نے ہونڈا میں چالیس سال تک کام کیا۔" (Mohammad Ali worked at Honda for 40 years), scoring it highly due to the direct match of numeric and contextual information. Conversely, sentences without direct numerical answers or only peripheral relevance to Honda and Mohammad Ali would receive significantly lower scores. This method effectively discerns the relevance and specificity of candidate answer sentences. However, when evaluating our system against state-of-the-art transformer-based models, such as those employing BERT or its derivatives, we notice a gap in achieving top-tier performance metrics like Exact Match (EM) and F1. This discrepancy can largely be attributed to the inherent limitations of N-gram and TF-IDF models in capturing the deep semantic structures that transformer models excel at.

Limitations

This study offers valuable insights into applying NLP techniques for Urdu language processing, but it does face limitations. The primary datasets used, UQuAD and UQA, while comprehensive, do not entirely capture the full diversity of Urdu language use due to synthetic nature of data. Additionally, this model focus mainly on syntactic and semantic features and do not extensively address other linguistic elements such as pragmatics and discourse context, which are vital for fully understanding complex questions. Furthermore, despite showing promising results in Urdu, the model's effectiveness in other low-resource or morphologically rich languages have not been explored. This may limit its broader applicability and scalability, especially in contexts where transformer-based models have shown superior performance.

Ethics Statement

This research adheres to the highest ethical standards. All datasets, including text and question-answer pairs, were sourced from publicly accessible repositories. We ensured that no private or sensitive data was utilized without explicit consent. All sources have been meticulously cited, and the use of any copyrighted material complies strictly with applicable legal standards, ensuring transparency and integrity in our research methodology.

References

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

Ali Mohamed Nabil Allam and Mohamed Hassan Haggag. 2012. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, 2(3).

Samee Arif, Sualeha Farid, Awais Athar, and Agha Ali Raza. 2024. Uqa: Corpus for urdu question answering. *arXiv preprint arXiv:2405.01458*.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2670–2676.

Riyaz Ahmad Bhat, Rajesh Bhatt, Annahita Farudi, Prescott Klassen, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Misra Sharma, Ashwini Vaidya, Sri Ramagurumurthy Vishnu, et al. 2017. The hindi/urdu treebank project. *Handbook of linguistic annotation*, pages 659–697.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Christian Charras and Thierry Lecroq. 2004. Handbook of exact string matching algorithms.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Deepak Gupta, Surabhi Kumari, Asif Ekbal, and Pushpak Bhattacharyya. 2018. Mmqa: A multi-domain multi-lingual question-answering framework for english and hindi. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

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.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Arantxa Otegi, Aitor Agirre, Jon Ander Campos, Aitor Soroa, and Eneko Agirre. 2020. Conversational question answering in low resource scenarios: A dataset and case study for basque. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 436–442.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 193–203.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings*

of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 373–382.

Mohamed Shaheen and Ahmed Magdy Ezzeldin. 2014. Arabic question answering: systems, resources, tools, and future trends. *Arabian Journal for Science and Engineering*, 39:4541–4564.

Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 858–867.