Tarikhi: Arabic Temporal Information Extraction From Arabic Historical Documents

Qusay Abdo[1], **Serin Atiani**[1,*], **Tariq Sarayji** [1],**Adnan Saeed**[1]
[1]Princess Sumaya University for Technology, Jordan
[*]Corresponding author: s.atiani@psut.edu.jo

Abstract

Arabic historical books and archival materials contain rich accounts of political, social, and cultural events, yet they remain largely underutilized computationally due to the scarcity of dedicated Arabic information extraction tools. The challenge is amplified in long-form, scanned historical documents, where optical character recognition noise, orthographic variation, and complex narrative structures complicate automatic processing. In this paper, we present Tarikhi, a retrieval-augmented generation framework for structured temporal event extraction from Arabic scanned books. The proposed pipeline integrates high-accuracy optical character recognition, chunking-based processing for long-document handling, Arabic named entity recognition, span refinement, and a retrieval-enhanced attribute extraction module that identifies event dates, locations, and descriptive summaries. Extracted events are consolidated and linked using semantic and temporal similarity measures, and linked through relation classification to construct structured temporal events. Evaluation on a selected part of modern Arabic historical books demonstrates the feasibility of temporal event extraction from long-form Arabic texts, achieving a 75.3% F1-score under dual human verification. Tarikhi represents a step toward scalable temporal knowledge construction for Arabic digital humanities resources.

Keywords: Temporal Event Extraction, Arabic Natural Language Processing, Retrieval Augmented Generation, Long-form Document Processing

1. Introduction

The rapid digitization of historical Arabic archives has generated vast information repositories, creating new opportunities for computational analysis of cultural and historical knowledge. Automated information extraction systems are increasingly needed to transform unstructured historical documents into structured representations that support search, analytics, and scholarly research. Event extraction is the task of identifying structured descriptions of events from unstructured text, it has emerged as a fundamental component of modern information extraction pipelines and is widely applied across domains such as historical analysis, knowledge base construction, and digital humanities research (Zhang and Han, 2025; Huang et al., 2024). However, extracting temporal events from historical sources remains challenging due to linguistic ambiguity, narrative complexity, and the need to reason across long textual contexts.

Recent advances in large language models and retrieval-augmented generation (RAG) architectures have demonstrated strong capabilities in contextual reasoning and knowledge-intensive tasks by integrating retrieval mechanisms with generative models (Krasadakis et al., 2024). In event temporal relation extraction and related tasks, retrieval-augmented approaches have shown promise in improving performance by incorporating external knowledge and contextual evidence (Zhang et al., 2024). The above-mentioned research predominantly focuses on English and a limited set of high-resource languages, leaving a significant gap in tools designed for low-resource languages, specifically for Arabic historical materials.

Arabic historical resources present unique challenges for computational analysis and event extraction due to its rich morphology, orthographic variability, and limited availability of annotated resources, particularly for specialized tasks such as temporal event extraction (Darwish et al., 2021; Alayba, 2025). Existing studies highlight the scarcity of datasets and models tailored to Arabic event extraction and emphasize the need for domain-specific approaches that address linguistic and resource limitations (Alayba, 2025; Aljabari et al., 2024a). Previous work has explored information extraction from Arabic manuscripts, focusing mainly on classification or basic information extraction rather than comprehensive temporal event modeling (Bashir et al., 2023). Consequently, there is a pressing need for solutions that can efficiently extract structured historical events from Arabic resources while leveraging modern retrieval and language modeling techniques.

This paper proposes a retrieval-augmented generation (RAG) framework designed specifically for temporal event extraction from Arabic long-form historical sources. The proposed system ingests textual resources, including books, and archives in PDF format, transcribes them into text and integrates these resources into a retrieval-driven pipeline that enhances contextual understanding and enables efficient extraction of temporal historical events. The paper is organized as follows: Sec-

Proceedings of the 2nd International Workshop on Nakba Narratives as Language Resources (Nakba-NLP 2026) @ LREC 2026, pages 60–69
11 May 2026.

tion 2 reviews related work, section 3 outlines the methodology, section 4 demonstrates system results at different components of the system and system evaluation, sections 5 discussion and section 6 outlining limitations of the work.

