

Article

Event-Conditioned Causal Extraction in Saudi Dialect: A Comparative Study of Dialect-Trained BERTs and LLM Prompting

Mariam Elhussein ^{1,*}, Samiha Brahimi ², Reem Osman ³ and Suhier Elfaki ⁴ ¹ Department of Mathematics and Computing, Mount Royal University, Calgary, AB T3E 6K6, Canada² Department of Mathematics and Computer Science, University of Lethbridge, Calgary, AB T2G 4V1, Canada; samiha.brahimi@uleth.ca³ Computer Department, Applied College, Imam Abdulrahman Bin Faisal University, Dammam 34212, Eastern Province, Saudi Arabia; roosman@iau.edu.sa⁴ College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam 34212, Eastern Province, Saudi Arabia; sbelfaki@iau.edu.sa

* Correspondence: melhussein1@mtroyal.ca

Abstract

Causality extraction is an important task in natural language processing, yet it remains underexplored in informal Arabic social media text, particularly in dialectal contexts. This study investigates causal-reason extraction from Saudi Arabic tweets related to sick-leave requests. A gold-standard dataset was annotated for multiple causality-related tasks, including cause-presence detection, cause-span extraction, cause-category classification, causal-marker detection, and causal marker text identification. The study compares two modeling paradigms: fine-tuned BERT-based models, represented by SaudiBERT and AraBERT, and prompting-based large language models (LLMs), represented by GPT-4.1-mini and Gemini-2.5-flash. The descriptive analysis showed strong class imbalance, substantial implicit causality, and uneven cause-category distributions. Results showed that SaudiBERT generally outperformed AraBERT when macro-level and minority-class performance were considered. Among LLMs, Gemini-2.5-flash achieved the strongest overall performance, particularly under natural 10-shot single-tweet prompting, while balanced few-shot prompting improved macro-F1 for cause-category classification. However, step-wise prompting did not consistently improve performance and may have introduced error propagation. Overall, the findings show that causality extraction in informal Saudi Arabic remains challenging, especially for implicit causal expression. The study highlights the complementary strengths of dialect-specific transformers and LLM-based prompting for Arabic causality extraction.

Keywords: Arabic NLP; causality extraction; Saudi Arabic; dialectal social media; large language models; BERT-based models



Academic Editor: Patricia Anthony

Received: 16 May 2026

Revised: 6 July 2026

Accepted: 13 July 2026

Published: 17 July 2026

Copyright: © 2026 by the authors.

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

The misuse of sick-leave documents in Saudi Arabic was investigated in an earlier work [1]. Within the employment context, employees who fail to report to work are generally required to provide verified proof of illness. Sick-leave certificates are therefore issued by licensed medical institutions to confirm that an employee is medically unfit for duty for a specified period. However, it has been observed that some employees and students obtain these documents in order to secure undeserved leaves. Although governmental

regulations and legal controls have been established to address absenteeism, fraudulent sick-leave certificates continue to circulate.

Social media platforms, particularly Twitter, have been identified as key channels through which fraudulent sick-leave documents are promoted and sold. These seller accounts are typically operated anonymously or under pseudonyms using trending hashtags along with ones relevant to sick-leaves. Individuals seeking to obtain sick-leave certificates through Twitter commonly search for and engage with these hashtags. The same hashtags are also used to express the reasons for seeking such documentation. Some of these stated reasons appear legitimate, whereas others are clearly expressed in a humorous or non-serious manner. However, based on observation, it is evident that even humorous statements may still reflect genuine motivations for obtaining a sick-leave document. Although the authenticity of such claims cannot be verified, they nevertheless provide insight into perceived or potential reasons for requesting sick-leaves and should therefore be taken into consideration during analysis. The fact that the tweet includes a hashtag relevant to sick-leave online selling would automatically align it as relevant. Reasons and causes are expected to provide insights that can be useful for government officials to design policies to combat this problem.

Causation is “a textual relationship that relates two situations” [2]. It is a formal branch of natural language processing (NLP) that focuses on causal relation extraction. The task in which an event (cause) and another event (effect) are related is detected. The first event is a consequence of the second. There are many classifications for causality extraction that rely on the structure of the sentence. Ref. [3] introduced some of these classes including the widely known explicit vs. implicit causality. An explicit causality is one that has all its elements, including cause, effect, and link. In an implicit causality, one or more of these elements are missing [2]. A recent notion introduced as a subtype of implicit causality is “event-conditioned causality”, in which one side of the causal relation (either a cause or an effect) is assumed to be known in advance, and its counterpart is inferred from surrounding text or a larger corpus [4].

In this study, event-conditioned causality is applied to identify reasons for obtaining sick-leaves. The assumption is that some of the tweets that use the sick-leave hashtags mention the reason for obtaining one. In this case, the effect is known, which is to obtain a sick-leave. This is true whether the user proceeded with obtaining the document or not. Methodologies of causality extraction include rule-based, supervised, and transformer-based. Since rule-based approaches rely on known sentence structures, they were successful in explicit causality extraction. On the other hand, supervised and transformer-based approaches rely heavily on the existence of handcrafted datasets in order to train the models. Arabic causality extraction research focuses predominantly on explicit causal relations in Modern Standard Arabic (MSA) through the use of manually labeled linguistic patterns identifying connectives such as لأن, بسبب, and نتيجة لـ meaning “because”, “because of”, and “as a result of”. More efforts have been supported by pretrained Arabic language models and discourse-level corpora such as the Leeds Arabic Discourse Treebank [5]. At the same time, increasing volumes of Arabic online communication have been produced in dialectal Arabic. A local version of Arabic has distinct characteristics, many of which are not reflected in MSA-based datasets. In general, few researchers were focused on NLP tasks pertaining to specific dialects such as Egyptian [6] or Moroccan [7]. Nonetheless, very limited efforts were focused on causality extraction for the Arabic language and no work was found to have addressed mono-dialect causality.

Recent progress has been reported to support NLP research based on Saudi Arabian Dialect (SAD), most notably SaudiBERT [8], through which strong performance has been demonstrated across dialectal understanding tasks. In parallel, large language models

(LLMs) such as ChatGPT and Gemini have exhibited reasoning capabilities across languages and domains, including tasks involving causal inference. Thus far, no systematic comparison has been conducted between dialect-specialized pretrained models and general-purpose LLMs for causal extraction in Saudi dialect. This absence of empirical comparison has been viewed as especially important, since LLMs are often assumed to offer annotation-efficient reasoning, while dialect-specific pretrained models are assumed to provide stronger lexical grounding.

Therefore, the main research question examined is the following: to what extent can dialect-specific pretrained transformers and general-purpose LLMs successfully perform this task?

The principal contributions of this work are summarized as follows.

- i. Two modeling paradigms are benchmarked quantitatively, including
 - BERT-based neural models fine-tuned for causal classification, namely AraBERT and SaudiBERT;
 - LLMs (ChatGPT and Gemini) evaluated under prompting-based settings for causal classification and span and marker text extraction.
- ii. A qualitative analysis of LLM span extraction and categorization of errors observed are performed.
- iii. To the best of our knowledge, this is the first study to provide a comparative analysis across these paradigms.
- iv. A curated dataset of Saudi-dialect causal relations is introduced, consisting of real-world social media text annotated for cause role, explicit versus implicit realization, and event granularity. This is the first available dataset to the best of the researchers' knowledge.

This paper is structured as follows. First, a comprehensive literature review is presented, covering research on causality extraction in general as well as in MSA, and identifying the gaps to which this study contributes. The methodology is then described through an outline of the two approaches introduced earlier. This is followed by the results and finally the discussion and conclusion.

2. Related Work

Causality extraction has been treated as a specialized information extraction task within NLP, with applications ranging from health surveillance on social media to decision support in business analytics. In the health domain, ref. [9] proposed lexico-syntactic patterns over dependency parses to extract explicit causal relations from Twitter, focusing on three targets: stress, insomnia, and headache. Using a corpus of 24 million tweets, their rule-based pipeline was evaluated against human annotations and achieved high precision (74.59–92.27%). At a broader level, ref. [10] surveyed causality extraction methods across domains and categorized studies based on the used methodology into knowledge-based, statistical machine-learning, and deep-learning approaches. They concluded that most systems are evaluated with standard classification metrics (precision, recall, F1, accuracy) on benchmark corpora such as SemEval-2010 Task 8, Causal-TimeBank, and DoCRED. They also discussed that deep neural models generally outperform rule-based and feature-engineered baselines for explicit, intra-sentential causality, while performance remains substantially weaker for implicit and document-level relations.

