

Article

Generalised Cross-Dialectal Arabic Question Answering Through Adaptive Code-Mixed Data Augmentation

Maha Jarallah Althobaiti 

Department of Computer Science, College of Computers and Information Technology, Taif University,
Taif 21944, Saudi Arabia; maha.j@tu.edu.sa

Abstract

Modern Standard Arabic (MSA) and the many regional dialects differ substantially in vocabulary, morphology, and pragmatic usage. Most available annotated resources are in MSA, and zero-shot transfer from MSA to dialectal tasks suffers a large performance drop. This paper addresses generalised cross-dialectal Arabic question answering (QA), where the context and the question are written in different Arabic varieties. We propose a training-free augmentation framework that generates code-mixed questions to bridge lexical gaps across Arabic varieties. The method produces semantically faithful, balanced code-mixed questions through the following two-stage procedure: lexicon-based partial substitution with semantic similarity and substitution-rate constraints, followed by fallback neural machine translation with word-level alignment when needed. We also introduce automated multidialectal lexicon construction using machine translation, embedding-based alignment, and semantic checks. We carry out our evaluation in a zero-shot setting, where the model is fine-tuned only on MSA and then tested on dialectal inputs using ArDQA, covering five Arabic varieties and three domains (SQuAD, Vlogs, and Narratives). Experiments show consistent improvements under context-question dialect mismatch as follows: +1.09 F1/+0.87 EM on SQuAD, +1.54/+1.25 on Vlogs, and +2.75/+2.27 on Narratives, with the largest gains for Maghrebi questions in Narratives (+12.13 F1/+8.45 EM). These results show that our method improves zero-shot cross-dialectal transfer without fine-tuning or retraining.

Keywords: Arabic question answering; Arabic dialects; cross-dialectal transfer; embedding-based alignment; generalised zero-shot transfer; MSA; neural machine translation; semantic validation; word-level alignment



Academic Editor: Rodolfo Delmonte

Received: 24 December 2025

Revised: 26 January 2026

Accepted: 27 January 2026

Published: 1 February 2026

Copyright: © 2026 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Arabic comprises multiple varieties, including Modern Standard Arabic (MSA) and numerous regional dialects that differ in phonology, morphology, vocabulary, and pragmatics [1–3]. In contrast to Arabic dialects, MSA has received considerable attention in natural language processing (NLP). Several labelled resources and large language models (LLMs) developed for Arabic primarily target MSA texts [4–7]. While MSA dominates most labelled resources, downstream applications frequently operate on dialectal Arabic inputs [8–11]. As most annotated resources are in MSA, models trained on MSA often generalise poorly to dialectal inputs [12,13]. Therefore, robust cross-dialectal Arabic understanding remains a central challenge for Arabic NLP. Moreover, zero-shot transfer from MSA to dialectal tasks typically suffers a substantial performance drop [8,12,14]. Recent work on cross-dialectal translation has also shown that large language models struggle with dialects that lack

publicly available data [15]. These findings highlight the need for methods that bridge lexical and semantic gaps between Arabic varieties in various NLP tasks.

Extractive question answering (QA) models identify and extract exact answer spans from a given context. The context may take the form of a text passage, a table, or even an HTML page [16–18]. In generalised cross-dialectal QA, where the context is written in one Arabic variety while the question is posed in another variety, such mismatches introduce lexical and semantic gaps that degrade answer span extraction. Models trained on MSA often misinterpret dialectal questions and frequently fail to align them with the context [12]. A practical challenge is that dialect-specific QA training data are often unavailable, especially for lower-resource varieties, making additional supervised adaptation infeasible. In such settings, an attractive alternative is to reduce mismatch at test time by rewriting the input question to better match the context variety without retraining the QA model. However, naive rewriting risks changing the question meaning and can introduce incorrect lexical cues. This motivates a controlled code-mixing strategy that injects limited context-dialect signal while explicitly preserving semantic fidelity. To address this setting, we propose a test time code-mixed augmentation method that rewrites questions into balanced code-mixed forms closer to the context variety, while preserving semantic fidelity. Our key idea is to inject only the minimal context-dialect signal required for alignment, using controlled lexical substitution with explicit fidelity constraints. We can summarise our contributions as follows:

- We proposed a novel two-stage code-mixed augmentation framework. It first attempts lexicon-based partial substitution with balanced substitution and semantic similarity constraints, and it then falls back to neural machine with embedding-based word alignment when needed.
- We introduce an automated pipeline for building multidialectal Arabic lexicons using forward/back translation, embedding-based alignment, and SBERT-based fidelity/consistency checks.
- We provide a comprehensive evaluation of generalised zero-shot cross-dialectal transfer in Arabic QA by applying our proposed code-mixed data augmentation method and comparing it with an MSA-only QA baseline on ArDQA [12], a parallel QA dataset that covers five Arabic varieties and three domains (SQuAD, Vlogs, and Narratives)

The remainder of this paper is organised as follows. Section 2 reviews prior work on Arabic extractive QA and code-mixed data augmentation, and Section 3 presents our two-stage augmentation framework and lexicon construction pipeline. Sections 4 and 5 describe the experimental setup and results, while Section 6 analyses findings and errors before we conclude in Section 7.

2. Related Work

2.1. Arabic Extractive QA Models

The field of Arabic extractive QA has undergone significant transformation over the past decade. It has evolved from classical rule-based approaches to advanced neural architectures. This section provides a comprehensive overview of the progress in Arabic extractive QA systems.

The early development of Arabic extractive QA systems relied primarily on rule-based methods and traditional information retrieval techniques. These systems depended on retrieval strategies and predefined linguistic rules to perform key tasks such as question analysis, passage retrieval, and answer extraction [19,20]. A key characteristic of these early systems was their focus on factoid questions. They were typically evaluated on small test sets developed by individual research groups, without standardised benchmarks.

Additionally, these systems were limited by their inability to process complex linguistic phenomena and their reliance on handcrafted rules [19–21]. In [22], the authors provided one of the first comprehensive surveys of Arabic question answering systems.

The transition from classical to neural approaches began with the adoption of attention-based models. The Bidirectional Attention Flow (BiDAF) [23] was among the first neural architectures applied to Arabic extractive QA [24]. The BiDAF architecture followed a hierarchical, multi-stage process consisting of character embedding, word embedding, contextual embedding, attention flow, modelling, and output layers. In [24], the BiDAF model achieved 32% exact match (EM) and 32% F1 on the AQAD dataset.

Sequence-to-sequence (Seq2Seq) models were also explored as baseline approaches [25]. The experiments evaluated different recurrent neural networks such as long short-term memory (LSTM) and gated recurrent unit (GRU) architectures. However, these models consistently underperformed compared to attention-based architectures such as BiDAF. Seq2Seq models exhibited a substantial performance gap relative to human performance, with human annotators achieving 42% EM on the ASER dataset [25], while the best Seq2Seq model achieved only 3.6% EM. In contrast, BiDAF nearly matched human performance at 38% EM.

The emergence of transformer-based models marked a major shift in Arabic extractive QA. The Bidirectional Encoder Representations from Transformers (BERT) architecture emerged as the dominant framework, achieving state-of-the-art performance across multiple Arabic extractive QA datasets [16,26–28]. The development of pretrained models specifically designed for Arabic significantly advanced the field. AraBERT [4], developed for Arabic text processing, became a foundational architecture for Arabic extractive QA in numerous studies [6,24,28–30]. For example, AraBERTv0.2-large achieved the best performance on ArQuAD [31], with an EM of 68.95% and an F1 of 87.15%. Nevertheless, this performance remained below human levels (86.0% EM and 95.5% F1). Several comparative studies have evaluated different Arabic transformer-based models across various extractive QA datasets. In [32], AraBERTv1 achieved the highest overall performance, with EM and F1 scores of 76.50% and 81.77%, respectively, on the Arabic WikiReading test set. AraBERTv1 outperformed multilingual BERT due to its domain-specific pretraining on a large Arabic corpus. This dedicated training allowed the model to capture the linguistic nuances of Arabic more effectively. Recent developments in Arabic NLP have leveraged more advanced pretraining strategies. AraELECTRA [33], based on the ELECTRA pretraining approach [34], replaces the traditional masked language modelling objective with *replaced token detection*. Rather than predicting masked tokens, the model learns to distinguish between original and artificially replaced tokens in the input sequence. This strategy allows the model to learn from all tokens instead of only the masked ones, leading to more efficient and effective training. AraELECTRA has been successfully applied to Arabic QA tasks, achieving competitive performance [31,35].

2.2. Code-Mixed Augmentation for NLP

Code-mixing refers to switching from one language to another within a single utterance or within the same spoken or written text [36]. It presents distinct difficulties for various NLP tasks including QA [37–39]. For example, Chandu et al. [37] created a crowd-sourced factoid QA dataset in three mixed-language pairs (Hindi–English “Hinglish”, Telugu–English “Tenglish”, and Tamil–English “Tamlish”), resulting in, 1694 Hinglish, 2848 Tamlish, and 1391 Tenglish QA pairs. Although this manual construction yields naturally code-switched questions, it is highly labour-intensive. This has motivated a range of automatic code-mixed data augmentation methods to enrich training data for numerous languages, including those that are low-resource and morphologically rich [40–42].

One line of research uses lexical substitution and hand-crafted linguistic rules to produce mixed-language text. Sabty et al. [43] introduced a straightforward Arabic–English code-switching augmentation method for Arabic named entity recognition, achieving a modest F-score gain of 1.51% [42]. Pratapa et al. [44] leveraged linguistic theory (e.g., constraints on where switches may occur) to generate code-mixed sentences from parallel English–Spanish corpora.

More recent work explores learning-based code-mixing augmentation, often to improve cross-lingual transfer. Qin et al. [45] proposed CoSDA-ML, a framework that fine-tunes a multilingual BERT on dynamically generated code-switched data across many languages. By mixing languages within sentences (without relying on parallel corpora), this approach encourages the model to align representations across languages. CoSDA-ML achieved significant gains on zero-shot cross-lingual tasks including sentiment classification and natural language inference (NLI). Similarly, Khanuja et al. [46] showed that fine-tuning multilingual models on synthetic code-mixed examples yields the best performance on code-switched QA and NLI in their GLUECoS benchmark, outperforming both zero-shot and monolingual-trained baselines. These studies demonstrate that introducing code-mixed training instances can substantially boost cross-lingual understanding in NLP systems.

Another direction is generating code-mixed text using model-based algorithms. Hamed et al. [41] investigated Arabic–English code-switching data augmentation through lexical substitutions. They contrasted random switching strategies with learned switching strategies. Their results show that using a seq2seq model to predict optimal switch points yields more natural mixed sentences than random token replacement, as verified by human evaluations. Incorporating these synthetic code-switched samples into training data delivers substantial downstream improvements, including a 34% decrease in language-model perplexity, gains of +4.0–5.1 BLEU points for machine translation, and a 5.2% relative reduction in word-error rate (WER) for speech recognition.

Differences with Existing Studies

Despite significant progress, Arabic extractive QA continues to face several challenges. The scarcity of large-scale annotated datasets suitable for low-resource settings remains a key limitation. The coexistence of multiple Arabic varieties, including Classical Arabic, MSA, and various dialects, adds further complexity to model development. Moreover, the rich morphological structure of Arabic continues to provide difficulties for neural models, particularly when handling dialectal or noisy text.

Existing cross-dialectal Arabic QA work is limited. Our investigation revealed that only a few studies [12,47] have investigated cross-dialectal Arabic QA. However, the TyDiQA-WANA dataset, in [47], includes four Arabic dialects (Algerian, Egyptian, Iraqi, and Jordanian) but deliberately pairs these dialectal questions with passages from Arabic Wikipedia, which are written in MSA. Therefore, it only supports “dialect-to-MSA” scenarios and cannot investigate the more general case where both the question and context come from different dialects. The focus of the study was on the situations where users may pose questions in dialectal Arabic while seeking information from MSA sources. The constructed dataset benchmark cannot be utilised effectively for investigating cross-dialectal Arabic QA, where contexts can be in other Arabic varieties, not only MSA. The recently proposed ArDQA benchmark [12] provides the first parallel QA dataset across five Arabic varieties (Egyptian, Gulf, Levantine, Maghrebi, and MSA) and three domains (SQuAD-style text, vlogs, and narratives). Native speakers translated contexts and questions into each variety and aligned answer spans. To establish baselines, the author of [12] fine-tuned three MSA-trained transformers (AraELECTRA, CAMeLBERT-MSA, and AraBERT) on an MSA QA

corpus and then tested them in zero-shot and generalised cross-dialectal settings. Their results revealed substantial performance drops, especially when the context and question belong to different dialects. Therefore, the study highlights the need for methods that bridge lexical and semantic gaps across various Arabic varieties.

Our work differs from both of these studies in two key ways. First, we handle the fully generalised cross-dialectal scenario, where both the question and context may belong to any pair of Arabic varieties, not just dialect-to-MSA. Second, and most importantly, we propose a code-mixed data-augmentation framework that actively bridges lexical and semantic gaps between dialects to enhance the generalisation in zero-shot transfer. It automatically constructs multidialect lexicons and injects controlled amounts of context-dialect vocabulary into questions, validated via semantic fidelity checks and using neural machine translation and embedding-based embeddings when needed. Our method yields balanced code-mixed questions that align more closely with the target context.

3. Methodology

This section describes our proposed method for enhancing generalised cross-dialectal transfer performance in Arabic question answering. Specifically, our approach addresses scenarios where the context is in one Arabic variety, while the question is in a different variety. Generally, the approach relies on partially substituting words in the question with their counterparts in the Arabic dialect of the context. It systematically augments Arabic questions by generating diverse high-quality code-mixed questions. The approach helps bridge the lexical and semantic gaps between Arabic dialects, facilitating better alignment between questions and context. To this end, it integrates the following two primary methods: lexicon-based partial substitution, and neural machine translation with word-level alignment in an adaptive workflow. Figure 1 shows the general framework of the proposed method.