2. Related Work

Temporal Information Extraction (TIE) is the process of identifying temporal expressions in natural language text and extracting the temporal relations between events and time expressions. The date and time associated with a temporal expression constitute essential identifiers of a temporal event, enabling structured temporal reasoning over text. Identifying temporal relations between events (e.g., *before*, *after*, *includes*) and anchoring them to temporal expressions facilitates the construction of timelines in which events are organized according to their temporal dependencies (Leeuwenberg and Moens, 2018).

Early research on temporal processing was strongly influenced by the TimeML specification (Pustejovsky et al., 2003), which introduced a standardized annotation schema for events, temporal expressions (TIMEX3), and temporal links (TLINKs). Shared tasks such as TempEval (Verhagen et al., 2007; UzZaman et al., 2013) and Clinical TempEval (Bethard et al., 2015) played a pivotal role in establishing benchmarks for temporal expression extraction, event detection, temporal relation classification, and temporal anchoring. Although temporal expression extraction has sometimes been framed as a sub-task of Named Entity Recognition (NER), it has evolved into a distinct research trajectory due to the additional requirements of normalization, temporal reasoning, and cross-sentence relation modeling (Wong et al., 2005). Subsequent research has advanced temporal expression extraction (Filannino and Nenadić, 2015), temporal annotation frameworks and corpora construction (D'Souza and Ng, 2014; Mostafazadeh et al., 2016), and temporal disambiguation and reasoning techniques, including recent self-supervised and neural approaches (Wenzel and Jatowt, 2023; Cai et al., 2023).

The rise of deep learning has significantly improved performance on temporal tasks. Transformer-based language models such as BERT have been adapted for event detection and temporal relation classification (Devlin et al., 2019; Ning et al., 2019). Recent work has explored graph-based neural models and global inference mechanisms to improve temporal extraction across documents (Leeuwenberg and Moens, 2018). Moreover, large-scale pretraining and prompt-based learning have improved temporal extraction in low-resource and domain-specific scenarios (Cai et al., 2023).

TIE for Arabic has been mostly studied as a subset of event extraction in general, and it has been addressed by studies that have looked more in general into named entity recognition, with few exceptions (Saleh et al., 2011; Zaraket, 2012; Lhioui et al., 2017). Several studies have focused on developing dedicated corpora and annotation schemes. ARA-Timex introduced two annotated datasets compiled from WikiNews and news articles to support temporal expression extraction and normalization in Arabic (Boudaa et al., 2018). AraTimeBank presented a revised Arabic TimeML dataset that addressed limitations in earlier annotations and enriched the corpus to better support TIE tasks (Haffar et al., 2020). More recently, the Wojood project has contributed substantially to Arabic event and relation extraction. Wojood$^{\text{Hadath}}$ introduced an enhanced version of the Wojood corpus enriched with event-argument annotations (Aljabari et al., 2024b), while Wojood$^{\text{Relation}}$ further extended the dataset with annotated event-event relations, supporting structured temporal and causal modeling (Aljabari et al., 2025). These resources collectively advance the feasibility of temporal event extraction pipelines in Arabic.

In parallel, transformer-based Arabic language models have significantly improved performance across core NLP tasks, including NER, which forms a foundational component of temporal expression and event extraction. AraBERT demonstrated competitive results on Arabic NER and other downstream tasks without explicitly targeting temporal extraction (Antoun et al., 2020). JABER and SABER BERT models further improved upon prior transformer-based models through large-scale pretraining tailored to Arabic linguistic characteristics (Ghaddar et al., 2022). The Wojood benchmark established strong performance on nested NER using transformer-based architectures, providing an essential stepping stone for fine-grained temporal and event extraction tasks (Jarrar et al., 2022).

Despite the substantial strides made in event extraction and NER in Arabic, temporal information extraction and temporal information linking require further exploration and development, specifically in long form Arabic texts like books. To address this gap, we introduce Tarikhi, a fully automated pipeline for structured temporal information extraction from scanned Arabic historical documents.

3. Methodology

In this section, we describe the methodology used to implement **Tarikhi**. The proposed pipeline transforms documents into structured event records containing event descriptions, dates,

and locations, while also constructing semantic links between events, such as SAME_AS and RELATED_TO relations. Figure 1 provides a high-level illustration of the pipeline components. Scanned long-form resources are ingested and using OCR are transcribed into text. The text is preprocessed and chunked. Temporal events are identified and refined using Silma-9B (AI, 2024). RAG-based retrieval is employed to extract information from the document and then the extracted events are consolidated and linked using SAME_AS and RELATED_TO relations.