Recent advances in causality extraction and understanding have leveraged pretrained language models and benchmark datasets to capture nuanced causal relationships between events and questions. CausalBERT [11] explored the capacity of transformer-based models to identify causal associations when events are expressed in natural language, par-

ticularly targeting the detection of implicit causal links that classical pattern-based methods struggle to capture; experiments demonstrated state-of-the-art performance across three evaluation datasets (e.g., SemEval 2007, SemEval 2010, and ADE). Their work showed that BERT was able to improve generalization for causality detection tasks. Beyond task framing and datasets, efforts to embed causal knowledge directly into pretrained models have been explored in [12]. BERT is enhanced by integrating causal pairs collected from unstructured text alongside a regularization strategy to preserve existing linguistic knowledge, across multiple causal pair classifications, causal QA, and causal inference tasks. This model consistently outperformed baseline pretrained models, suggesting that explicit incorporation of causal information into model parameters improves performance on diverse causality-oriented benchmarks. Extending these ideas to enterprise reporting, ref. [13] introduced the Text2CausalGraph framework, which constructs causal knowledge graphs from corporate 10-K reports via a transformer-based pipeline. BERT is fine-tuned to classify sentences as causal or non-causal (macro-F1 $\approx$ 0.88–0.89), SpanBERT and DistilBERT are adapted for token-level cause/effect/trigger extraction (macro-F1 up to 0.89), and a logistic-regression classifier with TF-IDF n-grams is used to map extracted phrases into a stakeholder taxonomy (macro-F1 = 0.78), all evaluated on a manually annotated gold-standard dataset with 80/20 train–test splits and cross-validation.

Research on Arabic causality extraction has largely concentrated on explicit, linguistically signaled relations in MSA. Early work such as [2] adopted a rule-based approach relying on handcrafted lexical and syntactic patterns. Connectives such as لأن، بسبب، نتيجة ل meaning “because”, “because of”, and “as a result of” were the main indicator of causality. The performance was evaluated using standard precision–recall measures. Results demonstrated high precision (typically above 80%) but lower recall. More recent efforts introduced morphological awareness into causal detection. In [2], causality identification was enhanced by modeling proclitic forms that attach to Arabic words (e.g., ك، ل، ب) and function as discourse markers. A supervised classification framework was trained using these morphological cues as features, and experiments reported improved F1-scores over lexical-only baselines. However, these studies remain constrained to MSA and explicit causality, and performance is largely reported on curated datasets rather than user-generated content. More recently, progress in Arabic dialect modeling has been supported by SaudiBERT [8], a transformer language model pretrained on large-scale Saudi dialect corpora. SaudiBERT has been shown to outperform MSA-trained models on dialectal downstream tasks including sentiment and topic classification with consistent accuracy and macro-F1 improvements across benchmarks. Although SaudiBERT was not originally developed for causality extraction, its demonstrated ability to capture dialect-specific morphology and usage strongly suggests that dialect-specialized pretrained models hold significant promise for dialectal causal inference, particularly in Saudi colloquial text such as Twitter, which forms the basis of the present study.

LLMs have recently been examined for their capacity to perform causal reasoning tasks without explicit task-specific supervision. Early evidence suggested that instruction-tuned prompting strategies, particularly chain-of-thought prompting, improve causal inference accuracy over zero-shot baselines on counterfactual and intervention-style multiple-choice datasets [14]. More systematic evaluation has emerged in work such as Causal-Bench, which benchmarked GPT-3.5, GPT-4, and other LLMs on causal discovery and effect identification tasks. Results demonstrated that GPT-4 performed substantially above random chance, achieving up to 0.74 accuracy on intervention questions, although performance varied widely across task types. The researchers concluded that causal reasoning ability remains highly context-dependent [15]. Recent work has examined the use of LLMs for zero-shot text classification and information extraction, including relation extraction

and causality-related tasks. Ref. [16] specifically evaluate zero-shot and few-shot LLM prompting for classifying causal relations, showing that causal relation identification remains challenging even for state-of-the-art LLMs. Relatedly, the CASE 2022 Event Causality Identification shared task formalized causal relation detection as both a sentence-level classification task and a span identification task for cause, effect, and signal spans [17]. For Arabic language, studies demonstrate the growing use of zero-shot LLM prompting for classification tasks, although not necessarily causality-specific [18,19]. Collectively, these findings suggest that LLMs exhibit emergent causal reasoning capabilities, yet their performance remains uneven across causal subtasks and sensitive to prompt design and dataset characteristics.

Research on event-conditioned causality and reason extraction has been conducted in other languages. For example, the FinReason dataset for Chinese financial announcements was introduced by [4], in which the task was defined as extracting the textual spans that explain why a previously identified event occurs. In this framework, the event is already known and is represented as a structured record that includes the event type and participant roles, and the model is required to retrieve one or more reason spans from the associated document. Two main modeling strategies were demonstrated: a sequence-labeling approach (BiLSTM-CRF), in which reason spans were treated as argument roles of the event, and a reading-comprehension approach based on BERT, in which the task was formulated as answering a “why” question conditioned on the event structure.

The reviewed literature highlights several key limitations in research. Most existing research focuses on explicit, sentence-level causality, whereas implicit and event-conditioned causality remain underexplored, particularly outside English and Chinese. Second, Arabic causality extraction is dominated by MSA, with very limited work addressing dialectal Arabic, despite the fact that dialects account for a substantial proportion of social media communication. Third, although dialect-specialized pretrained models (e.g., SaudiBERT) and general-purpose LLMs both demonstrate strong representational capabilities, a systematic comparison between these model families for dialectal causal inference has not yet been conducted. The present study addresses these gaps by applying event-conditioned causality extraction to Saudi-dialect Twitter data and by comparing the performance of dialect-specialized pretrained models and general-purpose LLMs on this task. Table 1 provides a summary of the relevant literature.

Table 1. Summary of closely related Arabic NLP studies and their relevance to the research.

Ref.	Language/Variety	Focus	Method/Model	Relevance to This Study
[20]	Arabic/MSA	Explicit causality extraction	Rule-based lexical and syntactic patterns	Early Arabic causality work, but limited to explicit MSA relations.
[2]	Arabic/MSA	Causality identification	Supervised classification with morphological causal markers	Shows the value of Arabic morphology, but does not address dialectal data.
[8]	Saudi Arabic dialect	Dialectal Arabic NLP	SaudiBERT pretrained language model	Closely related to Saudi dialect modeling and useful for comparing dialect-specialized models.
[19]	Arabic	Text classification	Zero-shot LLM prompting	Relevant to Arabic LLM-based classification, though not causality-specific.
[10]	Arabic	Text classification	Zero-shot/prompt-based LLM methods	Supports the use of LLMs for Arabic NLP tasks, but lacks focus on causal extraction.

3. Methodology

3.1. Task Definition

In this study, event-conditioned causal extraction is addressed under the constraint that the effect is fixed and implicitly signaled. The effect corresponds to the intention to obtain a sick-leave, which is not expressed directly in the text but is implied through the use of sick-leave-related hashtags. For each instance, a tweet containing such a hashtag is provided, and the objective is to automatically identify the underlying cause motivating the sick-leave request.

Formally, for a tweet T containing a sick-leave hashtag H , the model is required to identify a textual span C representing the cause in the causal relation $EC \rightarrow E$, where E denotes the implied effect “obtaining a sick-leave.” Tweets in which no valid cause is expressed are labeled as negative. The model is also expected to identify the causal-marker presence in the case of explicit cause, causal marker text, causal span text, and causal category.

3.2. Dataset Description

A dataset consisting of 7656 Saudi-dialect tweets was reused from prior work on sick-leave discourse in Saudi Twitter (collected and described in Section 3.1 of the original study [1]). In the present study, this dataset was repurposed for causal analysis under an event-conditioned setup. The data was collected using a set of keywords identified through Hashtagify.me (<https://hashtagify.app/> (accessed on 19 May 2026)), a tool that suggests frequently co-occurring hashtags. The initial keyword used to generate the list was “سكليف,” a transliteration of the English term “sick leave,” which is commonly used to refer to sick-leave documents. The resulting keyword list included Arabic equivalents and related variations with similar meanings. In total, nine keywords were used, and a tweet was retrieved if it contained at least one of them. The tweet location was restricted to Saudi Arabia as a condition for inclusion.

Tweets posted between 1 January 2010, and 8 January 2021 were downloaded and manually labeled into two categories: promoter and non-promoter. During annotation, it was observed that users referred to sick leave in different ways, including joking about needing sick leave, promoting sick-leave services, or requesting sick leave. Since most tweets were humorous and only a small number involved requests, both joking and request-related tweets were grouped under the non-promoter category. Tweets advertising or promoting sick-leave services were labeled as promoters. Only non-promoting sick-leaves were kept for the purpose of this dataset.

Each tweet in the dataset was authored by a unique user or by repeated users who produced multiple tweets during the collection period. Tweet-level metadata accompanying each instance included the number of replies, the number of retweets, and the number of hashtags used. Only tweets written predominantly in Saudi dialect Arabic were retained. The presence of a sick-leave-related hashtag (e.g., variants of words such as مرضية إجازة, مرضية إجازة, مرضية, #إجازة_مرضية all are variations referring to the same word which is sick-leave) was treated as an implicit signal of the effect that the user sought or referenced obtaining a sick-leave. Only the cause motivating the sick-leave request was to be identified. Candidate causal instances were initially screened using heuristic lexical signals commonly associated with causal meaning in Saudi dialect (e.g., عشان بسبب, عشان كذا translating to (“because”, “the reason why”, and “the reason is”), and additional examples exhibiting implicit causality were included to avoid over-reliance on lexical cues.