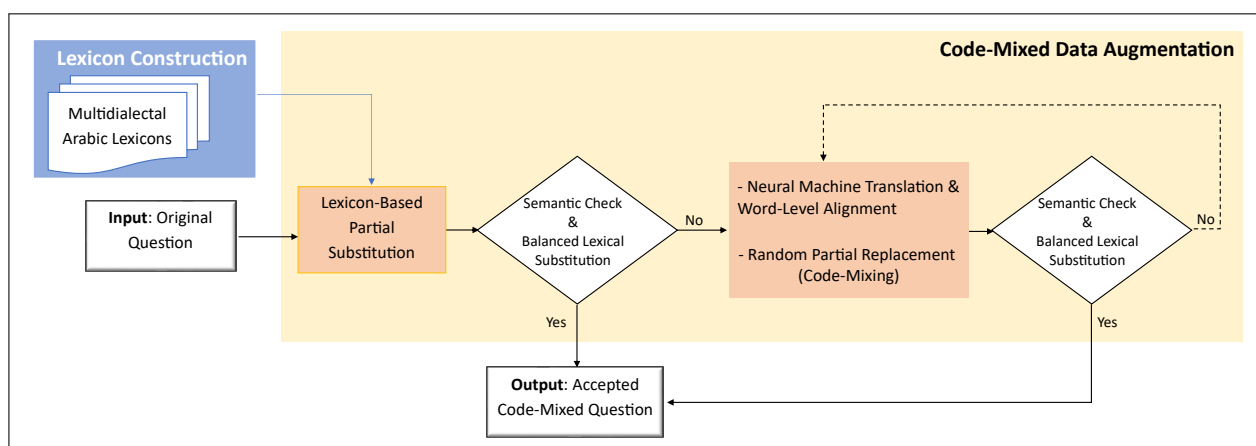


Figure 1. General framework of the proposed method.

Initially, multidialectal Arabic lexicons are developed, and then these lexicons are used to perform lexicon-based partial substitution on the original Arabic questions. Following this, the method employs semantic validation measures to assess how well the modified questions maintain their original meaning and remain logically coherent and understandable after modification. In addition, a balanced-substitution constraint is enforced, requiring that the proportion of replaced tokens remain between 20% and 60%. Candidates with fewer than 20% substitutions are too close to the original question to provide meaningful additional lexical overlap with the context, whereas candidates with more than 60% substitutions tend to behave more like full translations and are more vulnerable to lexicon and alignment noise. Therefore, the 20–60% band aims to preserve enough of the original

wording of the question variety while injecting enough tokens from the context variety to be useful at test time. In SQuAD-style domains, where the context variety is MSA and the question variety is already very close to MSA, the rewritten question may occasionally become almost fully MSA-like. We consider this acceptable, since the primary goal is to improve lexical alignment between the question and the MSA context variety rather than to enforce a fixed degree of mixing in every individual question. Questions that meet both criteria are accepted as final code-mixed questions. If a question fails the validation, the lexicon-substituted version is discarded and the original question is passed to the neural machine translation with word-level alignment phase. Further details regarding each phase are provided in the subsections below.

Although our proposed code-mixed data augmentation method was developed to improve generalised cross-dialectal transfer in Arabic question answering, the procedure itself is inherently task-agnostic. Its ability to automatically generate balanced code-mixed versions of any text dataset makes it adaptable for diverse applications within NLP.

3.1. Lexicon Construction

This section explains the novel approach used in order to create an initial multidialectal Arabic lexicon between five Arabic varieties, namely, Egyptian, Gulf, Levantine, Maghrebi, and MSA. The proposed method involves a systematic and automated pipeline that integrates various translation models, embedding-based alignments, and semantic validation procedures to ensure the reliability of lexical mappings between MSA and various Arabic dialects. Figure 2 summarises the steps followed to create the lexicons.

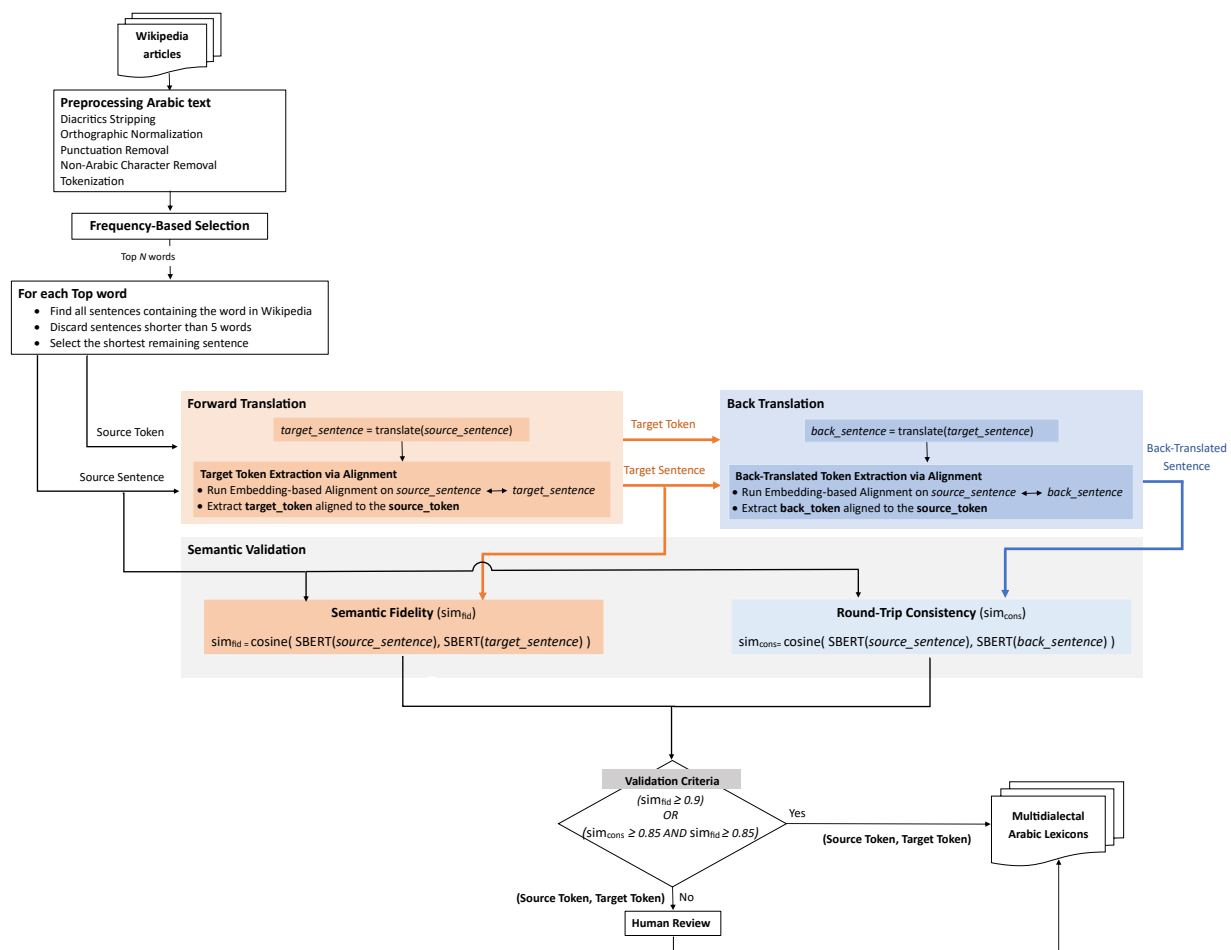


Figure 2. The Process of creating multidialectal Arabic lexicons.

MSA serves as the central Arabic variety, used to gather initial words which are then translated into four other Arabic dialects. Wikipedia serves as the source for collecting these words due to its broad coverage of diverse topics and domains. The process of creating the initial lexicon was conducted on the 20 March 2025 Arabic version of the Wikipedia dump (<https://dumps.wikimedia.org/arwiki/>, accessed on 10 August 2025). Then, we use WikiExtractor [48] in order to discard markup/templates and extract plain text. Arabic text extracted from Wikipedia articles undergoes preprocessing steps, including diacritics stripping which involves removing the short vowel marks. The preprocessing steps also include orthographic normalisation where Arabic letters written in interchangeable orthographic forms are converted into a single consistent form. For example, converting all forms of Alif to a simple bare Alif. The aforementioned two preprocessing steps are important to standardise the text and reduce sparsity in the vocabulary. Then, we remove punctuation and non-Arabic characters. Finally, we apply simple tokenisation, the process of splitting the text into individual words based on whitespaces.

Following preprocessing, tokens undergo frequency-based selection to ensure lexicon construction is initially based on common lexical tokens and their associated sentences. Specifically, all tokens resulted from the preprocessed text are ranked based on their frequency of occurrence and only the top 500 most frequent tokens are selected. This initial list serves as a foundational resource. However, later, during the process of generating code-mixed augmented questions, the lexicon dynamically expands as additional tokens within contexts are identified, thus progressively enhancing the lexicon's coverage.

For each identified top word, we select its associated sentence from the Wikipedia corpus by adopting the following steps:

- Gathering candidate sentences: All sentences containing the identified top word within the corpus are initially collected.
- Filtering candidate sentences: Short sentences (fewer than five words) are discarded. Short sentences usually lack sufficient context for accurate translation and alignment.
- Selecting optimal sentence: From the remaining candidate sentences, the shortest sentence is chosen. This choice provides enough context for accurate machine translation without introducing unnecessary complexity or ambiguity.

The aforementioned steps ensure that each word selected for the lexicon is associated with a specific example sentence. This helps in accurate translation and semantic verification later on. That is, the translation process begins with two key inputs for each lexical entry as follows:

- A source token, representing the word for which dialectal Arabic mapping is desired.
- A corresponding source sentence, providing contextual information to facilitate accurate translation and alignment.

Forward translation: The source sentence in MSA undergoes forward translation into the target dialect (e.g., Gulf) using a neural translation model. The resulting translated sentence, referred to as the target sentence, provides a contextualised counterpart of the source sentence in the target Arabic dialect.

After translation, an embedding-based alignment algorithm is conducted on the pair of source and target sentences. The primary purpose of this step is to precisely identify the target token in the target sentence that semantically corresponds to the original source token. Accurate token-level alignment is crucial to establish reliable lexical mappings. The alignment algorithm exploits transformer-based contextual embeddings obtained from models trained on texts in MSA and various Arabic dialects. A detailed description of the algorithm is provided in Section 3.2 and outlined in Algorithm 1.

Algorithm 1: Embedding-based alignment**Input:** Source sentence S , Translated sentence T , layer ℓ , threshold θ **Output:** Aligned token index pairs $\mathcal{P}$ Run the transformer on (S, T) with `output_hidden_states = True`Extract hidden states at layer ℓ :

$$H_S \in \mathbb{R}^{|S| \times d}, \quad H_T \in \mathbb{R}^{|T| \times d}$$

Remove special tokens ([CLS], [SEP]) from H_S and H_T

Normalize embeddings to unit vectors:

$$\hat{H}_S[i] = \frac{H_S[i]}{\|H_S[i]\|}, \quad \hat{H}_T[j] = \frac{H_T[j]}{\|H_T[j]\|}$$

Compute similarity matrix:

$$\text{Sim} = \hat{H}_S \times \hat{H}_T^\top$$

Initialize $\mathcal{P} \leftarrow \emptyset$ **for** $i \leftarrow 1$ **to** $|S|$ **do** $j^* \leftarrow \arg \max_j \text{Sim}_{i,j}$ **if** $\text{Sim}_{i,j^*} \geq \theta$ **then** $\mathcal{P} \leftarrow \mathcal{P} \cup \{(i, j^*)\}$ **return** $\mathcal{P}$

Back translation: The target sentence is translated back into the original Arabic variety of the source sentence, generating a back-translated sentence. This step helps evaluate the consistency and reliability of the initial forward translation.

The embedding-based alignment algorithm (see Section 3.2, Algorithm 1) is applied again, this time to align the source sentence with the back-translated sentence. This alignment identifies the back-translated token (referred to in the figure as “back_token” for brevity) aligned to the original source token, providing additional confirmation of lexical consistency throughout the translation process.

Semantic validation: Semantic validation involves two complementary assessments to ensure the semantic quality and reliability of the lexicon entries as follows:

- **Semantic fidelity** (Sim_{fid}): Semantic fidelity measures the semantic closeness between the original source sentence and the forward-translated (target) sentence. It evaluates how well the forward-translated (target) sentence preserves the meaning of the original source sentence [49,50].
- **Round-trip consistency** (Sim_{cons}): Round-trip consistency measures the semantic closeness between the original source sentence and the back-translated sentence. It evaluates whether meaning is preserved through a two-step transformation (i.e., translating a sentence from a source dialect to a target dialect and then back to a source dialect) [51,52].

Both semantic validation measures (i.e., semantic fidelity and round-trip consistency) are computed using cosine similarity between sentence-level embeddings produced by the Sentence-BERT (SBERT) model [53]. In general, given two sentence embeddings a and b , cosine similarity is computed as follows:

$$\text{Cosine Similarity}(a, b) = \frac{a \cdot b}{\|a\| \|b\|} = \frac{\sum_{i=1}^d a_i b_i}{\sqrt{\sum_{i=1}^d a_i^2} \sqrt{\sum_{i=1}^d b_i^2}}. \quad (1)$$

where d represents the dimensionality of the sentence embeddings produced by the Sentence-BERT model.

Although both semantic validation measures share the same mathematical formulation, they differ in the pairs of sentences they evaluate. For semantic fidelity (Sim_{fid}), embeddings from the original source sentence and the forward-translated target sentence are compared. For round-trip consistency (Sim_{cons}), embeddings from the original source sentence and the back-translated sentence are compared. The use of both semantic validation measures helps in checking different dimensions of semantic integrity in the lexicon creation process.

A higher semantic fidelity score (close to 1.0) indicates excellent semantic preservation, indicating a reliable translation for lexicon inclusion. Similarly, a high round-trip consistency score indicates that the source and its back-translation are semantically aligned, providing evidence of cycle-consistency.

Validation criteria: Lexicon entries undergo strict validation based on semantic thresholds to determine their acceptance into the final multidialectal Arabic lexicons.

- Entries (i.e., source token and target token) are automatically accepted if they fulfil either of the following conditions:
 - High semantic fidelity: $Sim_{fid} \geq 0.9$.
 - Moderately high scores in both round-trip consistency and semantic fidelity: $Sim_{cons} \geq 0.85$ and $Sim_{fid} \geq 0.85$.
- Entries that do not meet these conditions are marked for human review in order to ensure human oversight for uncertain cases.

3.2. Embedding-Based Alignment Algorithm

Embedding-based alignment is used to identify corresponding words between sentence pairs by comparing their contextual embeddings produced by a transformer-based model trained on texts from various Arabic varieties.

Below, the embedding-based alignment algorithm is described in detail using general notation. For simplicity and clarity, the sentences involved are referred to as the source sentence (S) and the translated sentence (T).

Algorithm 1 illustrates the embedding-based alignment, an unsupervised word-alignment technique that leverages contextual embeddings.