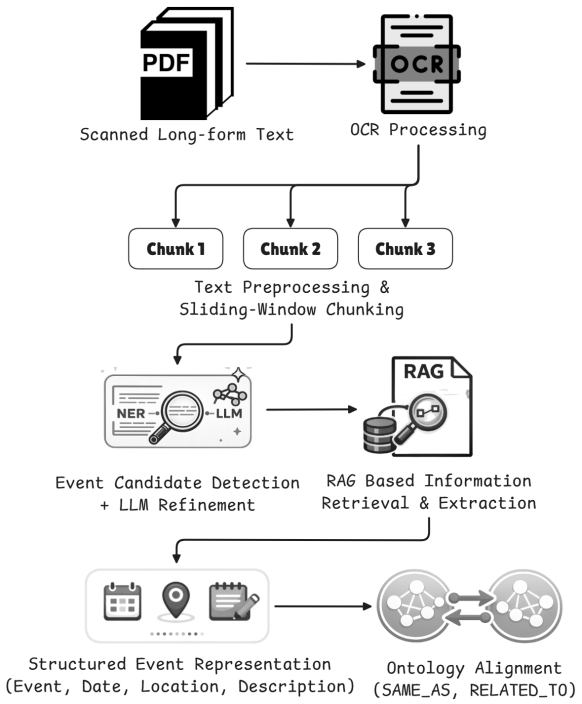

Figure 1: High-Level Pipeline

3.1. Data Sources

To run and evaluate Tarikhi, we used two books, the Arabic translation of the Hundred Years' War on Palestine by Rashid Khalidi, translation by Amer Shaikhouni first edition 2021 (Khalidi, 2021), and The 1967 War: Secrets and Mysteries by Amin Howeidi, original text in Arabic, first edition 2006 (Amin Huwaidi, 2006). Both books were selected for their dense references to political events, dates, actors, and locations, which make them suitable for evaluating temporal event extraction.

3.2. Text Recognition and Cleaning

We evaluated multiple OCR approaches during development, including the standard OCR engine Surya (Contributors, 2024), LLM-based OCR models such as Qari (Wasfy et al., 2025) and DeepSeek OCR (Wei et al., 2025), and a hybrid voting-based approach that combined the outputs of the three models by selecting word tokens through majority voting, while the Surya output was used as a fallback when all three models produced different outputs. For the final pipeline configuration, we selected the Surya OCR framework (Contributors, 2024) after conducting a small evaluation by manually transcribing 20 randomly selected pages, with 10 pages sampled from each of the two source books. Each OCR model was then run on the same pages and its generated output was compared against the manually written ground truth by computing alignment-based character-level and word-level similarity scores using `difflib.SequenceMatcher`, Table 1 shows the average results across the evaluated pages.

Model	Character Accuracy	Word Accuracy
DeepSeek	0.7449	0.8718
Qari	0.6760	0.9009
Surya	0.8592	0.9366
Hybrid	0.7746	0.8829

Table 1: Average OCR performance comparison across evaluated models.

Surya OCR operates using a two-stage architecture: a text line detection model followed by a text recognition model. The detection model identifies line-level text regions on each page in the given document and returns cropped line image segments that are stored locally. In the second stage, each detected line image is processed individually by the text recognition model and then concatenated to produce the Arabic text transcription. A text cleaning stage was applied prior to chunking to enhance processing quality. The cleaning method removes characters outside the Arabic letter range with the exceptions of digits and a small set of punctuation symbols (periods, hyphens, and slashes) to preserve dates and references.

3.3. Chunking Strategy and Embeddings Vector Space

To handle long documents within the context limitations of large language models, we adopted a tokenizer-based chunking strategy.

Each document is first tokenized using the same tokenizer associated with the target LLM; the exact one we used is silma-9b, and the token sequence is segmented into fixed-size chunks of 256 tokens with an overlap of 50 tokens between consecutive windows.

For semantic retrieval, each text chunk is encoded into a dense vector representation using the **intfloat/multilingual-E5-large** sentence em-

bedding model (Wang et al., 2024), embeddings were computed at the chunk level.

Each chunk is stored as a structured chunk object containing:

- The chunk text.
- Its embedding vector.
- Metadata fields for subsequent event extraction outputs.

Consequently, all chunk embeddings are indexed using a **FAISS** index, and similarity search is performed in the embedding space using top-k nearest neighbor queries. In the final configuration, we set $k = 6$ to provide sufficient context to the LLM without exceeding its context window.