Although the dataset had been prepared for the earlier study, explicit causal labels were not available. Therefore, manual verification and labeling were carried out in the present research. Each tweet was examined by native Saudi Arabic speakers to determine

whether a cause motivating the sick-leave request was expressed. Two groups of labelers were working on the dataset; each group received half of the data. Within the group, there was a main annotator and a reviewer, who exchanged roles. Before the labeling process, the researchers met with the annotators and explained, with various examples, the required tasks. The main researcher also created a video guide explaining the various points of possible disagreement, specifically explicit vs. implicit causality. When a cause was present, the corresponding cause span was marked. Tweets that did not contain an expressed cause were labeled as negative examples. Ambiguous cases were adjudicated through discussion, and only consensus-validated instances were retained. Annotation was conducted in two levels. In Level 1, annotators determined whether each tweet contained a causal relation. Out of 7656 tweets, 429 were identified as containing causality. There were 47 disagreements at this level, and these cases were discarded, resulting in a final dataset of 7609 tweets.

In Level 2, causal tweets were further annotated for explicit/implicit causality, causal markers, and causal text. Thirteen disagreements were found in the explicit/implicit classification and were resolved through discussion. Inter-annotator agreement was assessed using disagreement rate and Cohen's kappa. In Level 1, which determined whether each tweet contained a causal relation, the disagreement rate was 0.61% (47/7656), with a Cohen's kappa value of 0.91, indicating very strong agreement. In Level 2, which classified causal tweets as explicit or implicit, the disagreement rate was 3.03% (13/429), with a Cohen's kappa value of 0.87, also indicating strong agreement. No disagreements were found in the annotation of causal markers or causal text; therefore, agreement for these tasks was 100%.

For each tweet, a set of descriptive features was derived to support analysis. First, the type of causality was recorded, distinguishing between explicit and implicit expression. Explicit causality was defined as cases in which a linguistic marker overtly indicated a causal link. Implicit causality referred to cases in which the cause was inferable from context without the use of a direct causal marker. For tweets exhibiting explicit causality, the causal marker employed in the text was documented.

Finally, the semantic category of the cause was assigned. Causes were grouped into broad thematic classes, including:

- Health-related causes (e.g., illness, injury, medical conditions);
- Well-being or psychological factors (e.g., stress, burnout, exhaustion);
- Caregiving responsibilities (e.g., supporting a child or family member);
- Social or lifestyle factors (e.g., entertainment, travel, leisure activity);
- General or unspecified reasons where no clear motivation was expressed.

These features were used to enable both qualitative interpretation and quantitative modeling of causal expression in Saudi-dialect sick-leave discourse. The study combines quantitative modeling with qualitative interpretation. Quantitative modeling is used to evaluate model performance across the causality extraction tasks using standard classification and text-matching metrics. Qualitative interpretation is then used to examine model outputs in more detail, identify recurring error patterns, and explain the linguistic and contextual factors that affect performance. A feature schema was created so that causal expression could be analyzed. A summary of the annotated feature space is provided in Table 2.

Table 2. Summary of annotated dataset features.

Feature	Description
Causality type	Each tweet was labeled as expressing either explicit causality or implicit causality.
Causal marker (explicit only)	For tweets expressing explicit causality, the specific linguistic marker was recorded.

Table 2. Cont.

Feature	Description
Cause presence	A binary label indicating whether a cause motivating the sick-leave request was expressed.
Cause span	The exact textual span corresponding to the cause, where present.
Cause category	The semantic class of the cause, grouped into broad categories explicit vs. implicit.

3.3. Modeling Paradigms

Two modeling paradigms were employed in this study to compare supervised dialect-specific learning with prompting-based large language model inference. The fine-tuned BERT-based models were used for classification tasks, while the LLM-based models were evaluated on both classification and text-span identification tasks.

Accordingly, the study addressed the following tasks:

- i. Cause-relation detection: determining whether a cause was expressed in the tweet;
- ii. Cause-span extraction: identifying the textual span corresponding to the expressed cause;
- iii. Cause-category classification: assigning the expressed cause to a general cause category;
- iv. Causal-marker identification: detecting whether causality was explicitly signaled and, when applicable, identifying the causal marker present in the text.

The text-span tasks, namely cause-span extraction and causal marker text identification, were conducted only using the LLM-based prompting approach. This is because applying BERT-based models to span extraction would require a different annotation and modeling setup. Specifically, the data would need to be prepared using token-level labels, such as BIO tagging, or start and end span positions. In addition, BERT classifiers produce predefined label outputs, whereas LLMs can naturally generate free-text spans in response to prompts. Therefore, using BERT for span extraction would require a separate architecture and evaluation pipeline, distinct from the classification pipeline used in the reported BERT experiments.

3.3.1. Dialect-Specific Pretrained Model

AraBERT and SaudiBERT were employed as the supervised modeling approach. The models had been pretrained on Arabic MSA, whereas SaudiBERT was trained on large-scale Saudi Arabic social media text, and therefore, it has been trained to encode lexical and morphological variation characteristics of Saudi dialects more effectively than general Arabic.

To investigate the effectiveness of dialect-specialized pretrained transformers for event-conditioned causality analysis in Saudi dialect Arabic, four supervised transformer-based classification models were developed and evaluated. The comparison between the two models was intended to examine whether dialect-specialized pretraining provides measurable advantages for causal reasoning tasks in Saudi social media discourse.

The first task addressed was cause-relation detection, in which tweets were classified into two categories:

- i. Tweets expressing a cause motivating the sick-leave request;
- ii. Tweets not expressing a cause.

The second task addressed causality-type classification. Only tweets previously identified as containing a cause were included in this stage. The objective was to determine whether the causal relation was expressed explicitly through a linguistic causal marker or implicitly through contextual inference. Two additional classification models were therefore developed:

- i. SaudiBERT for explicit/implicit causality classification;
- ii. AraBERT for explicit/implicit causality classification.

For all four models, supervised fine-tuning was conducted using the HuggingFace Transformers framework. A sequence-classification head was added on top of the pre-trained transformer encoder. The final hidden representation associated with the special classification token was passed through a linear classification layer to predict the target class label.

Tweets were tokenized using the native tokenizer associated with each pretrained model. No conversion to Modern Standard Arabic was applied in order to preserve dialectal lexical and morphological characteristics. Maximum input length was initially fixed at 128 tokens. Additional experiments using longer sequence lengths were also explored during model optimization, although no substantial improvement was observed for the explicit/implicit classification task.

The dataset was divided into training, development, and test partitions using stratified splitting. To reduce the risk of information leakage, user-level separation was enforced whenever possible so that tweets originating from the same user did not appear across different partitions. Model selection was based on development-set performance.

Fine-tuning was conducted for five epochs using the AdamW optimizer with linear learning-rate scheduling. Learning-rate values between 1×10^{-5} and 2×10^{-5} were experimentally explored during optimization. Batch size was fixed at eight due to GPU memory limitations and the relatively small dataset size. During training, weighted cross-entropy loss was employed for the cause/no-cause task in order to mitigate class imbalance. Inverse-frequency class weights were applied, together with an additional scaling factor of 1.5 for the minority CAUSE class. Gradient clipping and early stopping based on development macro-F1 were additionally employed to reduce overfitting.

3.3.2. Large Language Models (LLMs)

General-purpose large language models (LLMs), specifically ChatGPT using GPT-4.1-mini and Gemini using Gemini-2.5-flash, were also evaluated. These models were selected as comparable efficient general-purpose LLMs suitable for high-volume text annotation tasks, although they are not assumed to be equivalent in architecture, training data, or language-specific performance. Unlike the SaudiBERT and AraBERT models, the LLMs were not fine-tuned on the study dataset. Instead, they were assessed through prompting-based experiments in order to examine their ability to generate structured labels directly from Saudi Arabic tweet text.

The LLM evaluation focused on the same annotation attributes used in the supervised BERT-based experiments. This design allows the LLM outputs to be compared with the labels produced by the SaudiBERT and AraBERT configurations.

The prompting-based LLM experiments were conducted through the API rather than the web interface. The data was uploaded on a Google Sheet and an API call for the relevant LLM would provide the results in cells next to each tweet. For all experiments, the generation settings were kept fixed across models and prompting configurations. Two main settings were set: the temperature, which controls the randomness of the model output, and the top-p, which controls the range of candidate tokens considered during generation. In this study, temperature was set to 0 to reduce randomness and improve reproducibility, while top-p was set to 1.0 to avoid additional restrictions on token selection.

The reported results are based on a single complete run for each model and prompting configuration. Multiple repeated runs were not averaged. This decision was made because the experiments used deterministic or near-deterministic settings, and the main objective

was to compare model performance under controlled prompting conditions. Prompts used are available in the Appendix A.

Due to the cost and resource requirements of API-based LLM experiments, the prompting-based evaluation was conducted on a balanced subset of the gold-standard dataset rather than the full dataset. The subset contained 858 tweets that was randomly selected and was constructed to include approximately equal numbers of cause-present and cause-absent instances. Downstream tasks, including explicit/implicit classification, cause-span extraction, and causal-marker identification, were evaluated only on the cause-present instances within this balanced evaluation subset. Table 3 shows the breakdown of the sub dataset.

Table 3. Evaluation subsets used for the LLM experiments, including the balanced cause-presence set and the cause-present subset used for downstream tasks.