Implementation details for reproducibility (word-level alignment): We use MARBERT to obtain contextual representations, extracting embeddings from the 8th hidden layer (0-indexed). Since MARBERT applies subword tokenisation, we perform alignment at the word level by aggregating subword embeddings that belong to the same original word as follows: for each word, we compute the *mean* of its subword vectors to obtain one word embedding. We exclude special tokens (e.g., [CLS] and [SEP]) and padding tokens using the attention mask. Alignment is computed in the direction $S \rightarrow T$ (many-to-one is allowed) by selecting, for each source word, the target word with maximum cosine similarity, while ties are broken by choosing the smallest target index. We keep an aligned pair only if its similarity is $\geq \theta$ (we set $\theta = 0.45$, see experimental settings in Section 4.5).

The inputs, in addition to the source sentence (S) and translated sentence (T), are layer ℓ , which is the transformer layer selected for extracting hidden states, and threshold θ , which is a similarity cutoff value specifying acceptable alignments. The output is a set $\mathcal{P}$ containing pairs of aligned token indices, where each pair represents a mapping from a source token to translated token. For example, $[(0, 0), (1, 2), (2, 3)]$ indicate that source token index 0 aligned to translated index 0 and source index 1 aligned to translated index 2 and so on.

The sentences S and T are first encoded using a transformer-based model. The model produces hidden states (embeddings capturing contextual meaning) at multiple layers. We specifically extracted the hidden states from the selected layer ℓ . The resulting matrices are H_S , the hidden states matrix for sentence S with dimensions $|S| \times d$, and H_T , the hidden states matrix for sentence T with dimensions $|T| \times d$, where $|S|$ and $|T|$ represent the token counts for each sentence, and d is the dimensionality of the hidden state embeddings. By specifying `output_hidden_states = True`, we are asking the model to return all of its layer-by-layer representations, not just the final output. This is necessary because our alignment algorithm requires extracting embeddings from an intermediate layer. Lower layers lack sufficient context, while top layers are too specialised. Therefore, using lower layers or only the final layer is insufficient [54].

The special tokens introduced by the transformer such as $[CLS]$ and $[SEP]$ are removed from H_S and H_T in order to ensure that only meaningful and context-relevant tokens remain for accurate alignment. Then, each embedding vector (i.e., each row in matrices H_S and H_T) is normalised to have unit length in order to ensure that comparisons via cosine similarity are not affected by magnitude differences as follows:

$$\hat{H}_S[i] = \frac{H_S[i]}{\|H_S[i]\|}, \quad \hat{H}_T[j] = \frac{H_T[j]}{\|H_T[j]\|}$$

The normalised embeddings are then used to compute a similarity matrix Sim , where each entry $Sim_{i,j}$ represents the cosine similarity between token i from the source sentence and token j from the translated sentence as follows:

$$Sim = \hat{H}_S \times \hat{H}_T^T$$

The algorithm iterates over each source token (row in Sim). For each source token i , the algorithm identifies the most similar translated token j^* , having the highest similarity score, as follows: $j^* \leftarrow \arg \max_j Sim_{i,j}$

If the similarity score exceeds the predefined threshold θ , the pair (i, j^*) is added to the alignment set $\mathcal{P}$.

3.3. Proposed Code-Mixed Augmentation for QA

This section details the proposed two-stage pipeline for Arabic code-mixed data augmentation, which produces questions containing words from two Arabic varieties while preserving the original intent and meaning. Given an input question written in some Arabic variety, called the source variety V_s , the pipeline generates a code-mixed version by partially substituting tokens with counterparts drawn from a different mixing variety (V_m). We use source variety (V_s) to refer to the variety of the original question. The mixing variety (V_m) is the other variety from which replacements are drawn. Figure 3 illustrates how our code-mixed data augmentation method works. It takes original questions and transforms them into code-mixed questions.

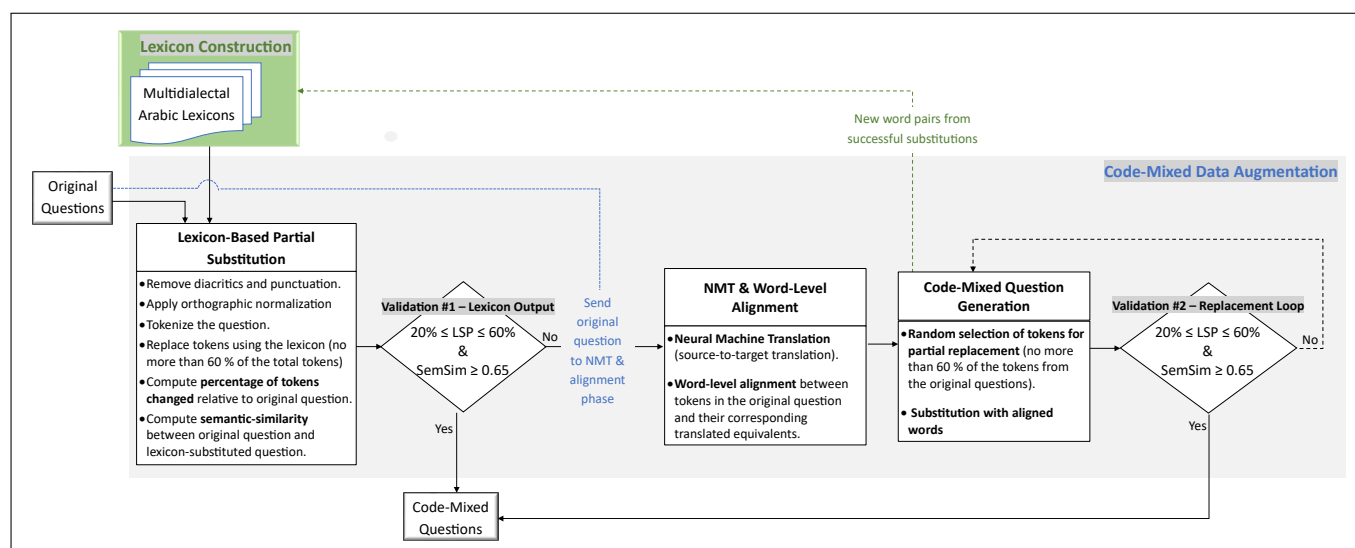


Figure 3. Process of code-mixed data augmentation. LSP stands for lexical substitution percentage, and SemSim denotes semantic similarity.

Lexicon-Based Partial Substitution: We begin with an original question written in the source variety (V_s) and a second variety, called the mixing variety (V_m), from which we will draw replacements. The algorithm starts by applying preprocessing steps to remove all Arabic diacritics and punctuation marks. It also applies orthographic normalisation, standardizing word formats to prevent missed lexicon matches. Then, the original question is split into tokens, and each word is looked up in a $V_s \rightarrow V_m$ lexicon. When a mapping exists, the V_s word is replaced by its V_m counterpart. When no mapping is found, the original word is kept. The method intentionally performs only a partial substitution, ensuring the number of replaced words does not exceed approximately 60% of the total tokens in the original question as the method's goal is to mix varieties, not to fully convert the question.

After substitution, the algorithm computes two important measures to validate the candidate code-mixed question (i.e., modified question) produced by lexicon partial substitution. The first measure is lexical change percentage (LSP), which measures the proportion of substituted words relative to the total original words. The second is semantic similarity (SemSim), which assesses how closely the candidate code-mixed question aligns semantically with the original. We then apply the validation criteria that the candidate is accepted only if $20\% \leq LSP \leq 60\%$ and $SemSim \geq 0.65$. We provide extended, step-by-step examples of accepted/rejected code-mixed candidates and how LSP/SemSim preserve semantic fidelity in Appendix B (Figure A1). The semantic similarity is computed using the cosine similarity between sentence embeddings of the original and candidate questions (see Section 3.1 for a detailed explanation and formulas regarding semantic validation). This explicit limitation of substituting no more than approximately 60% of the total tokens specifically intended to produce code-mixed questions with balanced Arabic dialect representation and semantic coherence. If the candidate passes, it is released as a final code-mixed question. If it fails, the modified question is discarded and the original question is sent to an alternative phase, which is neural machine translation with word-level alignment.

Neural Machine Translation and Word-Level Alignment: When the initial lexicon-based partial substitution does not produce an acceptable code-mixed text (i.e., fails to satisfy the lexical substitution and semantic similarity criteria), the algorithm employs a fallback approach involving neural machine translation (NMT) to obtain a full translation of the original question $V_s \rightarrow V_m$. The algorithm uses a transformer-based translation model to obtain a full translation of the original question. Translation here is instrumental

as it provides a parallel sentence that allows to identify reliable word-level alignments between V_s and V_m .

After obtaining the fully translated question, the algorithm establishes a word-level alignment between words in the original question and their corresponding translated counterparts. To extract these alignments, we apply a specialised Arabic transformer model and reuse the embedding-based alignment algorithm described in Section 3.2 (see Algorithm 1). In brief, the alignment algorithm runs a specialised Arabic transformer model which generates contextualised numeric representations (embeddings) for each word in both questions. These embeddings capture semantic information of words based on their context. Then the algorithm removes special markers, normalises those embeddings, and computes a similarity matrix between every source word and every translated word. For each source word it selects the best-matching translated word after checking if the similarity exceeds a predefined threshold. Finally, the pair is recorded as an aligned word pair.

Code-Mixed Sentence Generation: After alignment between source and translated questions, the algorithm proceeds to generate a code-mixed question through partial substitution of words, guided by the established word alignments. To ensure the generation of balanced sentences, the algorithm randomly selects approximately half and never exceeding 60% of the total words from the source question for replacement. Each randomly selected source word is replaced by its aligned translated counterpart as identified previously through the alignment process. This creates a new question composed of both original words and substituted translated words, resulting in a candidate code-mixed question. The process repeats iteratively, creating multiple candidate code-mixed questions, each time randomly choosing a different subset of source words for substitution. This process of random sampling promotes lexical diversity in the augmented data, while the alignments ensure substitutions remain context-appropriate. After each iteration, the candidate code-mixed question is evaluated according to the following two predefined criteria: percentage of substituted words (LSP) and cosine-based semantic similarity to the original question (SemSim). The pipeline accepts the first candidate that satisfies $LSP = 20\text{--}60\%$ and $SemSim \geq 0.65$, and the iterative process stops immediately.

4. Experimental Setup

In this section, we present the evaluation metrics, experimental settings, and datasets used in our generalised zero-shot cross-dialectal experiments to evaluate our proposed method (code-mixed data augmentation) and compare it against the baseline on the ArDQA benchmark [12].

4.1. Evaluation Metrics

Following the SQuAD machine reading comprehension benchmark [55], evaluation for extractive QA is typically based on lexical matching between the predicted answer span and the gold answer. We therefore report two standard metrics as follows:

Exact Match (EM). A strict, binary metric that assigns a score of 1 if the predicted answer exactly matches the gold answer and 0 otherwise. EM thus measures the proportion of questions for which the model recovers the gold answer span exactly.

$$EM_i = \begin{cases} 1, & \text{if } \hat{a}_i = a_i \\ 0, & \text{otherwise} \end{cases}$$

Aggregated over a dataset of N questions, EM is the mean of these indicator values, typically reported as a percentage as follows:

$$EM = \frac{1}{N} \sum_{q=1}^N EM_q$$

F1 Score. A more forgiving metric that gives partial credit for overlapping tokens between the prediction and the gold answer. Following [55], both prediction and gold answer are treated as bags of tokens, and F1 is computed as the harmonic mean of token-level precision and recall. This metric captures how well the predicted span overlaps with the gold span even when they are not identical.

Let P denote the bag of tokens in the prediction and G the bag of tokens in the gold answer. The number of overlapping tokens between P and G is computed as follows:

$$M = \sum_{w \in (P \cup G)} \min(\text{count}_P(w), \text{count}_G(w)).$$

At the token level, we compute *Precision* as the fraction of predicted tokens that are correct and *Recall* as the fraction of gold tokens that are recovered as follows:

$$\text{Precision} = \frac{M}{|P|}, \quad \text{Recall} = \frac{M}{|G|}.$$

where $|P|$ and $|G|$ represent the sizes (total token counts) of P and G , respectively. The token-level F1 score is then calculated as the harmonic mean of precision and recall:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

The official, widely used evaluation script for SQuAD v2.0 includes preprocessing operations such as stripping ASCII punctuation and removing extra spaces. For Arabic, we used a slightly modified version of the SQuAD v2.0 evaluation script to better handle Arabic text (<https://huggingface.co/ZeyadAhmed/AraElectra-Arabic-SQuADv2-QA> accessed on 10 August 2025; <https://github.com/zeyadahmed10/Arabic-MRC/blob/main/evaluatev2.py> accessed on 10 August 2025). A brief inspection showed that the modifications mainly involve the addition of an Arabic text preprocessor (AraBERT Preprocessor). This preprocessor performs standard Arabic text normalisation, including removing diacritical and elongation marks, normalising letter variants (e.g., different forms of *Alif*), and preserving Arabic punctuation marks such as the Arabic comma and the Arabic question mark while separating them with spaces. In our setup, we retained all these preprocessing steps except that we removed Arabic punctuation marks entirely.

4.2. Datasets

In our experiments, we use ArDQA, a parallel, multi-domain benchmark for cross-dialectal Arabic extractive QA [12]. ArDQA provides contexts, questions, and answer spans in five Arabic varieties (Egyptian (EGY), Gulf (GLF), Levantine (LEV), Maghrebi (MGR), and MSA), spanning three domains: SQuAD-style texts, lifestyle vlogs, and cultural narratives. Because ArDQA is parallel across five varieties, we distinguish between (i) *parallel* QA triples (unique underlying items shared across varieties) and (ii) individual QA triples ($\text{parallel} \times 5$). In total, ArDQA contains 8150 individual QA examples: 499 parallel triples from a revised Arabic-SQuAD sample (2495 individual triples), 607 parallel triples from lifestyle vlogs (3035 individual triples), and 524 parallel triples from narrative transcripts (2620 individual triples). Each domain is split into development and test sets as follows: ArDQA-SQuAD uses 655/1840 individual triples (dev/test), ArDQA-Vlogs uses 855/2180,

and ArDQA-Narratives uses 800/1820. To clarify the scale of the experimental pipeline, Table 1 summarises the training and evaluation set sizes.

Table 1. Training and evaluation data sizes used in our experiments. Arabic-SQuADv2.0 is used for MSA fine-tuning (train split only). ArDQA provides parallel QA triples across five Arabic varieties. We report both parallel triples (unique underlying items) and individual triples (parallel $\times$ 5).