3.4. Named Entity Recognition and Event Detection

For each stored chunk object, an event recognition step is performed in two stages. First, the events are triggered using **Sinatools**, a research-based open-source Python library used for various NLP tasks such as flat and nested Named Entity Recognition (NER), fully-flagged Word Sense Disambiguation (WSD), Semantic Relatedness, Synonymy Extractions and Evaluation, Lemmatization, Part-of-speech Tagging, Root Tagging, and additional helper utilities (Hammouda et al., 2024). In this study, we use their pre-trained Wojood NER model to trigger the events from the predicted tag sequence by merging spans that start with `B-EVENT`, which describe the first word of the event, and continue with `I-EVENT` which marks the remaining of the event name unless the event consists of a single word. This NER layer often yields incomplete or generic event names in Arabic.

To improve event name quality, we added an LLM assisted refinement layer on top of NER spans. For each extracted event span, we build a short local context window consisting of 10 tokens before the span and 20 tokens after it, which is then provided to the LLM as context. The LLM **SILMA-9B-Instruct-v1.0** (AI, 2024) takes the extracted event name, compares it with the context window, and returns a normalized, more representative event name. The refinement step uses deterministic decoding with temprature set to 0.0.

3.5. Event Disambiguation and Attribute Extraction

For each detected event, an event disambiguation stage is applied to determine the event's date, location, and description. This stage uses retrieval-augmented generation (RAG) with the LLM SILMA-9B.

Each event is encoded using the same multilingual embedding model used for chunk indexing, the event vector is used to query the FAISS vector index to retrieve the top-6 most semantically similar chunks. These retrieved chunks are then concatenated with the source chunk from which the event was originally triggered and provided as context for the LLM.

The extracted context is then provided to the LLM to extract specific attributes through prompting. Separate prompts and calls have been used for date, location, and description extraction, prompts enforce strict JSON-only responses with predefined schemas such as `{"date": "dd-mm-yyyy"}` and `{"location": "<place>"}`, and explicitly instruct the model to rely only on the provided text context. For date extraction, if no date is found, the model outputs `xx-xx-xxxx`, if only part of the date is available, missing components are replaced with `xx`. For location extraction, the output is left blank when no explicit location is found in the provided context.

The output of this stage is a structured event record in which each event is associated with its date, location, and description to be used and processed for event-linking.

3.6. Event Linking

To form links between the extracted events, we defined two edge types: `SAME_AS`, `RELATED_TO`. Linking is performed in two stages: (1) merge duplicate mentions into event clusters via `SAME_AS`, then (2) connect clusters with `RELATED_TO`.

3.6.1. `SAME_AS` clustering.

Given two event mentions e_i and e_j, we compute a composite similarity score combining:

- **Semantic similarity** s_{sem}: we concatenate the event name and description, encode the text using `multilingual-e5-large`, and compute cosine similarity between the resulting embeddings.
- **Temporal similarity** s_{temp}: when both mentions have valid dates, we compute an exponential decay over the absolute day difference,

$$s_{temp} = \exp\left(-\frac{\Delta t}{\tau}\right),$$

  where $\tau = 180$ days.
- **Location similarity** s_{loc}: computed by fuzzy location name matching.

The final score is computed using a weighted combination of the available scores

$$s = \frac{w_{emb}s_{sem} + w_{date}s_{temp} + w_{loc}s_{loc}}{w_{emb} + w_{date} + w_{loc}}$$

where $w_{emb} = 0.6$, $w_{date} = 0.3$, and $w_loc = 0.1$. Two events are merged when their score is at least 0.9, This was a conservative threshold that was set based on repeated observations, to only merge events when there is a a very strong evidence of their equivalence. When one of the scores is missing due to unavailable date or location, its weight is omitted from both the numerator and denominator of the weighted score. After clustering, each cluster node, represents one event that embodies all SAME_AS events. The event represented by the cluster node is summarized by a centroid embedding computed as the mean of member embeddings, a representative date defined as the median of available dates, a representative location defined as the most frequently mentioned location.

3.6.2. RELATED_TO Linking

RELATED_TO event edges are generated at the cluster level which differs from the SAME_AS relations generation that rely on the original extracted events. For each source cluster c_i, we compute cosine similarity between its event centroid embedding and all other cluster centroids. RELATED_TO candidate events are ranked by semantic similarity, and evaluate them in descending order.