Task	Evaluation Subset	N
Cause presence	Balanced LLM evaluation subset	858
Explicit/implicit classification	Cause-present tweets within the balanced subset	429

Several prompting configurations were tested. First, a zero-shot setting was used, in which the models received only the task instructions and label definitions, without any annotated examples. Second, a few-shot setting with 10 examples was used to examine whether limited in-context demonstrations improve label consistency and extraction accuracy. To examine the effect of in-context example selection and to address label imbalance, four versions of the 10-example few-shot prompt were created. These versions varied in the composition of examples, particularly in how they represented underrepresented labels and less frequent annotation cases. This allowed the study to assess whether few-shot performance was affected not only by the presence of examples, but also by which examples were included in the prompt.

The few-shot examples were designed to support all annotation attributes: cause-presence detection, cause-span extraction, cause-category assignment, causal-marker detection, and causal-marker extraction. The experiments therefore compared zero-shot prompting with multiple imbalance-aware few-shot prompt variants, while keeping the core task instructions and output structure consistent across runs.

A step-wise zero-shot prompting setting was also introduced to examine whether task decomposition improves performance. In this configuration, no labeled examples were provided. Instead, the annotation process was structured sequentially: cause presence was identified first, followed by cause-span extraction, cause-category assignment, causal-marker detection, and causal-marker extraction when applicable. This setting will be compared to multi-task zero-shot prompting.

Each prompting configuration was executed in a separate session to avoid carry-over effects from previous prompts or model responses. The prompts were submitted independently for each experimental condition. The same input tweets and output schema were used across models to ensure that differences in performance were attributable to model behavior rather than differences in prompt structure or data input.

3.4. Evaluation Protocol

3.4.1. Quantitative Evaluation Protocol

For classification-based attributes (in BERT and LLMs), including cause presence, cause category, and causal-marker presence, model outputs were compared directly against the gold-standard labels. Performance was measured using standard classification metrics, including precision, recall, F1-score, and accuracy. Because some labels were im-

balanced, macro-averaged and weighted-average F1-scores were used where appropriate. Macro-F1 was used to assess performance across labels without allowing majority classes to dominate the results, while weighted-F1 was used to reflect performance in relation to the actual label distribution.

For extraction-based attributes, including cause span and causal-marker span, the predicted text spans were compared with the gold-standard spans. Exact-match evaluation was used to determine whether the model extracted the same span as the annotated reference. In addition, partial-match analysis was considered to account for cases where the model extracted a semantically correct span but differed slightly from the gold-standard boundary. This was particularly important where cases like informal spelling, dialectal expressions, and flexible phrase boundaries may lead to small surface-level differences in extracted spans.

For the LLM experiments, results were compared across three dimensions. First, zero-shot and few-shot settings were compared to assess the effect of in-context examples. Second, the few-shot prompt versions were compared to examine whether example composition and imbalance-aware example selection affected performance. Third, single-tweet processing and batch processing were compared to assess whether batching influenced annotation quality. For batch-processing experiments, performance was calculated separately for each batch size in order to identify the point at which model performance began to decline.

The final evaluation therefore compared model performance at multiple levels: between SaudiBERT and AraBERT, between the two LLMs, across prompting strategies, across few-shot prompt versions, and across single-tweet and batch-processing modes. A summary of the evaluation measures applied to the cause-relation detection and cause-span extraction tasks is presented in Table 4.

Table 4. Summary of evaluation tasks, metrics, and scoring criteria.

Task	Definition	Evaluation Unit	Primary Metrics	Scoring Condition
Cause-Relation Detection	Determine whether a tweet expresses a cause motivating the sick-leave request	Tweet-level	Precision, Recall, Macro-F1, Accuracy	A prediction is counted as correct only when the predicted binary label matches the gold label
Cause-Span Extraction	Identify the textual span corresponding to the expressed cause	Span-level/ text-level	Exact Match, Partial Match/token-level F1	Span evaluation is applied only to tweets labeled as containing a cause. Exact match requires the predicted span to match the gold span exactly; partial/token-level scoring captures overlap between predicted and gold spans
Cause-Category Classification	Assign the identified cause to one of the predefined semantic categories, such as health, well-being, caregiving, lifestyle, or general/unspecified	Tweet-level, cause-positive instances only	Accuracy, Macro-F1	A prediction is treated as correct only when the predicted category matches the annotated gold category
Causal-Marker Presence Detection	Determine whether the tweet contains an explicit causal marker	Tweet-level	Precision, Recall, Macro-F1, Accuracy	A prediction is counted as correct only when the predicted binary marker-presence label matches the gold label
Causal-Marker Extraction	Identify the lexical causal marker used in the tweet, when one is present	Marker-level/ text-level	Marker-level Accuracy, Exact Match	A prediction is scored as correct only when the returned marker matches the annotated causal marker
Prompting Configuration Comparison	Compare model performance across zero-shot and 10-example few-shot prompting, including the four few-shot prompt versions	Experiment-level	Macro-F1, Accuracy, Invalid-output Rate, Performance Variation Across Prompt Versions	Performance is compared across prompt settings using the same test set and gold labels; differences are interpreted in relation to prompt type and example composition

Table 4. Cont.

Task	Definition	Evaluation Unit	Primary Metrics	Scoring Condition
Processing Mode Comparison	Compare single-tweet processing with batch processing and examine the effect of batch size	Experiment-level/ batch-size level	Macro-F1, Accuracy, Span-match Scores, Invalid-output Rate	Performance is calculated separately for single-tweet and batch settings; batch performance is examined across batch sizes to identify where output quality begins to decline

3.4.2. Qualitative Evaluation Protocol

To complement the quantitative evaluation, a qualitative analysis was conducted to examine how the models handled causal meaning in Arabic sick-leave tweets. The purpose of this analysis was not only to identify whether predictions were correct or incorrect, but also to understand the types of linguistic and contextual challenges that affected model performance.

A sample of model outputs was selected from the evaluation results for closer inspection. The sample included correct predictions, false positives, false negatives, partially correct span extractions, and incorrect cause-category assignments. Particular attention was given to cases where the quantitative scores alone did not fully explain model behavior, such as tweets with implicit causality, dialectal expressions, overlapping cause categories, or causal markers embedded in informal language.

Each selected tweet was manually reviewed by comparing the gold annotation with the model prediction. The analysis focused on five aspects: whether the model correctly detected the presence of a cause, whether the extracted cause span matched the intended causal explanation, whether the assigned cause category reflected the meaning of the tweet, whether causal markers were correctly identified, and whether the model added, omitted, or misinterpreted relevant text.

The qualitative findings were used to interpret the quantitative results and explain differences between models and prompting strategies. For example, cases with partial span overlap helped clarify why token-overlap scores were higher than exact-match scores, while examples of minority-category confusion helped explain low macro-F1-scores despite reasonable accuracy. This qualitative layer therefore provided a deeper understanding of the strengths and limitations of LLMs' zero-shot prompting approach in extracting causal information from informal Arabic tweets.

4. Results

This section presents the empirical results of the study in three subsections. First, descriptive statistics are reported for the gold-standard annotated dataset in order to characterize the distribution of causal annotations and provide context for interpreting model performance. Second, the results of the fine-tuned BERT-based models are presented, comparing SaudiBERT and AraBERT on the main classification tasks. Finally, the LLM-based results are reported, comparing GPT-4.1-mini and Gemini-2.5-flash across prompting configurations, classification performance, and text-span identification tasks.

4.1. Gold-Standard Dataset Overview

The descriptive statistics were calculated after excluding tweets without a valid cause-presence annotation. The analyzed gold-standard dataset therefore consisted of 7609 tweets. The dataset was highly imbalanced: only 429 tweets were annotated as containing an expressed cause (5.6%), while 7180 tweets were annotated as not containing an expressed cause (94.4%). Among the 429 cause-present tweets, 398 tweets had an annotated cause span, representing 92.8% of the cause-present subset. The cause spans were generally short, with an average length of approximately 2.77 tokens and a median of

2 tokens. This suggests that causes were often expressed through short phrases rather than long explanatory clauses, which is consistent with the maximum text length allowed by Twitter/X.

The explicit/implicit causality labels were available for 429 cause-present tweets. Of these, 232 tweets were labeled as explicit (54%), while 197 tweets were labeled as implicit (45.9%). Causal markers were annotated in 232 cause-present tweets.

Cause-category labels were available for 429 cause-present tweets. The most frequent category was well-being or psychological, with 187 tweets (43.5%). This was followed by general unclear with 106 tweets (24.7%) and health-related with 100 tweets (23.3%). The less frequent categories were social or lifestyle factors with 24 tweets (5.5%) and caregiving responsibilities with 12 tweets (3.0%). Table 5 provides a summary of these statistics.

Table 5. Descriptive statistics of the gold-standard dataset.