Dataset	Split	Parallel Triples	Individual Triples
Arabic-SQuADv2.0	Train	–	76,840
ArDQA-SQuAD	Dev/Test	131/368	655/1840
ArDQA-Vlogs	Dev/Test	171/436	855/2180
ArDQA-Narratives	Dev/Test	160/364	800/1820

We first fine-tune the MSA QA baseline on the training split only of Arabic-SQuADv2.0 [35] (76,840 QA triples). We then evaluate both the baseline and the proposed inference-time question adaptation method on the ArDQA test sets under generalised zero-shot cross-dialectal transfer. The ArDQA development sets are used *only* for hyperparameter tuning of the MSA QA model (i.e., they are not used as additional supervised training data). Table 2 reports the corresponding development (validation) performance per domain.

Table 2. Development (validation) performance used for hyperparameter tuning on ArDQA (per domain). The ArDQA development splits are used only for tuning. All reported results in the paper are on the corresponding ArDQA test splits.

Domain (ArDQA Dev)	F1	EM
SQuAD-style	72.20	55.34
Vlogs	62.50	39.88
Narratives	39.11	10.24

4.3. Dataset Preprocessing

The data were preprocessed before carrying out experiments and training models in order to standardise the text and reduce vocabulary sparsity. This normalisation is applied to the text used in our pipeline (training/inference), while EM/F1 are computed using the evaluation-script preprocessing described in Section 4.1. The preprocessing pipeline involved removing diacritics and normalising letter variants, such that Arabic letters that can be written in interchangeable orthographic forms were mapped to a single form. For example, all forms of *Alif* (throughout the paper, Arabic words appearing in the text are represented using *Buckwalter transliteration* [56]) were normalised to a bare *Alif*, *Alif maqSwrah* was mapped to dotted *yA'*, and *tA' mrbwTh* was converted to *hA'*. The AraNLP library [57] was used to preprocess the data.

4.4. Baseline

To evaluate the effect of our proposed code-mixed data augmentation method on generalised zero-shot cross-dialectal Arabic QA, we compared it against the generalised zero-shot baseline introduced by Althobaiti [12]. To the best of our knowledge, this is the only existing work that systematically investigates cross-dialectal Arabic QA. In their study, Althobaiti evaluated three transformer-based MSA QA models fine-tuned on the Arabic-SQuADv2.0 dataset [35], and they found that an AraELECTRA-based model achieved the best overall performance on ArDQA. AraELECTRA is an Arabic adaptation of the

ELECTRA architecture, pretrained using a discriminator-based replaced-token detection objective [33]. When fine-tuned on Arabic-SQuADv2.0 for extractive QA, this model is referred to as AraELECTRA-MSA-QA. On ArDQA, AraELECTRA-MSA-QA constitutes the strongest MSA QA system reported by Althobaiti, and we therefore adopted it as our baseline. We re-implemented their generalised zero-shot cross-dialectal evaluation setting for AraELECTRA-MSA-QA and treated this reproduction as the baseline model in our experiments.

4.5. Experimental Settings

For sentence-level semantic similarity, we used the *paraphrase-multilingual-MiniLM-L12-v2* model, a multilingual sentence-embedding model from the Sentence-Transformers family. It maps sentences from more than 50 languages into a shared vector space, such that semantically similar texts are embedded close to each other [53]. The model is trained in the Sentence-BERT (SBERT) framework with knowledge distillation to obtain a compact encoder that preserves the semantic geometry learned by a larger teacher model across many languages [53,58,59].

For neural machine translation, we used the *nllb-200-distilled-600M* model, a compact sequence-to-sequence Transformer for machine translation across 200 languages, released by Meta's No Language Left Behind (NLLB) project [60,61]. The "distilled-600M" variant is a dense student model (approximately 600M parameters) trained via knowledge distillation from larger NLLB teachers to provide strong translation quality with substantially lower latency and memory requirements than the largest models [62].

For token-level alignment between the original source sentence and its translation, we used MARBERT [63], a BERT-based masked language model pretrained specifically for Arabic, with a heavy focus on dialectal Arabic from Twitter in addition to MSA. MARBERT follows the BERT-base architecture with 12 Transformer layers and 12 attention heads. When inducing source–target token alignments, we extracted token embeddings from the 8th hidden layer (0-indexed) before computing the similarity matrix. This mid-layer offered a good balance as follows: lower layers overemphasised surface form and lacked sufficient contextual understanding, whereas upper layers were more task-specialised and less stable for word matching. In brief development experiments across layers 6 to 11, layer 8 consistently yielded the best trade-off between alignment precision and coverage. A source–target token pair was accepted only if its cosine similarity exceeded a predefined threshold. We set the token-level acceptance threshold to $\theta = 0.45$ in the alignment algorithm (see Algorithm 1). This value was selected based on preliminary development experiments aimed at balancing precision and coverage as follows: lower values (below 0.45) produced aligned pairs but increased the risk of incorrect alignments, whereas values at or above 0.45 resulted in fewer aligned pairs and more missing correspondences.

Efficiency and Scalability. In our setting, the augmentation pipeline is applied as a preprocessing step to generate augmented questions prior to evaluation, and therefore it does not add overhead to QA inference. In deployment, Stage 1 (lexicon-based partial substitution with semantic-fidelity validation) can be used as a low-latency online preprocessing step, while the more expensive Stage 2 (NMT + embedding-based alignment) can be invoked selectively (e.g., only when Stage 1 fails to produce a valid candidate), enabling a cost–latency trade-off. Computationally, Stage 1 is lightweight (linear in question length with constant-time lexicon lookups), whereas Stage 2 is dominated by NMT decoding. The alignment step adds a similarity matrix computation that remains small in practice because questions are short. Finally, the augmentation procedure is independent per question, enabling straightforward parallelisation. Neural computations (SemSim/NMT/alignment embeddings) can be executed in batches for efficiency.

5. Results

5.1. Baseline Performance

Tables 3–5 report the generalised zero-shot cross-dialectal results of the AraELECTRA-MSA-QA baseline model on the three ArDQA domains (ArDQA-SQuAD, ArDQA-Vlogs, and ArDQA-Narratives). Each table shows F1/EM scores for all combinations of context (rows) and question (columns) varieties. Our re-implementation reproduces the results of Althobaiti [12] on all three domains.

Table 3. Generalised zero-shot cross-dialectal results (F1/EM) of the AraELECTRA-MSA-QA baseline model on the ArDQA-SQuAD test set. Rows indicate the context variety and columns the question variety. Variety abbreviations are as follows: Egyptian (EGY), Gulf (GLF), Levantine (LEV), Maghrebi (MGR), and Modern Standard Arabic (MSA).

C/Q	EGY	GLF	LEV	MGR	MSA
EGY	71.66/59.51	70.84/57.34	70.12/57.88	65.57/51.63	70.54/57.88
GLF	70.83/58.70	73.76/60.87	70.41/58.15	66.90/54.89	73.86/61.41
LEV	67.17/50.27	66.73/49.73	66.72/50.54	63.43/47.01	66.52/50.27
MGR	69.12/57.34	72.44/58.42	70.28/57.61	66.35/53.80	72.24/59.24
MSA	71.51/57.07	72.89/58.70	70.64/56.52	65.98/53.26	76.19/61.96

Table 4. Generalised zero-shot cross-dialectal results (F1/EM) of the baseline model on the ArDQA-Vlogs test set. Rows indicate the context variety and columns the question variety.

C/Q	EGY	GLF	LEV	MGR	MSA
EGY	63.90/37.93	60.36/34.71	59.80/34.94	49.06/28.05	64.06/38.16
GLF	61.25/38.62	64.47/41.74	62.35/39.45	53.11/31.42	66.14/43.58
LEV	59.73/37.70	64.19/41.74	63.00/40.37	52.94/32.57	64.87/43.12
MGR	53.29/27.59	56.81/30.73	56.41/29.36	57.11/31.19	59.58/32.34
MSA	61.20/36.32	65.11/40.14	63.10/38.53	54.30/33.49	67.01/42.20

Table 5. Generalised zero-shot cross-dialectal results (F1/EM) of the baseline model on the ArDQA-Narratives test set. Rows indicate the context variety and columns the question variety.

C/Q	EGY	GLF	LEV	MGR	MSA
EGY	35.75/11.26	30.15/8.24	37.55/12.36	28.50/7.97	39.11/13.46
GLF	37.02/13.74	40.80/14.29	39.10/13.46	31.59/10.16	42.22/16.21
LEV	32.62/9.94	35.14/9.67	38.31/12.98	27.05/6.35	38.51/12.98
MGR	30.00/6.32	32.84/7.69	34.62/7.69	31.70/6.87	36.56/9.34
MSA	37.28/10.47	34.78/8.82	35.60/9.09	30.35/7.16	43.83/14.05

Overall, the baseline model performs best on the ArDQA-SQuAD domain, where the formal, edited texts are structurally close to MSA. Dialect-matched settings (diagonal entries) yield the highest scores, whereas generalised cross-dialectal conditions (off-diagonal entries) show moderate but consistent drops. In line with [12], MSA behaves as the most stable “bridge” variety as follows: when used either as context or as question, it produces comparatively strong performance across other dialects, while Maghrebi questions tend to be the most challenging.

Performance decreases noticeably on ArDQA-Vlogs and is lowest on ArDQA-Narratives. These results reflect the increasing difficulty of spontaneous, conversational, and narrative discourse. Across these domains, MSA remains the most robust variety in both context and question roles, whereas Maghrebi questions again yield the weakest results. These results are consistent with the analysis reported by [12]. In the following subsections, we therefore focus on how our proposed code-mixed data augmentation method handles these patterns and improves over this baseline.

5.2. Performance of the Proposed Method

In this subsection, we analyse how our proposed code-mixed data augmentation method affects generalised zero-shot cross-dialectal performance on ArDQA. For consistency and comparability, we use the same AraELECTRA-MSA-QA model and experimental settings as the baseline model.

Tables 6–8 report the results of the proposed method on the three ArDQA domains, and we compare these scores directly with the baseline to quantify the gains from code-mixed training data. To avoid heuristic effect-size categorisation (e.g., labelling $\Delta > 1$ as “notable”), we report all Δ entries in Tables 6–8 without threshold-based formatting. Statistical evidence for each off-diagonal pair is provided via paired-bootstrap 95% confidence intervals and Holm–Bonferroni-corrected one-sided p -values in Appendix A (Tables A1–A3), with an aggregated domain-level summary in Table 9.

Overall, our proposed code-mixed question augmentation method improves performance across the three domains (Table 9), with gains concentrated in dialect-mismatched settings. This improvement is achieved by code-mixing the question with lexical cues from the context dialect, which reduces lexical divergence between the question and context.

Table 6. Results of the proposed method for generalised cross-dialectal transfer on the ArDQA-SQuAD test set. Columns show question variety; rows show context variety. (a) Absolute scores. (b) Relative improvement compared to baseline.

C/Q	EGY	GLF	LEV	MGR	MSA
(a) Absolute performance scores (F1/EM).					
EGY	71.66/59.51	71.15/58.16	70.37/58.46	65.74/51.69	72.17/59.41
GLF	72.67/60.23	73.76/60.87	72.07/59.68	69.49/56.42	75.29/62.18
LEV	67.17/50.32	68.16/51.03	66.72/50.54	66.77/49.30	67.91/50.27
MGR	69.17/58.22	73.12/58.77	70.48/58.73	66.35/53.80	73.35/60.48
MSA	72.80/58.39	74.13/58.94	70.80/56.80	66.99/53.26	76.19/61.96
(b) Relative improvement (Δ F1/ Δ EM) of the proposed method compared to the baseline.					
EGY	0.00/0.00	0.31/0.82	0.25/0.58	0.17/0.06	1.63/1.53
GLF	1.84/1.53	0.00/0.00	1.66/1.53	2.59/1.53	1.43/0.77
LEV	0.00/0.05	1.43/1.30	0.00/0.00	3.34/2.29	1.39/0.00
MGR	0.05/0.88	0.68/0.35	0.20/1.12	0.00/0.00	1.11/1.24
MSA	1.29/1.32	1.24/0.24	0.16/0.28	1.01/0.00	0.00/0.00

In the SQuAD domain, gains are generally modest, with statistical support appearing more often for EM than for F1 after Holm correction (Appendix A). Several dialect-mismatched pairs achieve significant improvements in both metrics, particularly for Maghrebi questions, as follows: Gulf context with Maghrebi questions improves by +2.59 F1 and +1.53 EM, and Levantine context with Maghrebi questions improves by

+3.34 F1 and +2.29 EM. Additional pairs show significant EM gains without a corresponding significant F1 gain, such as Egyptian context with MSA questions (+1.53 EM) and Gulf context with Levantine questions (+1.53 EM). We also observe settings where F1 improves significantly but EM does not, for example, MSA context with Gulf questions (+1.24 F1). Overall, the proposed method yields consistent positive trends across many off-diagonal pairs, with the strongest and most statistically supported improvements concentrated in specific dialect-mismatch settings.

Table 7. Results of the proposed method for generalised cross-dialectal transfer on the ArDQA-Vlogs test set. Columns show question variety; rows show context variety. (a) Absolute scores. (b) Relative improvement compared to baseline.

C/Q	EGY	GLF	LEV	MGR	MSA
(a) Absolute performance scores (F1/EM).					
EGY	63.90/37.93	61.74/37.36	60.93/34.94	51.20/30.28	65.73/39.75
GLF	62.97/39.96	64.47/41.74	63.92/39.45	55.04/32.59	68.80/45.32
LEV	61.12/38.80	66.92/41.74	63.00/40.37	56.17/34.24	65.87/44.45
MGR	54.74/28.71	58.63/32.07	57.59/31.72	57.11/31.19	60.58/33.23
MSA	61.21/37.66	65.30/41.48	63.98/40.28	55.93/33.49	67.01/42.20
(b) Relative improvement ($\Delta F1/\Delta EM$) of the proposed method compared to the baseline.					
EGY	0.00/0.00	1.38/2.65	1.13/0.00	2.14/2.23	1.67/1.59
GLF	1.72/1.34	0.00/0.00	1.57/0.00	1.93/1.17	2.66/1.74
LEV	1.39/1.10	2.73/0.00	0.00/0.00	3.23/1.67	1.00/1.33
MGR	1.45/1.12	1.82/1.34	1.18/2.36	0.00/0.00	1.00/0.89
MSA	0.01/1.34	0.19/1.34	0.88/1.75	1.63/0.00	0.00/0.00

Table 8. Results of the proposed method for generalised cross-dialectal transfer on the ArDQA-Narratives test set. Columns show question variety; rows show context variety. (a) Absolute scores. (b) Relative improvement compared to baseline.