For a candidate pair (c_i, c_j) we compute:

- **Semantic score** s_{sem}: cosine similarity of centroid embeddings.
- **Temporal proximity** s_{temp}: the same exponential decay function applied to the absolute difference between cluster median dates (set to 0 if either date is missing), we use a decay constant of $\tau = 365$ days.
- **Location score** s_{loc}: the same fuzzy matching technique between the most frequently mentioned locations of the two clusters.

These scores are combined as:

$$s = \alpha s_{sem} + \beta s_{temp} + \gamma s_{loc}.$$

where $\alpha = 0.7$, $\beta = 0.2$, and $\gamma = 0.1$. A RELATED_TO edge is added if $s \geq 0.75$, we implemented a lower threshold to allow graph to capture semantically connected events without requiring same-identity, we also increased the threshold for semantic similarity in RELATED_TO relations since related events are often connected through narrative context even when their dates or locations differ.

3.7. System Evaluation

We selected three chapters from the two resources; one chapter, 41 pages long, from The Hundred Years War on Palestine, and two chapters from The 1967 War, 16 and 15 pages respectively . The system ingested each of the chapters considering each of them as an independent source. Each of the chapters was processed, running all the elements of the system pipeline described above. Two of the researchers independently read the selected chapters before seeing the extracted events by the system. The two researchers created their own list of temporal events from the selected chapters. Each of the researcher matched the lists of human identified temporal events with the system extracted temporal events. The intersection of the two researchers' sets of identified correct temporal events was considered to be the correct set of extracted temporal events by the system, true positives. The set of commonly identified events by the researchers that were not extracted by the system were considered the missed set of extracted events or the false negatives. The events that were extracted by the system and were not identified by the researchers was deemed mistaken events, and are considered the false positives.

4. Results

Processing both resources Tarikhi extracted 944 events from 1077 text chunks, following SAME_AS clustering these events were consolidated into 341 unique events nodes, with 601 RELATED_TO edges.

4.1. System Results

4.1.1. LLM refinement

A critical stage in our system was the LLM refinement step. This stage takes on the identified temporal events from Wojood and refines the results using an LLM. This refinement layer improves incomplete or generic NER spans by providing the LLM of an input of the surrounding text

Raw NER Span	LLM-Normalized Event
اجتماع	اجتماع للجنة الكنيست لشؤون الدفاع والخارجية
تبادلٍ	تبادلٍ عنيف لإطلاق النار عبر الحدود
مؤتمر صحفيٍ	مؤتمر صحفيٍ في فندق الكومودور في بيروت لبنان

Table 2: Examples of LLM-based refinement of NER event spans

Table 2 provides examples of LLM refinement: "meeting" is refined using the context it was extracted from to "meeting of Knesset committee for foreign relations", "exchange" is refined to "heavy exchange of fire across the border", "press confer-

ence ” is refined to ”press conference in the Commodore hotel in Beirut Lebanon”

4.1.2. Structured Events

The output of the system is a set of structured events that are organized into four main attributes: event, date, location, and description.

The following examples show the final structured event representation produced by Tarikhi.

Example 1 Event: معركة ميسلون
Date: 23-07-1920
Location: Syria
Description:
معركة ميسلون كانت معركة حاسمة بين القوات العربية والفرنسية في يوليو 1920، وانتهت بانتصار الفرنسيين

The example shows the extracted event Maysalun battle, with its date and location, the description extracted is The Battle of Maysalun was a decisive battle between Arab and French forces in July 1920, ending in a French victory.

Example 2 Event: حرب السويس
Date: 29-10-1956
Location: Egypt
Description:
حرب السويس هي عملية عسكرية بدأت في 29 أكتوبر 1956 بمشاركة إسرائيل وفرنسا وبريطانيا ضد مصر عقب تأميم قناة السويس. انتهت العمليات العسكرية خلال أسابيع قليلة تحت ضغط دولي، وأعادت تشكيل موازين القوى الإقليمية.

This example's event is Suez war or as it is known in english literature, Suez crisis, with the extracted location and date. The description provided by the system is The Suez War was a military operation that began on October 29, 1956, involving Israel, France, and Britain against Egypt following the nationalization of the Suez Canal. The military operations ended within a few weeks under international pressure, and reshaped the regional balance of power.