Statistic	Count	Percentage/Value
Total analyzed tweets	7609	100%
Tweets with expressed cause	429	5.6%
Tweets without expressed cause	7180	94.4%
Cause-present tweets with annotated cause span	398	92.8%
Average cause-span length	—	2.77 tokens
Median cause-span length	—	2 tokens
Explicit causality	232	54.1%
Implicit causality	197	45.9%
Tweets with causal markers	232	54.1%
Well-being or psychological	187	43.6%
General/unclear	106	24.7%
Health-related	100	23.3%
Social or lifestyle factors	24	5.6%
Caregiving responsibilities	12	2.8%

4.2. BERT-Based Model Results

Two transformer-based language models, SaudiBERT and AraBERT, were evaluated on two downstream tasks: (i) cause-presence detection and (ii) explicit/implicit causality classification. Table 6 summarizes the overall performance of both models. Figures 1 and 2 present the confusion matrices and precision–recall (PR) curves for the cause-presence detection task, whereas Figure 3 illustrates the confusion matrices for the explicit/implicit classification task.

No cause	6784	396
	284	145
AraBERT		
No cause	6631	549
	231	198
SaudiBERT		

Figure 1. Confusion matrices for SaudiBERT and AraBERT on the cause-presence detection task. Color intensity indicates frequency; darker shades represent higher counts.

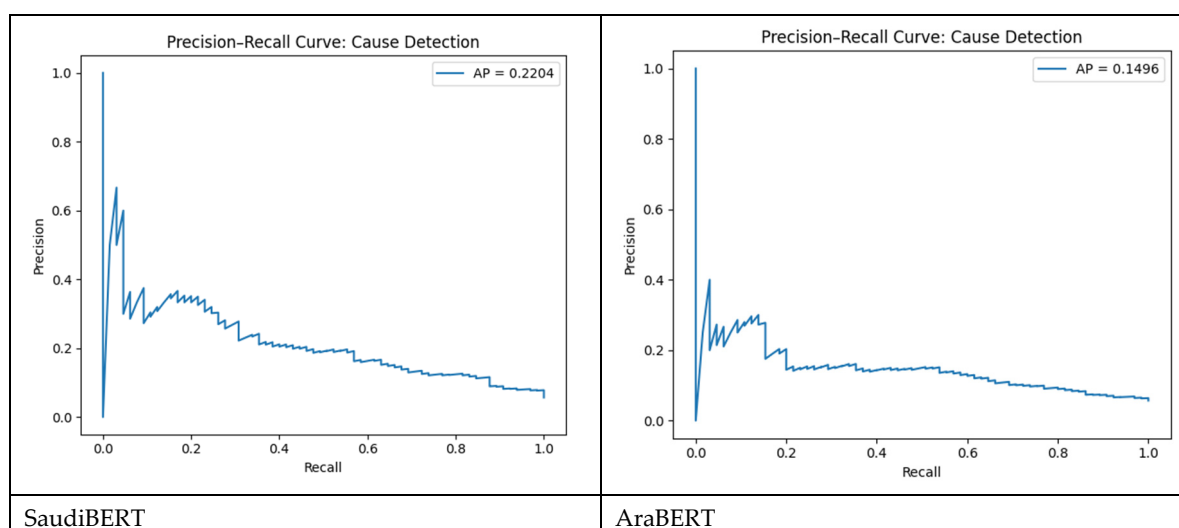


Figure 2. Precision–recall curves for SaudiBERT and AraBERT on the cause-presence detection task.

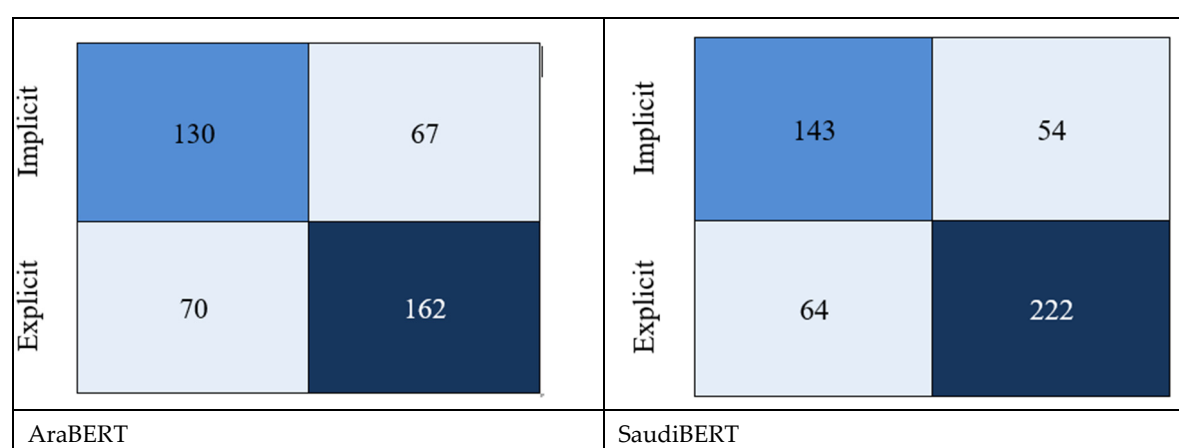


Figure 3. Confusion matrices for SaudiBERT and AraBERT on the explicit/implicit causality classification task. Color intensity indicates frequency; darker shades represent higher counts.

Table 6. Performance comparison of fine-tuned SaudiBERT and AraBERT models. Bold values indicate the best-performing result for each task.

Task	Model	Accuracy	Macro Precision	Macro Recall	Macro-F1	Weighted-F1	Key Positive/ Minority-Class Result
Cause-Presence Detection	SaudiBERT	0.8975	0.6169	0.6927	0.6417	0.9100	Cause-present F1 = 0.3390; Recall = 0.4615
Cause-Presence Detection	AraBERT	0.9098	0.6139	0.6414	0.6256	0.9147	Cause-present F1 = 0.2993; Recall = 0.3385
Cause-Category Classification	SaudiBERT	0.7246	0.7235	0.7242	0.7237	0.7249	Implicit F1 = 0.7077; Explicit F1 = 0.7397
Cause-Category Classification	AraBERT	0.6812	0.6795	0.6795	0.6795	0.6812	Implicit F1 = 0.6562; Explicit F1 = 0.7027

4.2.1. Cause-Presence Detection

Table 4 shows that both models achieved high overall accuracy; however, because only 5.6% of the dataset belongs to the CAUSE class, accuracy alone is insufficient for evaluating performance. As shown in Table 4, SaudiBERT achieved a substantially higher recall and F1-score for the minority CAUSE class, whereas AraBERT achieved slightly higher overall accuracy by favoring the majority NO_CAUSE class. This trend is clearly reflected in the confusion matrices (Figure 1), where SaudiBERT retrieved considerably more causal

tweets but also produced more false positives. Conversely, AraBERT generated fewer false positives at the expense of missing a larger number of genuine causal tweets.

The PR curves (Figure 2) provide a more informative assessment of the highly imbalanced dataset than overall accuracy. SaudiBERT maintained a stronger precision–recall trade-off across decision thresholds, supporting its superior ability to retrieve minority-class instances. The results reported in Table 4 further demonstrate that both models achieved precision and F1-scores above 93% for the NO_CAUSE class, while performance on the CAUSE class remained substantially lower because of the scarcity and linguistic variability of causal tweets.

Many false positives contained medical vocabulary frequently associated with sick leave without expressing an actual causal relationship. For example:

Original (Arabic)	“...بعد شوي بروح المستشفى اتمريض عشان اخذ عذر طبي”
English translation	“I am going to the hospital shortly to pretend to be sick so that I can obtain a medical excuse.”

Although the tweet contains words such as “hospital” and “medical excuse”, it merely describes an intention rather than a causal relation, leading the classifier to incorrectly predict the CAUSE class. Many false negatives involved implicit causal reasoning requiring contextual interpretation. For example:

Original (Arabic)	“...أنا ناقد وناقم على مستشفى رعاية الرياض... يرفض يعطيني سكليف”
English translation	“I am very dissatisfied with Riyadh Care Hospital... they refuse to give me a sick-leave certificate.”

Here, the causal relation is implied through the narrative rather than introduced by explicit causal markers, explaining why the model failed to recognize it.

4.2.2. Explicit/Implicit Causality Classification

As shown in Table 4, SaudiBERT also outperformed AraBERT on the explicit/implicit classification task. According to the results in Table 4, SaudiBERT achieved balanced precision and recall across both classes, indicating that it did not exhibit a strong bias towards either explicit or implicit causality. The confusion matrix (Figure 3) confirms that most remaining errors were due to confusion between semantically similar instances rather than systematic misclassification of one category.

Inspection of the error analysis indicates that the remaining errors mainly occur in borderline cases where the distinction between explicit and implicit reasoning depends on discourse context. For example:

Original (Arabic)	“...عندي سكليف من مستشفى خاص”
English translation	“I have a sick-leave certificate issued by a private hospital.”

Although annotated as an explicit reason, the classifier predicted it as implicit because the explanation relies largely on conversational context rather than overt discourse markers.

Overall, the combined evidence from Tables 4–6 and Figures 1–3 demonstrates that SaudiBERT provides more balanced performance than AraBERT. Its stronger minority-class recall, improved precision–recall characteristics and better handling of Saudi Arabic dialect make it the preferred model for subsequent experiments.

4.3. LLM-Based Results

The LLM-based experiments evaluated GPT-4.1-mini and Gemini-2.5-flash under zero-shot and few-shot prompting configurations. This subsection presents these findings.