C/Q	EGY	GLF	LEV	MGR	MSA
(a) Absolute performance scores (F1/EM).					
EGY	35.75/11.26	35.48/9.87	39.05/13.99	35.00/12.97	39.90/14.61
GLF	38.04/15.36	40.80/14.29	40.39/15.08	37.92/15.79	42.84/16.75
LEV	34.03/11.18	36.73/10.09	38.31/12.98	34.43/9.79	40.43/14.40
MGR	30.45/7.43	33.54/10.58	35.69/9.32	31.70/6.87	36.89/9.46
MSA	38.01/11.35	36.93/10.71	37.31/12.08	42.48/15.61	43.83/14.05
(b) Relative improvement ($\Delta F1/\Delta EM$) of the proposed method compared to the baseline.					
EGY	0.00/0.00	5.33/1.63	1.50/1.63	6.50/5.00	0.79/1.15
GLF	1.02/1.62	0.00/0.00	1.29/1.62	6.33/5.63	0.62/0.54
LEV	1.41/1.24	1.59/0.42	0.00/0.00	7.38/3.44	1.92/1.42
MGR	0.45/1.11	0.70/2.89	1.07/1.63	0.00/0.00	0.33/0.12
MSA	0.73/0.88	2.15/1.89	1.71/2.99	12.13/8.45	0.00/0.00

Table 9. Domain-level summary of cross-dialect transfer gains on ArDQA. For each domain, we consider the 20 off-diagonal dialect pairs (context $\neq$ question). We report the mean and median improvements in percentage points (Δ = proposed – baseline) across the 20 pairs. We also report how many of the 20 pairs remain significant after Holm–Bonferroni correction applied across the 20 off-diagonal tests within the domain (separately for $\Delta F1$ and ΔEM). Significance is computed using a one-sided paired bootstrap test over questions (testing $\Delta > 0$, $B = 10,000$).

Domain	Mean $\Delta F1$	Median $\Delta F1$	Mean ΔEM	Median ΔEM
SQuAD	+1.09	+1.18	+0.87	+0.85
Vlogs	+1.54	+1.51	+1.25	+1.34
Narratives	+2.75	+1.46	+2.27	+1.63

Note. Significant off-diagonal pairs after Holm–Bonferroni correction (out of 20): SQuAD: $\Delta F1 = 4/20$, $\Delta EM = 11/20$; Vlogs: $\Delta F1 = 3/20$, $\Delta EM = 16/20$; Narratives: $\Delta F1 = 6/20$, $\Delta EM = 17/20$.

In the Vlogs domain, the proposed method yields clear gains in many dialect-mismatched settings, with statistical support more frequently observed for EM than for F1 after Holm correction (Appendix A). For example, Egyptian context with Maghrebi questions improves by +2.14 F1 and +2.23 EM and remains significant for both metrics. Levantine context with Maghrebi questions shows similarly strong gains (+3.23 F1, +1.67 EM), also significant for both metrics. In contrast, some pairs show improvements primarily in EM as follows: Gulf context with MSA questions increases by +2.66 F1 and +1.74 EM, but only the EM gain remains significant after correction. Likewise, Maghrebi context with Gulf questions improves by +1.82 F1 and +1.34 EM, but neither metric remains significant after Holm correction. However, Maghrebi context with Levantine questions yields a significant EM gain (+2.36 EM). Overall, these results indicate that, in Vlogs, code-mixed question augmentation more consistently improves exact span matching (EM), while F1 gains are less uniformly significant across dialect pairs.

The Narratives domain shows the largest relative increases, which is consistent with its being the most challenging domain. Several context–question dialect-mismatched combinations show large gains, and many remain statistically significant after Holm correction (Appendix A). In particular, settings with Maghrebi questions exhibit strong and statistically supported gains as follows: Egyptian context with Maghrebi questions improves by +6.50 F1 and +5.00 EM, Gulf context with Maghrebi questions improves by +6.33 F1 and +5.63 EM, and Levantine context with Maghrebi questions increases by +7.38 F1 and +3.44 EM. Notably, MSA context with Maghrebi questions yields the largest gain, reaching +12.13 F1 and +8.45 EM. In contrast, improvements are generally smaller when Maghrebi serves as the context, reflecting the difficulty of Maghrebi narrative contexts.

The overall results confirm that our proposed method of code-mixed question augmentation consistently improves performance, with the largest improvement in generalised zero-shot cross-dialectal transfer involving Maghrebi questions, and with substantial gains in more difficult domains like narratives.

6. Discussion

The baseline generalised cross-dialectal results across context–question variety combinations reveal systematic weaknesses under dialect mismatch and domain shift. Performance is particularly sensitive to lexical divergence between question and context varieties, and it degrades further when moving from the more formal SQuAD texts to the more informal Vlogs and Narratives domains. These trends motivate explicitly addressing question–context lexical mismatch in a lightweight manner.

Tables 3–5 highlight three recurring limitations. First, performance drops substantially under domain shift from SQuAD to Vlogs and Narratives: for instance, the MSA/MSA

diagonal decreases from 76.19/61.96 (SQuAD) to 67.01/42.20 (Vlogs) to 43.83/14.05 (Narratives). Second, Maghrebi questions tend to yield the lowest scores in dialect-mismatched settings, as follows: for non-Maghrebi contexts, the MGR question column is typically the lowest (e.g., in Narratives, EGY/MGR = 28.50/7.97 and LEV/MGR = 27.05/6.35 in Table 5). Similar patterns appear in Vlogs and SQuAD in Tables 3 and 4). Third, Levantine contexts tend to underperform relative to other context varieties, particularly in SQuAD and Narratives (e.g., the LEV context row is generally lower than the EGY/GLF/MSA rows in Tables 3 and 5).

From a domain perspective, Table 9 shows that aggregate gains are largest in Narratives, followed by Vlogs and SQuAD. This aligns with increasing informality and dialectal variability in the source material, and with the added difficulty of spontaneous conversational and narrative discourse. Appendix A reports per-pair paired-bootstrap confidence intervals and Holm–Bonferroni-corrected p -values (Tables A1–A3), clarifying which dialect-mismatched settings drive these domain-level trends. Notably, after correction, statistically significant improvements are much more frequent for EM than for F1 across domains, suggesting that the method more consistently helps recover exact spans than uniformly improving partial token overlap.

Our proposed method mitigates these issues by introducing simple lexical adaptation. Specifically, it reduces lexical divergence between dialects, especially in interrogative forms. This interpretation is consistent with Appendix A, where the most robust gains are concentrated in dialect-mismatched pairs with higher lexical mismatch. Figure 4 illustrates four representative test examples in which the baseline fails to extract the correct span, while the proposed code-mixed augmentation succeeds.

In the first example, given that the model used in our experiments was pretrained primarily on MSA and fine-tuned on the MSA Arabic-SQuADv2.0 corpus, the baseline model appears to prefer the most salient, SQuAD-like descriptive pattern in the local neighbourhood of the target mention. When the Maghrebi question asks about *fAkhp bAkwy* “bacuri fruit”, the model’s attention moves to the region around *AlfAkhp AltAnyyp bAkwy* “The second fruit is bacuri”, where two competing descriptions occur: *TEmhA zy Allwz mE AlEsl* “It tastes like almonds with honey” (for *fAkhp mAry mAry* “mari-mari fruit”) and *zy Aldhb Al>byD* “like white gold” (for *fAkhp bAkwy* “bacuri fruit”). The baseline consistently prefers the familiar pattern *TEmhA zy* “tastes like” over the more colloquial *hy zy* “is like”. Therefore, the baseline incorrectly returns the description of “mari-mari fruit” rather than the correct description *zy Aldhb Al>byD* “like white gold” for *bAkwy* “bacuri fruit”.

In the second example (see Figure 4), for duration questions in Maghrebi over MSA contexts, the baseline frequently misinterprets the construction as a “who?” question and returns the name (e.g., Pedro) rather than the correct temporal span (e.g., *mn* 20 EAmA* “for 20 years”). This reflects a lack of robustness to dialectal time expressions such as *mn \$HAL hAdy* “How long has it been”.

Our proposed code-mixed data augmentation method addresses these issues by partially substituting dialectal tokens in the questions with context-dialect or MSA equivalents while enforcing semantic fidelity and balanced substitution. This produces questions where dialect words are explicitly linked to their MSA or context-dialect forms and share more words with the context sentences. This makes it easier for the model to interpret the question type and select the correct answer span. As a result, the model learns to associate dialectal ways of asking “how is X?” and “how long has Y been...?” with the appropriate answer types (short descriptive phrases or time expressions) and to focus on the right sentences, allowing it to recover the correct gold spans in many cases where the baseline fails.

Example 1	
Context (Egyptian)	Domin: Vlogs
.... أول ما تدخل منطقة الأمازون، هتشاف فواكه غريبة مش هتشافها في أي مكان ثاني. هتشاف الفاكهة دي لحظة وصولك للمطار. بتفتحها كده وتلاقي طعمها زي الآيس كريم، هي برده. دلوقتي هنجرب ماري ماري. بص إزاي شكلها ماري ماري. طعمها زي اللوز مع العسل. الفاكهة الثانية باكوري. بتفتحها كده، هي زي الذهب الأبيض. شريموليا لازم تفتحها كده، فاكهة غريبة ومميزه..... as soon as you enter the Amazon region, you will see strange fruits that you won't see anywhere else. You will see these fruits the moment you arrive at the airport. You open one like this and find that it tastes like ice cream, it is cold. Now we will try mari-mari. Look how mari-mari looks. It tastes like almonds with honey. The second fruit is bacuri. You open it like this, it is like white gold. Cherimoya, you have to open it like this, a strange and distinctive fruit	
Question (Maghrebi)	كيفاش كيوصفو باكوري ملي كتحلها؟ How is bacuri described when it is opened?
Answer	
Gold: زي الذهب الأبيض (like white gold)	
Baseline: اللوز مع العسل (almonds with honey) X	
Our proposed method: زي الذهب الأبيض (like white gold) ✓	

Example 2	
Context (MSA)	Domin: Vlogs
هذا بيدرو، يقول أنه لم يستقبل ضيفا منذ 20 عاما. الحياة غريبة! ليس لديهم صداق الحياة الحديثة. يكسب 700 دولار شهريا من بيع الأسماك. لديه ابنتان. This is Pedro, he says he has not received a guest for 20 years. Life is strange! They do not have the headache of modern life. He earns 700 dollars a month from selling fish. He has two daughters.	
Question (Maghrebi)	How long has it been since he received any guest? من شحال هادي وهو ما استقبل حتى ضيف؟
Answer	
Gold: منذ 20 عاما (for 20 years)	
Baseline: بيدرو (Pedro) X	
Our proposed method: منذ 20 عاما (for 20 years) ✓	

Example 3	
Context (Egyptian)	Domin: Narratives
الست سألتها: "جاعان يا ابني؟" "أه، جاعان جدا." قامت الست وادته أكل فورا، وشكرها على كرمها. قالت له: "يلا بسرعة، استخبي عشان جوزي راجع البيت!" جابت طبق كبير. وخبته تحت السرير. الولد كان تعبان جدا، بس كان خايف جدا بنام. لما الساعة دقت 12، سمع صوت زئير مرعب. التين The woman asked him, "Are you hungry, my son?" He replied, "Yes, I am very hungry." The woman got up and immediately gave him food, and he thanked her for her generosity. She said to him, "Come on quickly, hide, because my husband is coming home!" She brought a big plate and hid him under the bed. The boy was very tired, but he was very afraid to sleep. When the clock struck 12, he heard a terrifying roar. The dragon ...	
Question (Gulf)	وش عطت العجوز للولد؟ What did the old woman give to the boy?
Answer	
Gold: ادته أكل فورا (immediately gave him food)	
Baseline: طبق كبير (a big plate) X	
Our proposed method: ادته أكل فورا (immediately gave him food) ✓	

Example 4	
Context (Levantine)	Domin: SQuAD
مع زيادة شعبية نظام نينتندو الترفيهي ودخوله لملايين البيوت الأمريكية، بلشت بعض محلات تأجير الفيديو الصغيرة تشتري نسخها الخاصة من ألعاب NES ، وتأجرها للزبائن بنفس سعر تأجير أشرطة الفيديو لكم يوم. نينتندو ما كانت تحصل على أي ربح من هالشي غير من التكلفة الأولية للعبتها، وعلى عكس تأجير الأفلام، كانت اللعبة الجديدة توصل لرؤوف المحلات وتكون متاحة للتأجير بنفس اليوم. نينتندو اتخذت خطوات لتوقف تأجير الألعاب، بس ما اتخذت أي إجراء قانوني رسمي حتى بلشت شركة بلوكباستر فيديو تقدم تأجير الألعاب كخدمة واسعة النطاق. نينتندو ادعت إنه السماح للزبائن بتأجير الألعاب رح يقلل المبيعات With the growing popularity of the Nintendo Entertainment System and its entry into millions of American homes, some small video rental stores started buying their own copies of NES games and renting them to customers for the same price as renting videotapes for a few days. Nintendo was not making any profit from this except for the game's initial cost, and unlike movie rentals, the new game would arrive on store shelves and be available for rent on the same day. Nintendo took steps to try to stop game rentals, but it did not take any formal legal action until Blockbuster Video started offering game rentals as a large-scale service. Nintendo claimed that allowing customers to rent games would reduce sales.....	
Question (Maghrebi)	شكون خلا نينتندو تتخذ إجراء رسمي بشأن الكراء؟ Who made Nintendo take official action regarding renting?
Answer	
Gold: شركة بلوكباستر فيديو (Blockbuster Video)	
Baseline: محلات تأجير الفيديو الصغيرة (small video rental stores) X	
Our proposed method: شركة بلوكباستر فيديو (Blockbuster Video) ✓	

Figure 4. Illustration of four qualitative results from the test set where the proposed code-mixed question augmentation improves over the baseline in generalised zero-shot cross-dialectal Arabic QA. For each example, we show the original question, a context snippet, the gold answer span, the baseline prediction, and the prediction after applying our proposed method. English translations of the Arabic text are provided for clarity.

These improvements largely arise from reducing lexical divergence in short question sentences, particularly in interrogative forms. This is done by randomly replacing a subset of the question's tokens with counterparts in the context's dialect. Because question sentences are short and interrogatives carry high syntactic and semantic weight (e.g., what/when/where/why/how... etc.), substituting a few tokens, including interrogatives, can significantly improve the model's ability to map the question to the context. The effect is strongest in high-mismatch settings (especially when the question is Maghrebi), although Appendix A shows that many non-Maghrebi question varieties also exhibit significant EM gains across multiple pairs. For example, the question (\$*HAL*, "How much/How many") in Maghrebi is replaced with the MSA interrogative form (*km*). This substitution makes the interrogative directly recognisable by the MSA QA model and helps in aligning it with dialects that also use this near-MSA form. This pattern is also reflected in the per-pair significance analysis (Appendix A), where the strongest and most consistently significant gains tend to occur in settings involving higher lexical mismatch.