4.1.3. Event Clusters

In some cases, the process of identifying repeated extracted events is essential to reduce the cluttering of outputted events drastically improving the functionality of the system. These events were associated to each other using the SAME_AS link creating clusters of events that represent one canonical event

Cluster A (Size = 18) Canonical Event: الثورة العربية في فلسطين 1936-1939
Representative Variants:

- ثورة عربية كبري في فلسطين
- الثورة القومية
- الاضراب العام سنة 1936
- الثورة الكبري في 1936-1939
- الانتفاضة 1936-1939
- ثورة 1936-1939

This example shows a cluster of 18 extracted events that all refer to the canonical event, the Arab Revolt in Palestine (1936-1939). The cluster includes variants such as Major Arab Revolt in Palestine, The Great Revolt (1936-1939), and The 1936-1939 uprising.

Cluster B (Size = 42) Canonical Event: حرب لبنان 1982
Representative Variants:

- الغزو الاسرائيلي للبنان
- حصار بيروت
- اجتياح لبنان
- انسحاب منظمة التحرير الفلسطينية من بيروت
- غزو لبنان سنة 1982

This example show a cluster of 42 extracted events that all represent the canonical event Lebanon war 1982. the cluster has variants like Israeli invasion of lebanon, siege of Beirut , invasion of Lebanon, the withdrwal of Palestine Liberation Organization from Beirut, invasion of lebanon 1982

4.1.4. RELATED_TO Linking Examples

Another important element in the system is identifying related events and linking them to each other. Events are identified as referring to the same event were first clustered and represented as a single node, for each pair of candidate nodes, semantic similarity, temporal proximity, and location similarity were computed and combined into a weighted relationship score. Event nodes with relationship score greater than 75% were considered to be related events using the `RELATED_TO` relation.

The example in Table 3 shows the results of RELATED_TO events linking step for one event node 1948 war where it is linked to the following events:The massive Zionist attack, the fall of Palestinian towns and villages, the political and milliary victory of Zionism, Displacement from their homeland Secret agreement to prevent the dispatching of the Arabic army, The decision to partition Palestine with relationship scores of 0.93, 0.91, 0.88, 0.88, 0.87, 0.87 respectively.

Event: 1948 حرب	
Related event	**Score**
الهجوم الصهيوني الواسع	0.93
سقوط المدن والقرى الفلسطينية	0.91
انتصار الصهيونية العسكري والسياسي	0.88
النزوح عن وطنهم	0.88
اتفاقية سرية لمنع ارسال الجيش العربي عبر	0.87
قرار تقسيم فلسطين	0.87

Table 3: 1948 War Related Events

4.2. Evaluation

We selected three chapters from our resources to evaluate the system's output, one chapter (41 pages) from The Hundred Years War book and two chapters (31 pages total) from the 1967 War book. The evaluation method we used is designed to evaluate the performance of the system as a whole. The system processed the chapters independently as separate sources. The system extracted 156 initial temporal events. Out of these initially extracted events 95 canonical events were identified. Two researchers used cross over verification to identify correctly extracted temporal events vs mistaken events, which included non-events like names of individuals, or organizations, correct events associated with wrong location, or wrong date. Based on the researchers' assessment 75 temporal events were identified as correct events resulting with 78.8% recall, 72.1% precision and 75.3% F1-score.

5. Discussion

The results show that a layered RAG-based pipeline can feasibly extract structured temporal events from long-form Arabic historical texts, a task that has not been addressed before in an end-to-end manner. The 75.3% F1-score, while leaving room for improvement, is encouraging considering the difficulty of processing scanned Arabic books where OCR noise, orthographic variation, and the absence of diacritics all add to the challenges of event identification and attribute extraction. A key design choice in Tarikhi is the use of NER-based event triggering followed by LLM refinement, rather than relying on either component alone. The NER model provides structured, reproducible event spans, but often produces incomplete or overly generic names, as seen in Table 2. The LLM refinement layer addresses this by grounding those spans in their surrounding context, producing more descriptive and representative event names. This two-stage approach strikes a balance between the reliability of a trained NER model and the contextual reasoning of an LLM, though it also means that events missed entirely by the NER layer cannot be recovered downstream, since the LLM only refines what the NER has already detected.