4.3.1. Zero-Shot Performance

In the zero-shot setting, the models were provided with task instructions, label definitions, and the required output format, but no annotated examples. Tables 3 and 4 report the zero-shot performance of GPT and Gemini on the Saudi Arabic sick-leave tweet annotation tasks. The results include classification tasks and text-matching tasks. Classification tasks were evaluated using accuracy, macro-F1, and weighted-F1, while extraction tasks were evaluated using exact-match rate and average token-overlap score.

In the zero-shot setting, Gemini outperformed GPT on “cause presence” and “cause category”. For cause-presence detection, Gemini achieved an accuracy of 0.708 compared with GPT’s accuracy of 0.651. However, the class-level results show that both models still struggled with the minority “yes” class. GPT-mini achieved an F1-score of 0.34 for tweets where a cause was expressed, while Gemini achieved a slightly higher F1-score of 0.37. Gemini’s improvement therefore reflects better overall performance, but the detection of cause-positive tweets remained challenging for both models.

For cause-category classification, Gemini also achieved higher aggregate performance than GPT. Gemini reached an accuracy of 0.628 and weighted-F1 of 0.650, compared with GPT’s accuracy of 0.570 and weighted-F1 of 0.614. However, the macro-F1-scores for both models were very low: 0.064 for GPT and 0.068 for Gemini. This can be attributed to the class imbalance problem which influenced the performance.

For causal-marker presence detection, GPT achieved an accuracy of 0.854. The class-level results show a strong imbalance effect: the model performed well on the majority “no marker” class but poorly on the minority “yes marker” class. Table 7 displays the results for the zero-shot classification task for Gemini and ChatGPT.

Table 7. Zero-shot classification performance grouped by annotation task.

Task	Model	Accuracy	Macro Precision	Yes-Class Precision	Macro-F1	Weighted-F1
Cause Presence/Cause Expressed	GPT	0.651	0.555	0.277	0.550	0.675
Cause Presence/Cause Expressed	Gemini	0.708	0.585	0.327	0.590	0.719
Cause Category	GPT	0.570	—	—	0.064	0.614
Cause Category	Gemini	0.628	—	—	0.068	0.650
Causal-Marker Presence	GPT	0.854	0.575	0.280	0.506	0.810
Causal-Marker Presence	Gemini	0.838	0.590	0.300	0.561	0.816

For the text-matching tasks, Gemini outperformed GPT on cause-span extraction. Gemini achieved an exact-match rate of 0.645 and an average token-overlap score of 0.662, compared with GPT’s exact-match rate of 0.574 and average token-overlap score of 0.595. This suggests that Gemini was better able to identify the textual span corresponding to the expressed cause in the zero-shot setting.

For causal marker text matching, however, GPT achieved slightly higher performance than Gemini. GPT reached an exact-match rate of 0.849 and an average token-overlap score of 0.849, compared with Gemini’s exact-match rate of 0.825 and average token-overlap score of 0.826. Overall, the text-matching results suggest that the two models differed by task: Gemini performed better on cause-span extraction, while GPT performed slightly better on causal marker text matching. Table 8 presents these results.

Overall, the zero-shot results suggest that Gemini-2.5-flash performed better than GPT-4.1-mini on the reported cause-presence and cause-category tasks. Nevertheless, both models showed evidence of imbalance sensitivity. Weighted-F1-scores were consistently higher than macro-F1-scores, and performance on minority labels remained weak. The very low macro-F1 for cause-category classification in particular indicates that zero-shot

prompting alone was insufficient for reliable fine-grained cause classification. These results provide the baseline for assessing whether few-shot prompting and imbalance-aware example selection improve performance in later experiments.

Table 8. Zero-shot text-matching performance grouped by annotation task.

Task	Model	Exact Match	Average Token Overlap	Metric Type
Cause Span Text	GPT	0.574	0.595	Exact match + token overlap
Cause Span Text	Gemini	0.645	0.662	Exact match + token overlap
Causal Marker Text	GPT	0.849	0.849	Exact match + token overlap
Causal Marker Text	Gemini	0.825	0.826	Exact match + token overlap

A zero-shot step-wise prompting strategy was evaluated on the same dataset. Unlike the earlier multi-task prompting settings, this strategy decomposed the annotation task into four sequential prompts: cause-relation detection, cause-span extraction, cause-category classification, and causal-marker identification. The purpose of this experiment was to examine whether breaking the task into simpler subtasks would improve model performance without using any demonstration examples.

Overall, Gemini performed slightly better than ChatGPT on cause-presence detection and cause-category classification. For cause-presence detection, Gemini achieved an accuracy of 0.537 and a macro-F1 of 0.537, compared with ChatGPT's accuracy of 0.487 and macro-F1 of 0.481. A similar pattern was observed for cause-category classification. For causal-marker presence, ChatGPT outperformed Gemini. ChatGPT achieved an accuracy of 0.650 and macro-F1 of 0.568, whereas Gemini achieved an accuracy of 0.617 and macro-F1 of 0.500. However, both models showed weaker performance on the exact identification of causal marker text. Neither model achieved any exact matches for causal marker text, with exact match scores of 0.000 for both systems. The weakest results were observed for cause-span extraction. ChatGPT achieved an exact match score of 0.067 and token F1 of 0.111, while Gemini achieved an exact match score of 0.106 and token F1 of 0.189. Gemini therefore performed better on span extraction, but the overall scores remained low. This indicates that zero-shot step-wise prompting did not reliably lead the models to copy the exact annotated cause span. The models may have extracted broader phrases, paraphrased the cause, or selected partially overlapping text rather than matching the human annotation exactly. Table 9 provides the summary of the results of the zero-shot step-wise prompting experiment.

Table 9. Summary of the results of the zero-shot step-wise prompting experiment.

Task	Best Model	Key Result	Interpretation
Cause Presence/ Cause Expressed	Gemini	Accuracy = 0.537; Macro-F1 = 0.537	Gemini was slightly better at detecting whether the tweet expressed a cause for the sick-leave request.
Cause Category	Gemini	Accuracy = 0.490; Macro-F1 = 0.335	Gemini performed better, but category classification remained difficult for both models.
Causal-Marker Presence	ChatGPT	Accuracy = 0.650; Macro-F1 = 0.568	ChatGPT was better at identifying whether the cause was explicit or implicit.
Cause Span Text Matching	Gemini	Exact Match = 0.106; Token F1 = 0.189	Gemini extracted cause spans better, but exact span matching was still weak.
Causal Marker Text Matching	Gemini	Exact Match = 0.000; Token F1 = 0.060	Both models failed on exact marker matching; Gemini had slightly higher token overlap.

Compared with the zero-shot multi-task prompt, the zero-shot step-wise prompting strategy generally produced weaker results. For cause-presence detection, both GPT and Gemini performed better under the multi-task setting, with Gemini achieving the strongest overall performance. The difference was especially clear for text extraction tasks. Cause-span extraction dropped sharply under the step-wise approach, with GPT decreasing from 0.574 exact match to 0.067 and Gemini decreasing from 0.645 to 0.106. A similar pattern was observed for causal marker text extraction, where the multi-task prompt produced high exact-match scores for both GPT and Gemini, while the step-wise approach produced no exact matches.

4.3.2. Few-Shot Performance

The few-shot experiments used prompts containing annotated examples. Four versions of the few-shot prompt were tested to examine the effect of example selection and class imbalance on model performance. These versions varied in the composition of examples, particularly in how they represented underrepresented labels and less frequent annotation cases.

The few-shot experiments tested whether labeled examples in the prompt help the model improve over zero-shot performance, and whether example balance affects results under class imbalance. The same fixed test set was used across all experiments consisting of 300 test samples. The experiments were:

- E1: 10-shot natural distribution: Ten labeled examples were randomly selected from the training pool, preserving the dataset's original imbalance.
- E2: 10-shot balanced by cause presence: Ten examples were used: five cause-present and five cause-absent.
- E3: 10-shot minority-focused: Ten examples were selected with more emphasis on the minority class, seven minority examples and three majority examples.
- E4: 20-shot balanced by cause presence: Twenty examples were used: ten cause-present and ten cause-absent.

Table 10 provides a summary of the experiments and their results.

Table 10. Few-shot classification performance of GPT-4.1-mini and Gemini-2.5-flash across E1–E4.