Changing other tokens in the question, rather than interrogatives, also harmonises the question wording with the context's dialect and enhances the indicators used to locate

answer spans. This is illustrated in Example 3 in Figure 4. In Egyptian context with Gulf question *w\$ EaTt AlEjwz llwld?* “What did the old woman give to the boy?”, replacing the Gulf verb *EaTt* “gave” by the Egyptian equivalent *Adt* refines the matching evidence for answer spans.

In the generalised cross-dialectal baselines, we observe that Levantine contexts underperform because their long colloquial passages, which contain clitics and discourse particles, may be underrepresented in pretraining. We observed once the questions are code-mixed (e.g., by randomly replacing some tokens in the question with tokens from the context’s dialect), the model no longer has to handle both context noise and question divergence simultaneously (see Example 4 in Figure 4).

In terms of domains, the Narratives domain, where contexts are loosely structured, shows the largest improvements. This is consistent with Table 9, which shows higher mean and median Δ improvements in Narratives compared to Vlogs and SQuAD when aggregating across the 20 dialect-mismatched pairs. This can be partially attributed to the fact that questions in Narratives often contain more dialectal interrogatives and function words that the model does not recognise well. Replacing these forms with those from the context’s variety makes the question intent explicit. By contrast, the SQuAD domain is formal and closer to MSA. As a result, there is less lexical divergence, so code-mixing helps, but only moderately. The Vlogs domain consists of spontaneous speech with fillers and colloquial forms. Therefore, code-mixing helps more than in SQuAD, but the contexts are shorter and more topical, so the gains remain moderate. This demonstrates that even minimal lexical adaptation can substantially reduce the cumulative effects of domain shift and dialect divergence. Appendix A further shows that statistically significant improvements are more frequent for EM than for F1 across many dialect pairs, suggesting that the method more consistently helps the model recover exact spans than uniformly improving partial token overlap.

6.1. Qualitative Error Analysis

A qualitative error analysis is performed to better understand the limitations of our proposed code-mixed question augmentation. We manually inspect a set of test examples spanning the three ArDQA domains (SQuAD-style texts, Vlogs, and Narratives) and multiple context-question variety pairs. This analysis allows us to move beyond simple EM/F1 scores and to identify systematic patterns such as answer-type confusion, shallow temporal reasoning, weak use of location cues in narratives, confusion between lexical forms and their meta-descriptions, and discourse-level anaphora resolution errors. Figure 5 presents four representative error cases.

Time Reasoning Failures. In some cases, our proposed method does not help because the error arises from temporal reasoning rather than dialect mismatch. As in example 1 in Figure 5, when the context states that the story spans *ÎA\$nwAt (1933–1935)* “three years (1933–1935)” and the question asks “For how many years does To Kill a Mockingbird take place?”, the model selects the start year “1933” instead of the duration *ÎA\$nwAt* “three years”. In this example, lexical alignment is already good after code-mixed question augmentation, but the MSA QA model misinterprets the answer type (year vs duration). Also, the model exhibits shallow temporal reasoning. It seems to pick the first important temporal mention (1933). It does not perform basic temporal reasoning over the interval “1933–1935” as three years or connect this interval to the explicitly stated duration “three years”.

Example 1	
Context (MSA)	Domain: SQuAD
تدور القصة خلال ثلاث سنوات (1933-1935) من الكساد الكبير في البلدة الخيالية "المدينة القديمة المتعبة" في مايكومب، ألاباما، مقر مقاطعة مايكومب. تركز القصة على جين لويس فينش..... (The story takes place over three years (1933–1935) of the Great Depression in the fictional "tired old town" of Maycomb, Alabama, the seat of Maycomb County. The story focuses on Jean Louise Finch.....)	
Question (Maghrebi)	على مدى شحال من عام كندور أحداث رواية "قتل طائر بري"؟ (For how many years does <i>To Kill a Mockingbird</i> take place?)
Answer	Error Type
Gold: ثلاث سنوات (three years)	Answer-type confusion & shallow temporal reasoning
Our proposed method: 1933 X	
Example 2	
Context (Egyptian)	Domain: Narratives
..... الشيء الوحيد الذي كان يملكه الراي الفقير ده هو حمار صغير. وكان يقيضي أيامه في رعاية قطع الخرفان بتوح عمدة القرية. وفي يوم من الأيام، قعد يراقب القطيع لحد ما سمع صوت بكاء فجأة. "ساعدني! ساعدني!" الولد تبع الصوت ده ومشي لحد ما وصل لحفرة في الأرض، وهناك لقي تعبان بطنه صفراء ملفوف في النار. "ساعدني، وأنا هارد لك الجميل قريباً!" قعد الراي العصاة بتاعته، واتسلق التعبان اللي بطنه صفراء عليها وخرج من وسط النار الملتهبة (..... The only thing that this poor shepherd owned was a small donkey. He spent his days taking care of the flock of sheep that belonged to the mayor of the village. One day, he sat watching the flock until he suddenly heard the sound of someone crying. "Help me! Help me!" The boy followed that sound and walked until he reached a pit in the ground, and there he found a snake with a yellow belly wrapped in flames. "Help me, and I will repay your kindness soon!" So the shepherd stretched out his stick, and the snake with the yellow belly climbed up it and came out from the midst of the blazing fire)	
Question (Levantine): شو شاف الراي بالحفرة؟ (What did the shepherd see in the pit?)	Error Type
Answer	Weak use of location cues in narratives
Gold: لقي تعبان بطنه صفراء ملفوف في النار (he found a snake with a yellow belly wrapped in flames)	
Our proposed method: قعد يراقب القطيع (he sat watching the flock) X	
Example 3	
Context (Egyptian)	Domain: Vlogs
إيه هي قصة توفالو؟ ليه الناس ما بتزور هاش؟ ده هو الفندق. دول الناس اللي بيخرجوا من المطار. ده هو الفندق. هي من فيجي، مع إنها قريبة جداً. محدش بيזור البلد دي، وده اللي بقولوه لما يرحبوا بلغتهم: تالوفا. ده هو المطعم الوحيد هنا، والغرف على البحر. شوف المكان. دي هي الغرفة. توفالو عندها مركز شرطة واحد بس ومتجر واحد للاتصالات السلكية واللاسلكية في البلد كله. ده هو المتجر الوحيد اللي ممكن تشتري منه بطاقة SIM ، بلا نجيب واحدة. (What is the story of Tuvalu? Why don't people visit it? This is the hotel. These are the people coming out of the airport. This is the hotel. She is from Fiji, even though it is very close. Nobody visits this country, and this is what they say when they welcome you in their language: talofa. This is the only restaurant here, and the rooms are on the sea. Look at this place. This is the room. Tuvalu! It has only one police station and one telecommunications shop in the whole country. This is the only shop where you can buy a SIM card, let's go get one.)	
Question (Maghrebi): كيفاش كايقولو مرحباً فتوفالو؟ (How do they say welcome in Tuvalu?)	Error Type
Answer	Form vs description (word vs meta-description)
Gold: تالوفا (talofa)	
Our proposed method: بلغتهم (in their language) X	
Example 4	
Context (Egyptian)	Domain: Vlogs
..... حتى عندهم نظارات مكتوب عليها "تغير المناخ في توفالو". لتوسيع سطح الأرض، ملوا أجزاء من سطح المياه بالرمال. زي جزر النخيل في دبي. سكان البيت ده كانوا في البيت ده. عشان كده انتقلوا هنا مؤقتاً بعد ما هدموا بيتهم They even have glasses with "Climate Change in Tuvalu" written on them. To extend the land area, they filled parts of the water surface with sand. like the Palm Islands in Dubai. The residents of this house used to be in that house. That's why they moved here temporarily after they demolished their home	
Question (Maghrebi): بشنو كيشبهو هاد العملية؟ What does this process resemble?	Error Type
Answer	Discourse-level anaphora resolution error
Gold: زي جزر النخيل في دبي (like the Palm Islands in Dubai)	
Our proposed method: نظارات (glasses) X	

Figure 5. Representative error cases of the proposed code-mixed question augmentation method in generalised zero-shot cross-dialectal Arabic QA, illustrating the main qualitative error categories (answer-type confusion, weak use of location cues, form-description confusion, and discourse-level anaphora errors). English translations of the Arabic text are provided for clarity.

Spatial and Narrative Grounding Failures. In example 2 in Figure 5 about the shepherd story, the Levantine question is asked over an Egyptian context. After our proposed code-mixed question augmentation, the Levantine interrogative *šw* “what” is changed to its Egyptian equivalent *Ayh* “what”. The remaining three words in the question are largely shared between the two Arabic dialects. However, the model incorrectly predicts “he sat watching the flock”, taken from an earlier sentence. Here, the model focuses on the visual verb “watching the flock” and fails to combine the object and location constraints (“What did the shepherd see in the pit?”). Therefore, dialectal alignment is no longer the main bottleneck. Indeed, such an error highlights a limitation in how the model links questions to specific locations within the narrative context.

Lexical-form Extraction Failures (Form vs. Description). In some cases, our proposed code-mixed questions help the MSA QA model correctly align the question with the relevant sentence in the context. Nevertheless, the model fails to extract the intended lexical form. In example 3 in Figure 5 from the Tuvalu vlog, the Egyptian context states that *wdh Ally byqwlwh lma yrHbwA blgthm: tAlwfa* “and this is what they say when they welcome you in their language: talofa”, while the Maghrebi question asks *kyfA\$ kAyqwlw mrHbA ftwfAlw?* “How do they say welcome in Tuvalu?”. After applying our code-mixing algorithm to adapt the Maghrebi question to the Egyptian context, the model correctly focuses on the relevant sentence but predicts the phrase *blgthm* “in their language” instead of the greeting word *tAlwfa* “talofa” itself. In this example, the dialectal mismatch between the Egyptian context and the Maghrebi question is resolved by code-mixing. However,

the model still prefers a meta-descriptive fragment “in their language” over the actual lexical form required by the question. This indicates that the model is not distinguishing between the target lexical item requested by the question and its surrounding descriptive context, and instead selects a semantically related descriptor.

Discourse-level Anaphora Failures. For example 4 in Figure 5, the Maghrebi question *b\$nw ky\$bhw hAd AlEmlyp?* “What does this process resemble?” contains a demonstrative noun phrase *hAd AlEmlyp* “this process”, which should be resolved to the immediately preceding description *ltwsyE sTH Al>rD mlwA>jzA' mn sTH AlmyAh bAlrmAl* “To extend the land area, they fill parts of the water surface with sand”. The comparative phrase *zy jzr Alnxyl fy dby* “like the Palm Islands in Dubai” is the correct answer to the question. After applying the code-mixing to the question, we resolve the problem of dialect mismatch between Maghrebi and Egyptian by replacing some Maghrebi words in the question with their equivalents in the Egyptian context. This involves changing the Maghrebi word *hAd* “this” to the Egyptian word *dy* “this”. Although the dialectal mismatch is removed, the model selects the noun *nZArAt* “glasses”, taken from the earlier clause *HtY Endhm nZArAt mktwb ElyhA 'tgyr AlmnAx fy twfAlw'* “They even have glasses with ‘Climate Change in Tuvalu’ written on them”. This span is semantically unrelated to the process being queried and ignores the anaphoric link between *hAd AlEmlyp* “this process” and the land-filling description. The error thus reflects a failure to track discourse structure and resolve anaphora. The model prefers a salient noun over the event referred to by the demonstrative. Therefore, it seems that this error is not due to dialectal Arabic mismatch.

Error-driven Implications. The error analysis shows that the main challenges for the MSA QA model when applied on informal domains or dialectal Arabic are not only restricted to surface-level token matching. The challenges involve deeper semantic and discourse situations. The model often struggles to align the question with the appropriate answer type (e.g., duration vs. date) to exploit explicit spatial or comparative cues and to correctly resolve references to events or processes within multi-sentence narratives. These findings motivate future improvements focusing on stronger modelling of temporal and spatial reasoning and more explicitly handling the distinction between the required word form and its descriptive explanation in the answer spans. Incorporating such insights into model design, training objectives, or data augmentation strategies is likely to yield more robust cross-dialectal Arabic QA performance than relying on additional data or larger architectures alone.

7. Conclusions and Future Work

This paper introduces a test time code-mixed data augmentation method for generalised zero-shot cross-dialectal Arabic question answering. Starting from an MSA-only QA backbone (AraELECTRA-MSA-QA trained on Arabic-SQuADv2.0), we address the setting where the context and the question come from different Arabic varieties and no additional training data are available. We build multidialectal lexicons via neural machine translation and embedding-based alignment with semantic validation, and we use these lexicons (with an NMT-alignment fallback) to rewrite test questions into balanced code-mixed forms closer to the context variety.

On the ArDQA benchmark, our approach improves on the strongest available generalised zero-shot baseline across all three domains (SQuAD-style texts, Vlogs, and Narratives), with the largest gains under challenging conditions such as Maghrebi questions over non-MSA contexts and in the noisy Vlogs and Narratives domains. These results show that controlled code-mixed question rewriting can mitigate cross-dialectal lexical gaps at test time without retraining or modifying the QA model architecture.

Limitations and Future Directions. Performance remains bounded by the underlying MSA QA model. As discussed in the qualitative error analysis (Section 6.1), several residual errors are driven by reasoning and discourse phenomena rather than dialect mismatch, including answer-type confusion and shallow temporal reasoning (e.g., duration vs. year), weak use of location/temporal cues in narratives, and imprecise span selection in lexical-form or comparison questions. Future work can address these limitations by incorporating discourse-conscious and temporal/spatial-aware modelling (e.g., coreference-aware representations and answer-type constraints).

A second limitation concerns lexicon coverage. Our lexicon is primarily token-level, which may miss dialectal idioms and multiword expressions and may not fully capture morphology and cliticisation. Coverage can also be uneven across dialects, as follows: lower-resource varieties exhibit fewer standardised written resources and noisier orthographic variation, which reduces the availability and reliability of lexical mappings and may omit region-specific terms. Future work could expand the lexicon with multiword expressions and clitic-aware segmentation, and it could explore adaptive substitution strategies beyond fixed 20–60% substitution bands.