The SAME_AS clustering proved effective in handling the repetitive nature of historical narratives, where the same event is often mentioned multiple times in different surface forms. The reduction of 944 raw event mentions to 341 canonical nodes illustrates just how much redundancy exists in long-form historical texts and highlights why consolidation matters for producing usable structured output. The RELATED_TO linking adds further value by surfacing thematic connections between events that would otherwise require extensive manual reading to uncover. That said, relying on embedding-based similarity and fuzzy string matching for linking means that events connected through implicit or causal reasoning rather than surface-level similarity may go undetected. The fixed chunking strategy with 256-token windows and 50-token overlaps worked well for the selected resources, but historical texts with varying event density across pages could benefit from adaptive chunking strategies that take narrative structure into account rather than relying on fixed token counts.

Arabic-specific challenges remain a real factor in system performance. The rich morphology of Arabic means that the same event can appear in many inflected forms, which complicates both NER detection and embedding-based similarity. OCR errors introduce additional noise that carries through every downstream component, and while the text cleaning step helps reduce some of this, it cannot recover information that was lost during recognition. These challenges point to the ongoing need for investment in Arabic-specific tools that can handle the variability found in historical documents.

6. Limitations and future work

Tarikhi was evaluated on only two Arabic historical books, both centered on modern political history in the Middle East, which limits the generalizability of the reported results to other types of historical texts that are less date and location heavy. The evaluation methodology, while practical given the absence of gold-standard annotated corpora for long-form Arabic temporal event extraction, relies on dual human verification by two researchers, introducing potential subjectivity in what constitutes a correct event, the granularity of a temporal event, an acceptable date, or a valid location. Furthermore, the evaluation assessed the complete pipeline output rather than the performance of individual components, making it difficult to isolate specific bottlenecks or failure points within

the pipeline; evaluating these components individually requires dedicated long-form annotated corpora tailored to each task, such as book-level NER annotations for event triggering assessment and ground-truth event attributes for measuring RAG-based date, location, and description extraction accuracy, neither of which currently exist for Arabic historical texts. Future work should expand the resource base to include a wider variety of Arabic historical texts spanning different eras, and narrative styles, and should develop dedicated annotated benchmark corpora that enable both component-level and end-to-end evaluation.

7. Broader Impact

Tarikhi is a system that takes on scanned Arabic resources and enables structured temporal event extraction from long-form texts like books and lengthy articles. The proposed solution enables the creation of curated knowledge bases from selected resources and provides a systematic way to access and query historical temporal data. More importantly, Tarikhi has the potential to unlock the wealth of knowledge that lives in undigitized Arabic-language books, making it computationally accessible. This accessibility opens the door to considering alternative historical narratives and including perspectives that have been largely overlooked in the digital age, contributing to a more representative and diverse digital humanities landscape.

8. Acknowledgements

The authors would like to thank Dr. Fadi Zaraket for proposing the initial idea that inspired this work and for his valuable advice and informative discussions and suggestions.

References

SILMA AI. 2024. Silma-9b-instruct-v1.0. https://huggingface.co/silma-ai/SILMA-9B-Instruct-v1.0. Large Language Model.

Abdulaziz M. Alayba. 2025. Arabic natural language processing (nlp): A comprehensive review of challenges, techniques, and emerging trends. *Computers*, 14(11).

Alaa Aljabari, Lina Duaibes, Mustafa Jarrar, and Mohammed Khalilia. 2024a. Event-arguments extraction corpus and modeling using bert for arabic.

Alaa Aljabari, Lina Duaibes, Mustafa Jarrar, and Mohammed Khalilia. 2024b. Event-arguments extraction corpus and modeling using bert for arabic. *arXiv preprint arXiv:2407.21153*.

Alaa Aljabari, Mohammed Khalilia, and Mustafa Jarrar. 2025. Wojood[Relations]: Arabic relation extraction corpus and modeling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34342–34360.

هويدي، أمين Amin Huwaidi. 2006. *Harb 1967: Asrar wa Khabaya* , حرب 1967 : أسرار و خبايا. Al-Maktab Al-Misri Al-Hadith, المكتب المصري الحديث, Cairo, Egypt.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. 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 (OSACT 2020)*, pages 9–15. OSACT 2020.

M. H. Bashir, A. M. Azmi, H. Nawaz, et al. 2023. Arabic natural language processing for qur'anic research: A systematic review. *Artificial Intelligence Review*, 56:6801–6854.

Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. SemEval-2015 task 6: Clinical TempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814.

Tarik Boudaa, Mohamed El Marouani, and Nourddine Enneya. 2018. Arabic temporal expression tagging and normalization. In *Big Data, Cloud and Applications (BDCA 2018)*, volume 872 of *Communications in Computer and Information Science*, pages 271–284. Springer.