Task	Experiment	GPT Accuracy	GPT Macro Precision	GPT Macro-F1	GPT Weighted-F1	Gemini Accuracy	Gemini Macro Precision	Gemini Macro-F1	Gemini Weighted-F1
Cause Presence	E1 Natural 10-shot	0.575	0.588	0.536	0.619	0.767	0.630	0.634	0.771
Cause Presence	E2 Balanced 10-shot	0.568	0.573	0.525	0.614	0.673	0.616	0.606	0.708
Cause Presence	E3 Minority-focused 10-shot	0.525	0.587	0.502	0.569	0.546	0.614	0.524	0.589
Cause Presence	E4 Balanced 20-shot	0.541	0.583	0.511	0.586	0.636	0.609	0.582	0.677
Cause Category	E1 Natural 10-shot	0.489	0.041	0.035	0.561	0.739	0.218	0.241	0.724
Cause Category	E2 Balanced 10-shot	0.521	0.192	0.157	0.588	0.626	0.410	0.417	0.678
Cause Category	E3 Minority-focused 10-shot	0.459	0.220	0.194	0.536	0.508	0.354	0.348	0.584
Cause Category	E4 Balanced 20-shot	0.494	0.333	0.313	0.574	0.608	0.278	0.294	0.668
Causal-Marker Presence	E1 Natural 10-shot	0.740	0.578	0.578	0.783	0.877	0.619	0.569	0.856
Causal-Marker Presence	E2 Balanced 10-shot	0.660	0.559	0.530	0.724	0.838	0.610	0.621	0.847
Causal-Marker Presence	E3 Minority-focused 10-shot	0.560	0.558	0.479	0.640	0.768	0.552	0.555	0.797
Causal-Marker Presence	E4 Balanced 20-shot	0.604	0.567	0.509	0.679	0.768	0.596	0.605	0.805

Compared with GPT, Gemini showed stronger few-shot classification performance across most tasks and configurations. For cause-presence detection, Gemini consistently achieved higher accuracy, macro precision, macro-F1, and weighted-F1 than GPT, with its strongest result under E1. This pattern is consistent with the zero-shot results, where Gemini also outperformed GPT on cause presence, achieving higher accuracy, macro precision, yes-class precision, macro-F1, and weighted-F1. For cause-category classification, Gemini also performed better overall in the few-shot experiments, particularly in accuracy and weighted-F1, although both models continued to struggle with macro-level performance because of label imbalance. This differs slightly from the zero-shot results: while Gemini already had higher accuracy and weighted-F1 than GPT in zero-shot category classification, both models had extremely low macro-F1-scores (0.068 for Gemini and 0.064 for GPT). In the few-shot setting, macro-level category performance improved more clearly, especially when the examples were balanced or minority-focused. For causal-marker presence, the comparison is more nuanced. In zero-shot, GPT had slightly higher accuracy (0.854 vs. 0.838), but Gemini had better macro precision, yes-class precision, macro-F1, and weighted-F1, indicating more balanced performance on the minority “yes” class. In the few-shot experiments, Gemini generally surpassed GPT on this task as well, especially in weighted-F1 and macro-F1. Overall, the few-shot results strengthen the pattern already visible in zero-shot: Gemini tends to produce better and more balanced classification performance, while GPT is more affected by label imbalance and, in some cases, achieves competitive accuracy mainly by favoring majority-class predictions. Table 11 summarizes these experiments and their results.

Table 11. Few-shot text-matching performance of GPT-4.1-mini and Gemini-2.5-flash across E1–E4.

Task	Experiment	GPT Exact Match	GPT Avg. Token Overlap	Gemini Exact Match	Gemini Avg. Token Overlap
Cause Span Text	E1 Natural 10-shot	0.460	0.488	0.711	0.721
Cause Span Text	E2 Balanced 10-shot	0.457	0.485	0.576	0.596
Cause Span Text	E3 Minority-focused 10-shot	0.402	0.434	0.432	0.463
Cause Span Text	E4 Balanced 20-shot	0.425	0.456	0.539	0.567
Causal Marker Text	E1 Natural 10-shot	0.692	0.695	0.871	0.871
Causal Marker Text	E2 Balanced 10-shot	0.613	0.619	0.810	0.815
Causal Marker Text	E3 Minority-focused 10-shot	0.500	0.506	0.746	0.749
Causal Marker Text	E4 Balanced 20-shot	0.547	0.551	0.727	0.732

Across the few-shot text-matching experiments, Gemini consistently outperformed GPT on cause-span extraction, while GPT remained more competitive on causal marker text matching. For cause span text matching, Gemini achieved higher exact-match and token-overlap scores than GPT across all E1–E4 configurations, with its best performance under E1. This follows the same pattern observed in the zero-shot setting, where Gemini also outperformed GPT on cause-span matching, achieving an exact-match score of 0.645 and token-overlap score of 0.662, compared with GPT’s 0.574 and 0.595. However, Gemini’s strongest few-shot result, especially under E1, exceeded its zero-shot cause-span performance.

For causal marker text matching, the pattern was different. In the zero-shot setting, GPT performed slightly better than Gemini, with an exact-match and token-overlap score of 0.849, compared with Gemini's 0.825 and 0.826. In the few-shot setting, however, Gemini outperformed GPT across all E1–E4 configurations, with its strongest performance again under E1. This suggests that Gemini benefited more from the few-shot examples for marker extraction, while GPT's few-shot marker-matching scores declined compared with its zero-shot result. Overall, the text-matching results indicate that Gemini was more responsive to few-shot prompting for extraction tasks, particularly under the natural-distribution E1 setup, whereas GPT's zero-shot performance was stronger for causal-marker matching but did not improve under the few-shot configurations.

4.3.3. Single-Tweet Processing vs. Batch Processing

The same prompting configurations were tested under two processing modes: single-tweet processing and batch processing. In the single-tweet setting, each tweet was submitted to the model individually. In the batch-processing setting, multiple tweets were submitted within the same prompt.

The results will compare the quality of annotations generated under both modes. Batch-processing performance will also be examined across different batch sizes to identify whether there is a point at which performance begins to decline. This analysis will consider both predictive performance and output reliability, including formatting errors, missing labels, and inconsistencies in structured responses. Tables 12 and 13 summarize the batch 10 and text-matching experiments and their results.

Table 12. Batch of 10 classification performance.

Task	Experiment	GPT Accuracy	GPT Macro-F1	GPT Weighted-F1	Gemini Accuracy	Gemini Macro-F1	Gemini Weighted-F1
Cause Presence	E1 Natural 10-shot	0.621	0.547	0.662	0.690	0.550	0.743
Cause Presence	E2 Balanced 10-shot	0.512	0.484	0.560	0.684	0.546	0.740
Cause Presence	E3 Minority-focused 10-shot	0.497	0.475	0.542	0.558	0.464	0.638
Cause Presence	E4 Balanced 20-shot	0.492	0.468	0.539	0.648	0.538	0.711
Cause Category	E1 Natural 10-shot	0.568	0.054	0.617	0.650	0.061	0.729
Cause Category	E2 Balanced 10-shot	0.450	0.172	0.533	0.658	0.178	0.730
Cause Category	E3 Minority-focused 10-shot	0.438	0.182	0.516	0.538	0.196	0.634
Cause Category	E4 Balanced 20-shot	0.449	0.247	0.532	0.625	0.226	0.707
Causal-Marker Presence	E1 Natural 10-shot	0.758	0.560	0.793	0.815	0.536	0.837
Causal-Marker Presence	E2 Balanced 10-shot	0.637	0.519	0.707	0.850	0.534	0.854
Causal-Marker Presence	E3 Minority-focused 10-shot	0.584	0.487	0.663	0.821	0.500	0.834
Causal-Marker Presence	E4 Balanced 20-shot	0.604	0.505	0.680	0.841	0.568	0.854

Table 13. Batch of 10 text-matching performance.

Task	Experiment	GPT Exact Match	GPT Token Overlap	Gemini Exact Match	Gemini Token Overlap
Cause Span Text Matching	E1 Natural 10-shot	0.537	0.552	0.640	0.649
Cause Span Text Matching	E2 Balanced 10-shot	0.391	0.421	0.631	0.644
Cause Span Text Matching	E3 Minority-focused 10-shot	0.385	0.412	0.512	0.523
Cause Span Text Matching	E4 Balanced 20-shot	0.374	0.403	0.591	0.603
Causal Marker Text Matching	E1 Natural 10-shot	0.730	0.732	0.799	0.803
Causal Marker Text Matching	E2 Balanced 10-shot	0.581	0.585	0.847	0.847
Causal Marker Text Matching	E3 Minority-focused 10-shot	0.525	0.529	0.817	0.817
Causal Marker Text Matching	E4 Balanced 20-shot	0.548	0.554	0.831	0.832

In the batch-of-10 setting, Gemini-2.5-flash outperformed GPT-4.1-mini across almost all classification and text-matching tasks. Compared with single-tweet processing, batch processing affected the two models differently. For GPT-4.1-mini, batching sometimes improved performance under the natural-distribution prompt, especially E1, where batch processing improved cause-presence accuracy, cause-category accuracy, causal-marker accuracy, cause-span matching, and causal-marker matching. However, GPT's performance generally declined under the balanced or minority-focused configurations, suggesting that batching increased instability when the prompt examples already pushed the model toward minority classes. Gemini, in contrast, remained stronger than GPT in batch mode, but its batch performance was often lower than its own single-tweet performance, especially under E1. For example, Gemini's single-tweet E1 cause-presence accuracy was 0.767, while batch E1 dropped to 0.690; its cause-span exact match dropped from 0.711 to 0.640. Overall, batch processing appears more efficient but can reduce performance, particularly for Gemini relative to its single-tweet results. However, even with this reduction, Gemini remained the stronger model overall in the batch-of-10 setting.