Finally, while we evaluate on extractive QA, the proposed code-mixed augmentation framework is not QA-specific. Similar controlled rewriting (with semantic-fidelity validation and alignment-based projection when labels/spans are required) can be explored for other tasks affected by dialect mismatch, such as text classification, sequence labelling (e.g., named entity recognition), retrieval, and dialogue or generative QA.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The ArDQA dataset used in this study is publicly available through the author’s GitHub repository (<https://github.com/Maha-J-Althobaiti>, accessed on 20 October 2025). We used release v1.0. The dataset is released with predefined development and test splits.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Significance Analysis for Cross-Dialectal Arabic Extractive QA Transfer

Tables A1–A3 report the paired-bootstrap 95% confidence intervals and Holm–Bonferroni-corrected one-sided p -values for the mean improvement Δ (proposed–baseline) over the 20 off-diagonal dialect pairs (context $\neq$ question) within each domain, using $B = 10,000$ bootstrap resamples over questions. We treat results as statistically significant only when $p_{\text{holm}} < 0.05$.

Table A1. Paired bootstrap significance for off-diagonal dialect pairs on ArDQA-SQuAD. We report the mean improvements Δ (proposed–baseline) in percentage points and Holm–Bonferroni-corrected one-sided p -values across the 20 off-diagonal pairs. In the Sig. column, the left marker refers to ΔF1 and the right marker refers to ΔEM , using (**) for $p_{\text{holm}} < 0.01$, (*) for $p_{\text{holm}} < 0.05$, and (.) otherwise. C indicates the context variety and Q the question variety.

C	Q	n	ΔF1	95% CI	p_{holm}	ΔEM	95% CI	p_{holm}	Sig.
EGY	GLF	368	+0.31	[−1.86, 3.05]	0.323	+0.82	[0.00, 1.90]	0.049	./*
EGY	LEV	368	+0.25	[−2.00, 2.80]	0.384	+0.58	[0.00, 1.36]	0.136	./.
EGY	MGR	368	+0.17	[−3.11, 2.70]	0.566	+0.06	[0.00, 0.11]	0.760	./.
EGY	MSA	368	+1.63	[−1.55, 3.65]	0.219	+1.53	[0.54, 2.99]	0.002	./**

Table A1. *Cont.*

C	Q	<i>n</i>	$\Delta F1$	95% CI	p_{holm}	ΔEM	95% CI	p_{holm}	Sig.
GLF	EGY	368	+1.84	[0.36, 4.68]	0.046	+1.53	[0.53, 2.78]	0.002	*/**
GLF	LEV	368	+1.66	[−1.13, 3.97]	0.136	+1.53	[0.55, 2.86]	0.002	./**
GLF	MGR	368	+2.59	[0.33, 3.32]	0.04	+1.53	[0.56, 3.01]	0.002	*/**
GLF	MSA	368	+1.43	[−2.64, 2.19]	0.581	+0.77	[0.00, 1.90]	0.047	./*
LEV	EGY	368	0.00	[0.00, 0.00]	1.00	+0.05	[0.00, 0.15]	0.356	./.
LEV	GLF	368	+1.43	[−1.05, 3.77]	0.277	+1.30	[0.27, 2.71]	0.007	./**
LEV	MGR	368	+3.34	[0.67, 3.43]	0.03	+2.29	[0.82, 3.80]	0.006	*/**
LEV	MSA	368	+1.39	[−1.84, 3.86]	0.248	0.00	[0.00, 0.00]	1.00	./.
MGR	EGY	368	+0.05	[−1.94, 3.15]	0.330	+0.88	[0.00, 1.90]	0.051	./.
MGR	GLF	368	+0.68	[−1.47, 3.62]	0.19	+0.35	[0.00, 0.82]	0.12	./.
MGR	LEV	368	+0.20	[−1.41, 3.45]	0.17	+1.12	[0.27, 2.16]	0.017	./*
MGR	MSA	368	+1.11	[−1.66, 3.11]	0.273	+1.24	[0.29, 2.71]	0.006	./**
MSA	EGY	368	+1.29	[−2.15, 2.91]	0.40	+1.32	[0.26, 2.65]	0.005	./**
MSA	GLF	368	+1.24	[0.04, 4.57]	0.047	+0.24	[0.00, 0.81]	0.16	*./.
MSA	LEV	368	+0.16	[−1.98, 2.87]	0.230	+0.28	[0.00, 0.79]	0.226	./.
MSA	MGR	368	+1.01	[−2.32, 3.02]	0.347	0.00	[0.00, 0.00]	1.00	./.

Table A2. Paired bootstrap significance for off-diagonal dialect pairs on ArDQA-Vlogs. We report the mean improvements Δ (proposed–baseline) in percentage points and Holm–Bonferroni-corrected one-sided p -values across the 20 off-diagonal pairs. In the Sig. column, the left marker refers to $\Delta F1$ and the right marker refers to ΔEM , using (**) for $p_{\text{holm}} < 0.01$, (*) for $p_{\text{holm}} < 0.05$, and (.) otherwise. C indicates the context variety and Q the question variety.

C	Q	<i>n</i>	$\Delta F1$	95% CI	p_{holm}	ΔEM	95% CI	p_{holm}	Sig.
EGY	GLF	436	+1.38	[−2.31, 5.20]	0.230	+2.65	[1.38, 4.36]	<0.001	./**
EGY	LEV	436	+1.13	[−2.58, 4.88]	0.277	0.00	[0.00, 0.00]	1.00	./.
EGY	MGR	436	+2.14	[0.42, 5.07]	0.047	+2.23	[0.92, 3.90]	0.002	*/**
EGY	MSA	436	+1.67	[−2.44, 4.23]	0.301	+1.59	[0.46, 2.98]	0.006	./**
GLF	EGY	436	+1.72	[−0.91, 6.01]	0.067	+1.34	[0.41, 2.52]	0.025	./*
GLF	LEV	436	+1.57	[−0.38, 5.21]	0.097	0.00	[0.00, 0.00]	1.00	./.
GLF	MGR	436	+1.93	[0.89, 4.68]	0.044	+1.17	[0.32, 2.29]	0.007	*/**
GLF	MSA	436	+2.66	[0.28, 4.20]	0.058	+1.74	[0.69, 3.21]	0.009	./**
LEV	EGY	436	+1.39	[−1.49, 5.06]	0.143	+1.10	[0.23, 2.29]	0.045	./*
LEV	GLF	436	+2.73	[−0.33, 4.31]	0.095	0.00	[0.00, 0.00]	1.00	./.
LEV	MGR	436	+3.23	[0.23, 3.95]	0.039	+1.67	[0.67, 2.66]	0.008	*/**
LEV	MSA	436	+1.00	[−2.27, 4.32]	0.279	+1.33	[0.54, 2.53]	0.03	./*
MGR	EGY	436	+1.45	[−3.55, 4.27]	0.425	+1.12	[0.23, 2.29]	0.007	./**
MGR	GLF	436	+1.82	[−3.22, 4.05]	0.399	+1.34	[0.45, 2.52]	0.006	./**
MGR	LEV	436	+1.18	[−2.72, 5.06]	0.273	+2.36	[0.92, 3.67]	<0.001	./**
MGR	MSA	436	+1.00	[−2.25, 5.07]	0.214	+0.89	[0.23, 1.84]	0.034	./*
MSA	EGY	436	+0.01	[−1.89, 5.43]	0.168	+1.34	[0.78, 2.52]	0.030	./*
MSA	GLF	436	+0.19	[−2.51, 4.38]	0.292	+1.34	[0.46, 2.23]	0.028	./*
MSA	LEV	436	+0.88	[−1.17, 5.63]	0.095	+1.75	[0.68, 3.21]	0.003	./**
MSA	MGR	436	+1.63	[−1.38, 4.93]	0.251	0.00	[0.00, 0.00]	1.00	./.

Table A3. Paired bootstrap significance for off-diagonal dialect pairs on ArDQA-Narratives. We report the mean improvements Δ (proposed–baseline) in percentage points and Holm–Bonferroni-corrected one-sided p -values across the 20 off-diagonal pairs. In the Sig. column, the left marker refers to $\Delta F1$ and the right marker refers to ΔEM , using (**) for $p_{\text{holm}} < 0.01$, (*) for $p_{\text{holm}} < 0.05$, and (.) otherwise. C indicates the context variety and Q the question variety.

C	Q	<i>n</i>	$\Delta F1$	95% CI	p_{holm}	ΔEM	95% CI	p_{holm}	Sig.
EGY	GLF	364	+5.33	[1.51, 9.13]	0.061	+1.63	[0.55, 3.02]	0.028	./*
EGY	LEV	364	+1.50	[−1.27, 6.02]	0.190	+1.63	[0.92, 3.24]	0.019	./*
EGY	MGR	364	+6.50	[2.45, 10.66]	0.002	+5.00	[2.75, 7.14]	<0.001	**/**
EGY	MSA	364	+0.79	[−3.17, 5.25]	0.321	+1.15	[0.28, 2.20]	0.043	./*
GLF	EGY	364	+1.02	[−1.80, 5.26]	0.285	+1.62	[0.66, 3.12]	0.022	./*
GLF	LEV	364	+1.29	[−2.01, 4.60]	0.321	+1.62	[0.54, 3.10]	0.025	./*
GLF	MGR	364	+6.33	[2.19, 10.47]	0.001	+5.63	[3.30, 7.97]	<0.001	**/**
GLF	MSA	364	+0.62	[−3.18, 4.30]	0.386	+0.54	[0.00, 1.37]	0.124	./.
LEV	EGY	364	+1.41	[−1.64, 5.58]	0.245	+1.24	[0.28, 2.75]	0.048	./*
LEV	GLF	364	+1.59	[−1.85, 5.34]	0.276	+0.42	[0.00, 1.24]	0.098	./.
LEV	MGR	364	+7.38	[2.18, 10.45]	0.004	+3.44	[1.92, 5.49]	<0.001	**/**
LEV	MSA	364	+1.92	[0.32, 6.15]	0.039	+1.42	[0.34, 2.71]	0.046	*/*
MGR	EGY	364	+0.45	[−3.84, 4.46]	0.453	+1.11	[0.37, 2.17]	0.038	./*
MGR	GLF	364	+0.70	[−2.56, 4.63]	0.406	+2.89	[1.37, 4.95]	<0.001	./**
MGR	LEV	364	+1.07	[−2.83, 4.70]	0.315	+1.63	[0.69, 3.45]	0.012	./*
MGR	MSA	364	+0.33	[−5.15, 3.28]	0.673	+0.12	[0.00, 0.18]	0.873	./.
MSA	EGY	364	+0.73	[−3.61, 4.37]	0.433	+0.88	[0.00, 1.92]	0.046	./*
MSA	GLF	364	+2.15	[0.26, 6.35]	0.048	+1.89	[0.74, 3.35]	<0.001	*/**
MSA	LEV	364	+1.71	[−1.74, 5.62]	0.259	+2.99	[1.27, 4.93]	<0.001	./**
MSA	MGR	364	+12.13	[7.75, 16.52]	<0.001	+8.45	[5.77, 11.54]	<0.001	**/**

Appendix B. Extended Examples of Code-Mixed Question Generation

Source question (V_s) Maghrebi	Generated code-mixed question (V_m) Maghrebi + Egyptian	Lexical replacements ($V_s \rightarrow V_m$)	#Replacements / #Words	LSP (%)	SemSim	Decision / reason
شكون خلا نينتندو تتخذ إجراء رسمي بشأن الكراء؟ EN: “Who made Nintendo take official action regarding the rent?”	مين خلا نينتندو تتخذ إجراء رسمي بشأن الإيجار؟ EN: “Who made Nintendo take official action regarding the rent?”	مين (who) → شكون (who) الإيجار (rent) → الكراء (rent)	2 / 8	25.0	0.673	Accepted (LSP in [20%,60%] and SemSim $\geq$ 0.65)
شكون خلا نينتندو تتخذ إجراء رسمي بشأن الكراء؟ EN: “Who made Nintendo take official action regarding the rent?”	مين خلى نينتندو تتخذ إجراء رسمي بخصوص الإيجار؟ EN: “Who made Nintendo take official action regarding the rent?”	مين (who) → شكون (who) خلى (made) → خلا (made) بخصوص (regarding) → بشأن (regarding) الإيجار (rent) → الكراء (rent)	4 / 8	50.0	0.678	Accepted (LSP in [20%,60%] and SemSim $\geq$ 0.65)
شكون خلا نينتندو تتخذ إجراء رسمي بشأن الكراء؟ EN: “Who made Nintendo take official action regarding the rent?”	مين خلى نينتندو تاخذ إجراء رسمي بخصوص الإيجار؟ EN: “Who made Nintendo take official action regarding the rent?”	مين (who) → شكون (who) خلى (made) → خلا (made) بخصوص (regarding) → بشأن (regarding) الإيجار (rent) → الكراء (rent) تتخذ (take) → تاخذ (take)	5 / 8	62.5	0.699	Rejected (LSP > 60%)
شكون خلا نينتندو تتخذ إجراء رسمي بشأن الكراء؟ EN: “Who made Nintendo take official action regarding the rent?”	مين خلى نينتندو تاخذ إجراء رسمي بخصوص البيع؟ EN: “Who made Nintendo take official action regarding the sale?”	مين (who) → شكون (who) خلى (made) → خلا (made) بخصوص (regarding) → بشأن (regarding) تتخذ (take) → تاخذ (take) الكراء (rent) → البيع (sale)	5 / 8	62.5	0.601	Rejected (semantic drift, SemSim < 0.65)

Figure A1. Extended example of code-mixed question generation (Maghrebi $\rightarrow$ Egyptian) and semantic fidelity filtering using LSP and SemSim constraints. LSP denotes lexical substitution percentage, SemSim denotes semantic similarity. Arabic examples are accompanied by English glosses.