Bibo Cai, Xiao Ding, Zhouhao Sun, Bing Qin, Ting Liu, Baojun Wang, and Lifeng Shang. 2023. Self-supervised logic induction for explainable fuzzy temporal commonsense reasoning. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Surya OCR Contributors. 2024. Surya ocr: Multilingual document ocr toolkit. https://github.com/datalab-to/surya. Open-source OCR toolkit supporting 90+ languages.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj,

Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab world. *Commun. ACM*, 64(4):72–81.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Jennifer D'Souza and Vincent Ng. 2014. Annotating inter-sentence temporal relations in clinical notes. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2758–2765.

Michele Filannino and Goran Nenadić. 2015. Temporal expression extraction with extensive feature type selection and a posteriori label adjustment. *Data & Knowledge Engineering*, 100:19–33.

Abbas Ghaddar, Yimeng Wu, Ahmad Rashid, Khalil Bibi, Mehdi Rezagholizadeh, Chao Xing, Yasheng Wang, Duan Xinyu, Zhefeng Wang, Baoxing Huai, Xin Jiang, Qun Liu, and Philippe Langlais. 2022. Jaber and saber: Junior and senior arabic bert. *arXiv preprint arXiv:2112.04329*.

Nafaa Haffar, Emna Hkiri, and Mounir Zrigui. 2020. Enrichment of arabic timeml corpus. In *Computational Collective Intelligence (ICCCI 2020)*, volume 12496 of *Lecture Notes in Computer Science*, pages 233–244. Springer.

Tymaa Hammouda, Mustafa Jarrar, and Mohammed Khalilia. 2024. Sinatools: Open source toolkit for arabic natural language processing. *Procedia Computer Science*, 244:388–396. 6th International Conference on AI in Computational Linguistics.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Premkumar Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. 2024. Textee: Benchmark, reevaluation, reflections, and future challenges in event extraction.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested arabic named entity corpus and recognition using bert. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC 2022)*, pages 3626–3636.

Rashid Khalidi. 2021. . , . Arabic translation of "The Hundred Years' War on Palestine".

Panteleimon Krasadakis, Evangelos Sakkopoulos, and Vassilios S. Verykios. 2024. A survey on challenges and advances in natural language processing with a focus on legal informatics and low-resource languages. *Electronics*, 13(3).

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Temporal information extraction by predicting relative time-lines. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1237–1246.

Chahira Lhioui, Anis Zouaghi, and Mounir Zrigui. 2017. A rule-based approach for arabic temporal expression extraction. In *2017 International Conference on Engineering MIS (ICEMIS)*, pages 1–6.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. 2016. CaTeRS: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the Fourth Workshop on Events*, pages 51–61.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. 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)*, pages 6203–6209.

James Pustejovsky, José M. Castaño, Robert Ingria, Roser Saurí, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 2003. Timeml: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering*.

Iman Saleh, Lamia Tounsi, and Josef van Genabith. 2011. Zaman and raqm: Extracting temporal and numerical expressions in arabic. In *Information Retrieval Technology*, pages 562–573.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification.

In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. ArXiv:2402.05672v1 [cs.CL, cs.IR], Technical Report, submitted 2024-02-08.

Ahmed Wasfy, Omer Nacar, Abdelakreem Elkhateb, Mahmoud Reda, Omar Elshehy, Adel Ammar, and Wadii Boulila. 2025. Qari-ocr: High-fidelity arabic text recognition through multimodal large language model adaptation. *arXiv preprint arXiv:2506.02295*.

Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.

Georg Wenzel and Adam Jatowt. 2023. An overview of temporal commonsense reasoning and acquisition. *arXiv preprint arXiv:2307.15394*.

Kam-Fai Wong, Yunqing Xia, Wenjie Li, and Chunfa Yuan. 2005. An overview of temporal information extraction. *International Journal of Computer Processing of Languages*, 18(2):219–240.

Fadi A. Zaraket. 2012. Arabic temporal entity extraction using morphological analysis.

Xiaobin Zhang, Liangjun Zang, Qianwen Liu, Shuchong Wei, and Songlin Hu. 2024. Event temporal relation extraction based on retrieval-augmented on llms.

Y. Zhang and Q. Han. 2025. Enhancing pre-trained language model by answering natural questions for event extraction. *Frontiers in Artificial Intelligence*, 8:1520290.