References

1. Althobaiti, M.J. Automatic Arabic dialect identification systems for written texts: A survey. *Int. J. Comput. Linguist. (IJCL)* **2020**, *11*, 61–89.
2. Habash, N.Y. *Introduction to Arabic Natural Language Processing*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2010.
3. Althobaiti, M.J. Creation of annotated country-level dialectal Arabic resources: An unsupervised approach. *Nat. Lang. Eng.* **2022**, *28*, 607–648.
4. Antoun, W.; Baly, F.; Hajj, H. AraBERT: Transformer-based Model for Arabic Language Understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*; European Language Resources Association: Paris, France, 2020; pp. 9–15.
5. Althobaiti, M.J. A Simple Yet Robust Algorithm for Automatic Extraction of Parallel Sentences: A Case Study on Arabic-English Wikipedia Articles. *IEEE Access* **2021**, *10*, 401–420.
6. Mozannar, H.; Maamary, E.; El Hajal, K.; Hajj, H. Neural Arabic Question Answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*; El-Hajj, W., Belguith, L.H., Bougares, F., Magdy, W., Zitouni, I., Tomeh, N., El-Haj, M., Zaghoulani, W., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 108–118. Available online: <https://aclanthology.org/W19-4612/> (accessed on 10 August 2025).
7. Ahmed, Z. ZeyadAhmed/Arabic-SQuADv2.0—Dataset. Hugging Face, 2022. Available online: <https://huggingface.co/datasets/ZeyadAhmed/Arabic-SQuADv2.0> (accessed on 10 August 2025).
8. Khalifa, M.; Hassan, H.; Fahmy, A. Zero-resource Multi-dialectal Arabic Natural Language Understanding. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 577–591. <https://doi.org/10.14569/IJACSA.2021.0120369>.
9. Althobaiti, M.J. An open-source dataset for arabic fine-grained emotion recognition of online content amid COVID-19 pandemic. *Data Brief* **2023**, *51*, 109745. <https://doi.org/https://doi.org/10.1016/j.dib.2023.109745>.
10. Joshi, A.; Dabre, R.; Kanojia, D.; Li, Z.; Zhan, H.; Haffari, G.; Dippold, D. Natural language processing for dialects of a language: A survey. *ACM Comput. Surv.* **2025**, *57*, 149.
11. Althobaiti, M.J. BERT-based Approach to Arabic Hate Speech and Offensive Language Detection in Twitter: Exploiting Emojis and Sentiment Analysis. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 972–980.
12. Althobaiti, M.J. ArDQA: A Parallel Multidomain Benchmark for Cross-Dialectal Arabic Question Answering. *Preprint* **2025**. <https://doi.org/10.21203/rs.3.rs-8007777/v1>.
13. Althobaiti, M.J. Emotion Recognition in Arabic: A BERT-Based Transfer Learning Approach Leveraging Semantic Information of Online Comments. *J. Theor. Appl. Inf. Technol.* **2023**, *101*, 3270–3282.
14. Altakrori, M.H.; Habash, N.; Freihat, A.; Samih, Y.; Chirkunov, K.; AbuOdeh, M.; Florian, R.; Lynn, T.; Nakov, P.; Aji, A.F. DialectalArabicMMLU: Benchmarking Dialectal Capabilities in Arabic and Multilingual Language Models. *arXiv* **2025**, arXiv:2510.27543.
15. Beidas, A.; Ghaddar, F.; Mohi, K.; Ahmad, I.; Abed, S. Cross-dialectal Arabic translation: Comparative analysis on large language models. *Front. Artif. Intell.* **2025**, *8*, 1661789.
16. Lewis, P.; Oguz, B.; Rinott, R.; Riedel, S.; Schwenk, H. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Jurafsky, D., Chai, J., Schluter, N., Tetreault, J., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 7315–7330. Available online: <https://aclanthology.org/2020.acl-main.653/> (accessed on 10 August 2025).
17. Luo, M.; Hashimoto, K.; Yavuz, S.; Liu, Z.; Baral, C.; Zhou, Y. Choose Your QA Model Wisely: A Systematic Study of Generative and Extractive Readers for Question Answering. In *Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge*; Das, R., Lewis, P., Min, S., Thai, J., Zaheer, M., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 7–22. <https://doi.org/10.18653/v1/2022.spanlp-1.2>.
18. Elkomy, M.; Sarhan, A. TCE at Qur'an QA 2023 Shared Task: Low Resource Enhanced Transformer-based Ensemble Approach for Qur'anic QA. In *Proceedings of ArabicNLP 2023*; Sawaf, H., El-Beltagy, S., Zaghoulani, W., Magdy, W., Abdelali, A., Tomeh, N., Abu Farha, I., Habash, N., Khalifa, S., Keleg, A., et al., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 728–742. <https://doi.org/10.18653/v1/2023.arabicnlp-1.81>.
19. Hammo, B.; Abu-Salem, H.; Lytinen, S.L.; Evens, M. QARAB: A Question answering system to support the Arabic language. In *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 1–11.
20. Trigui, O.; Belguith, L.H.; Rosso, P. DefArabicQA: Arabic definition question answering system. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 40–45.

21. Bakari, W.; Bellot, P.; Neji, M. Literature Review of Arabic Question-Answering: Modeling, Generation, Experimentation and Performance Analysis. In *Proceedings of the Flexible Query Answering Systems 2015*; Andreassen, T., Christiansen, H., Kacprzyk, J., Larsen, H., Pasi, G., Pivert, O., De Tré, G., Vila, M.A., Yazici, A., Zadrozny, S., Eds.; Springer Publishing Company, Incorporated: Cham, Switzerland, 2016; pp. 321–334.
22. Shaheen, M.; Ezzeldin, A.M. Arabic Question Answering: Systems, Resources, Tools, and Future Trends. *Arab. J. Sci. Eng.* **2014**, *39*, 4541–4564. <https://doi.org/10.1007/s13369-014-1062-2>.
23. Seo, M.; Kembhavi, A.; Farhadi, A.; Hajishirzi, H. Bidirectional Attention Flow for Machine Comprehension. In *Proceedings of the International Conference on Learning Representations*, Toulon, France, 24–26 April 2017.
24. Atef, A.; Mattar, B.; Sherif, S.; Elrefai, E.; Torki, M. AQAD: 17,000+ Arabic Questions for Machine Comprehension of Text. In *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6. <https://doi.org/10.1109/AICCSA50499.2020.9316526>.
25. Biltawi, M.M.; Awajan, A.; Tedmori, S. Arabic Span Extraction-based Reading Comprehension Benchmark (ASER) and Neural Baseline Models. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2023**, *22*, 127. <https://doi.org/10.1145/3579047>.
26. Pearce, K.; Zhan, T.; Komanduri, A.; Zhan, J. A comparative study of transformer-based language models on extractive question answering. *arXiv* **2021**, arXiv:2110.03142.
27. Clark, J.H.; Choi, E.; Collins, M.; Garrette, D.; Kwiatkowski, T.; Nikolaev, V.; Palomaki, J. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 454–470. https://doi.org/10.1162/tacl_a_00317.
28. Artetxe, M.; Ruder, S.; Yogatama, D. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Jurafsky, D., Chai, J., Schluter, N., Tetreault, J., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 4623–4637. Available online: <https://aclanthology.org/2020.acl-main.421/> (accessed on 10 August 2025).
29. Abdallah, A.; Kasem, M.; Abdalla, M.; Mahmoud, M.; Elkasaby, M.; Elbendary, Y.; Jatowt, A. ArabicaQA: A Comprehensive Dataset for Arabic Question Answering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 2049–2059. <https://doi.org/10.1145/3626772.3657889>.
30. Alrowili, S.; Vijay-Shanker, K. ArTrivia: Harvesting Arabic Wikipedia to Build A New Arabic Question Answering Dataset. In *Proceedings of ArabicNLP 2023*; Sawaf, H., El-Beltagy, S., Zaghouani, W., Magdy, W., Abdelali, A., Tomeh, N., Abu Farha, I., Habash, N., Khalifa, S., Keleg, A., et al., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 191–207. <https://doi.org/10.18653/v1/2023.arabnlp-1.17>.
31. Obeidat, R.; Al-Harbi, M.; Al-Ayyoub, M.; Alawneh, L. ArQuAD: An expert-annotated Arabic machine reading comprehension dataset. *Cogn. Comput.* **2024**, *16*, 984–1003.
32. Albilali, E.; Al-Twairish, N.; Hosny, M. Constructing Arabic Reading Comprehension Datasets: Arabic WikiReading and KaifLematha. *Lang. Resour. Eval.* **2022**, *56*, 729–764.
33. Antoun, W.; Baly, F.; Hajj, H. AraELECTRA: Pre-Training Text Discriminators for Arabic Language Understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 191–195.
34. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *Proceedings of the International Conference on Learning Representations*, Addis Ababa, Ethiopia, 26–30 April 2020; pp. 1–14.
35. Ahmed, Z.; Zeyada, M.; Amin, Y.; Gamal, D.; Hindy, H. Introducing Arabic-SQuADv2.0 for Effective Arabic Machine Reading Comprehension. *Int. J. Educ. Manag. Eng.* **2023**, *13*, 34.
36. HO, J.W.Y. Code-mixing: Linguistic form and socio-cultural meaning. *Int. J. Lang. Soc. Cult.* **2007**, *21*, 23–30.
37. Chandu, K.; Loginova, E.; Gupta, V.; van Genabith, J.; Neumann, G.; Chinnakotla, M.; Nyberg, E.; Black, A.W. Code-Mixed Question Answering Challenge: Crowd-sourcing Data and Techniques. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*; Aguilar, G., AlGhamdi, F., Soto, V., Solorio, T., Diab, M., Hirschberg, J., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 29–38. <https://doi.org/10.18653/v1/W18-3204>.
38. Dahou, A.; Dahou, A.H.; Cheragui, M.A.; Abdedaïem, A.; Al-Qaness, M.A.A.; Abd Elaziz, M.; Ewees, A.A.; Zheng, Z. A Survey on Dialect Arabic Processing and Analysis: Recent Advances and Future Trends. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2025**, *24*, 84. <https://doi.org/10.1145/3747290>.
39. Alwaneen, T.H.; Azmi, A.M.; Aboalsamh, H.A.; Cambria, E.; Hussain, A. Arabic question answering system: A survey. *Artif. Intell. Rev.* **2022**, *55*, 207–253.
40. Rizvi, M.S.Z.; Srinivasan, A.; Ganu, T.; Choudhury, M.; Sitaram, S. GCM: A Toolkit for Generating Synthetic Code-mixed Text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 205–211. <https://doi.org/10.18653/v1/2021.eacl-demos.24>.

41. Hamed, I.; Habash, N.; Abdennadher, S.; Vu, N.T. Investigating Lexical Replacements for Arabic-English Code-Switched Data Augmentation. In *Proceedings of the Sixth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2023)*; Ojha, A.K., Liu, C.h., Vylomova, E., Pirinen, F., Abbott, J., Washington, J., Oco, N., Malykh, V., Logacheva, V., Zhao, X., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 86–100. <https://doi.org/10.18653/v1/2023.loresmt-1.7>.
42. ElSabagh, A.A.; Azab, S.S.; Hefny, H.A. A comprehensive survey on Arabic text augmentation: Approaches, challenges, and applications. *Neural Comput. Appl.* **2025**, *37*, 7015–7048.
43. Sabty, C.; Omar, I.; Wasfalla, F.; Islam, M.; Abdennadher, S. Data augmentation techniques on Arabic data for named entity recognition. *Procedia Comput. Sci.* **2021**, *189*, 292–299.
44. Pratapa, A.; Bhat, G.; Choudhury, M.; Sitaram, S.; Dandapat, S.; Bali, K. Language Modeling for Code-Mixing: The Role of Linguistic Theory based Synthetic Data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; Gurevych, I., Miyao, Y., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 1543–1553. <https://doi.org/10.18653/v1/P18-1143>.
45. Qin, L.; Ni, M.; Zhang, Y.; Che, W. CoSDA-ML: Multi-Lingual Code-Switching Data Augmentation for Zero-Shot Cross-Lingual NLP. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*; Bessiere, C., Ed.; International Joint Conferences on Artificial Intelligence Organization: Yokohama, Japan, 2020; pp. 3853–3860. <https://doi.org/10.24963/ijcai.2020/533>.
46. Khanuja, S.; Dandapat, S.; Srinivasan, A.; Sitaram, S.; Choudhury, M. GLUECoS: An Evaluation Benchmark for Code-Switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 3575–3585.
47. Riley, P.; Shakeri, S.; Ammar, W.; Clark, J.H. TyDi QA-WANA: A Benchmark for Information-Seeking Question Answering in Languages of West Asia and North Africa. *arXiv* **2025**, arXiv:2507.17709.
48. Attardi, G. WikiExtractor. Available online: <https://github.com/attardi/wikiextractor> (accessed on 7 October 2024).
49. Vela, M.; Tan, L. Predicting Machine Translation Adequacy with Document Embeddings. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 402–410.
50. Liu, X.; Ren, S.; Sun, M.; Liu, Y.; Su, Q. Predicting Sentence Translation Adequacy for Document-Level Neural Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4250–4260.
51. Moon, J.; Cho, H.; Park, E.L. Revisiting Round-trip Translation for Quality Estimation. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*; European Association for Machine Translation: Lisboa, Portugal, 2020; pp. 91–104.
52. Ye, Y.; Simpson, E.; Rodriguez, R.S. Using similarity to evaluate factual consistency in summaries. *arXiv* **2024**, arXiv:2409.15090.
53. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 3982–3992.
54. Jalili Sabet, M.; Dufter, P.; Yvon, F.; Schütze, H. SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 1627–1643. Available online: <https://aclanthology.org/2020.findings-emnlp.147/> (accessed on 1 December 2025).
55. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 2383–2392. <https://doi.org/10.18653/v1/D16-1264>.
56. Buckwalter, T. Buckwalter Arabic Transliteration. Qamus Arabic Dictionaries and Tools. 2002. Available online: <http://www.qamus.org/transliteration.htm> (accessed on 1 December 2025).
57. Althobaiti, M.; Kruschwitz, U.; Poesio, M. AraNLP: A Java-based Library for the Processing of Arabic Text. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*; Association for Computational Linguistics: Reykjavik, Iceland, 2014; pp. 4134–4138.
58. Reimers, N.; Gurevych, I. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 4512–4525. <https://doi.org/10.18653/v1/2020.emnlp-main.365>.
59. Reimers, N.; Team, S.T. Paraphrase-Multilingual-MiniLM-L12-v2. Hugging Face Model Card, 2021. Available online: <https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2> (accessed on 25 June 2025).
60. Costa-Jussà, M.R.; Cross, J.; Çelebi, O.; Elbayad, M.; Heafield, K.; Heffernan, K.; Kalbassi, E.; Lam, J.; Licht, D.; Maillard, J.; et al. No language left behind: Scaling human-centered machine translation. *arXiv* **2022**, arXiv:2207.04672. <https://doi.org/10.48550/arXiv.2207.04672>.

61. Team, N. Scaling neural machine translation to 200 languages. *Nature* **2024**, *630*, 841–846. <https://doi.org/10.1038/s41586-024-07335-x>.
62. AI, M. Facebook/nllb-200-distilled-600M. Hugging Face Model Card, 2022. License: CC-BY-NC 4.0. Available online: <https://huggingface.co/facebook/nllb-200-distilled-600M> (accessed on 23 May 2025)
63. Abdul-Mageed, M.; Elmadany, A.; Nagoudi, E.M.B. ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 7088–7105. <https://doi.org/10.18653/v1/2021.acl-long.551>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.