Another experiment was set up to find the best number of shots to use to get the best performance. The experiment was performed on Gemini since it achieved the best results in natural shot distribution. The results of the classification tasks, mainly cause expressed, show some variation across the 2–9 shot settings, but no steady improvement as the number of examples increases. Performance starts at 0.6066 with two shots and improves slightly with three and four shots, reaching 0.6263 and 0.6062, respectively. It then drops at five shots to 0.5974, before increasing again at six shots and reaching its highest value at seven shots with 0.6517. After that, performance declines at eight and nine shots, with nine shots producing the lowest score (0.5573). Compared with the 10-shot setting (0.6300), the 7-shot configuration performs better. Similar results were shown from the causal-marker presence with best performance was with six-shot examples. In the cause category attribute, the results show best performance with 10-shot examples. Figure 4 shows a summary of these values for the classification tasks.

In the case of text marching, the results are also inconsistent. For cause span text matching, the best performance was obtained with two-shot examples. Other performance results did not show much variation. For the causal marker text matching, the best performance was obtained though 10-shot examples with similar performance with 7-shot examples. Figure 5 shows the results of the experiments for text-matching tasks.

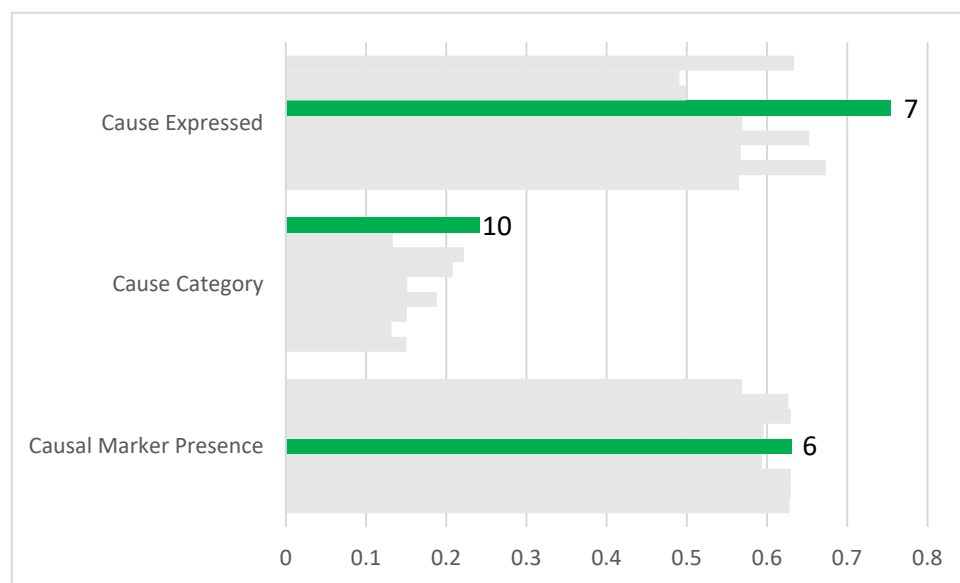


Figure 4. Gemini few-shot performance across 2–10-shot settings for classification tasks. The green color in each classification task shows the number of shots with best performance.

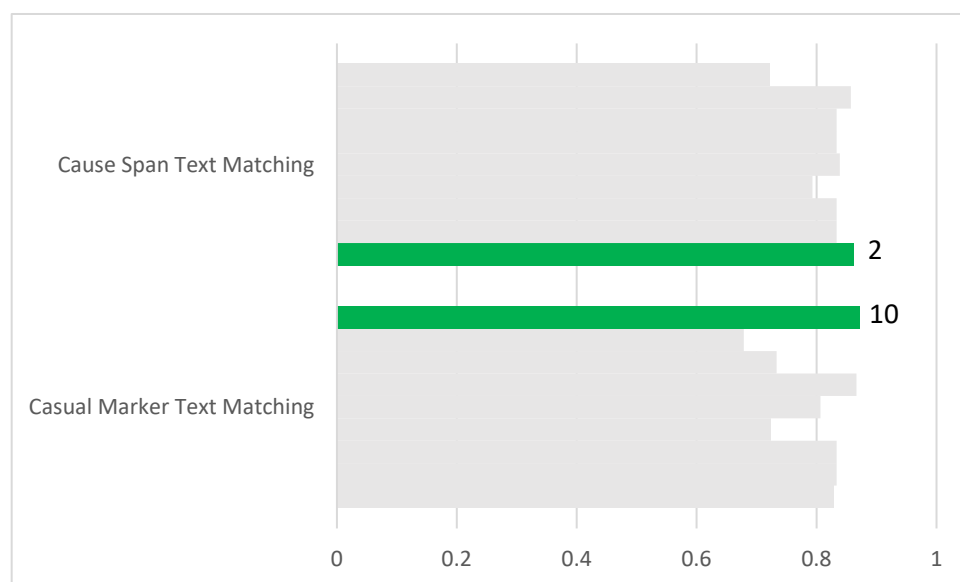


Figure 5. Gemini few-shot performance across 2–10-shot settings for text-matching tasks. The green color in each text-matching task shows the number of shots with best performance.

4.3.4. Comparison Between GPT-4.1-Mini and Gemini-2.5-Flash

The final LLM comparison examines differences between GPT-4.1-mini and Gemini-2.5-flash across all prompting configurations and processing modes. This comparison is intended to assess whether one model produced more accurate, stable, or consistently structured annotations than the other.

The comparison will consider classification performance, span-extraction quality, causal-marker identification, sensitivity to few-shot example composition, and robustness under batch processing.

The strongest overall configuration was Gemini-2.5-flash with E1 natural 10-shot examples under single-tweet processing. This setting achieved the best overall performance across cause-presence detection, cause-span extraction, and causal marker text matching. However, for cause-category classification specifically, the best result was obtained by

Gemini E2 balanced 10-shot, suggesting that balanced examples improved the model's handling of fine-grained and imbalanced category labels. Table 14 summarizes the results.

Table 14. LLM comparison results across prompting configurations and processing modes.

Task	Metric Values
Cause Presence	Accuracy 0.767; Macro Precision 0.630; Macro-F1 0.634; Weighted-F1 0.771
Cause Category	Accuracy 0.739; Macro Precision 0.218; Macro-F1 0.241; Weighted-F1 0.724
Causal-Marker Presence	Accuracy 0.877; Macro Precision 0.619; Macro-F1 0.569; Weighted-F1 0.856
Cause Span Text Matching	Exact Match 0.711; Average Token Overlap 0.721
Causal Marker Text Matching	Exact Match 0.871; Average Token Overlap 0.871

One important exception: for cause category macro-F1, the best value was not Gemini E1. It was Gemini E2 Balanced 10-shot, with: accuracy 0.626; macro precision 0.410; macro-F1 0.417; weighted-F1 0.678.

4.4. Summary of Quantitative Findings

This subsection summarizes the main findings across the two modeling paradigms evaluated in this study. The comparison is organized in three parts: first, the performance of the fine-tuned BERT-based models is compared; second, the LLM-based results are summarized by comparing Gemini-2.5-flash and GPT-4.1-mini; and finally, the BERT-based and LLM-based approaches are compared more broadly.

The fine-tuned BERT-based results show that SaudiBERT generally provided stronger performance than AraBERT, particularly when macro-level and minority-class performance are considered. SaudiBERT was more effective in identifying tweets where a cause was actually expressed. It achieved a higher recall for the cause-present class (0.4615) than AraBERT (0.3385), as well as a higher cause-present F1-score (0.3390 compared with 0.2993). SaudiBERT also achieved a higher macro-F1-score (0.6417) than AraBERT (0.6256).

For explicit/implicit causality classification, SaudiBERT clearly outperformed AraBERT. SaudiBERT achieved an accuracy of 0.7246 and a macro-F1-score of 0.7237, compared with AraBERT's accuracy of 0.6812 and macro-F1 of 0.6795. The improvement was observed for both classes. For implicit causality, SaudiBERT achieved an F1-score of 0.7077, compared with 0.6562 for AraBERT. For explicit causality, SaudiBERT achieved an F1-score of 0.7397, compared with 0.7027 for AraBERT.

The LLM-based experiments compared GPT and Gemini across prompting configurations and processing modes. This comparison considered classification performance, span-extraction quality, causal-marker identification, sensitivity to few-shot example selection, and robustness under batch versus single-tweet processing.

Across the LLM experiments, Gemini produced the strongest overall results. The best overall configuration was Gemini with E1 natural 10-shot examples under single-tweet processing. This setting achieved the strongest combined performance across several tasks, including cause-presence detection, cause-span extraction, and causal marker text matching.

Under this configuration, Gemini achieved 0.767 accuracy and 0.634 macro-F1. For causal-marker presence, it achieved 0.877 accuracy and 0.569 macro-F1. The same setting also performed well on text-span tasks, achieving 0.711 exact match and 0.721 average token overlap for cause span text matching, and 0.871 exact match and 0.871 average token overlap for causal marker text matching.

However, explicit/implicit classification showed a different pattern. Although Gemini E1 natural 10-shot achieved an accuracy of 0.739, its macro-F1 was relatively low (0.241), indicating continued difficulty with imbalanced and fine-grained category labels. The best

macro-F1 for explicit/implicit classification was instead achieved by Gemini E2 balanced 10-shot, with 0.626 accuracy and 0.417 macro-F1.

Comparing the results of SaudiBERT and Gemini based on the classification tasks, for cause-presence detection, SaudiBERT achieved a higher accuracy (0.8975) than the strongest LLM configuration reported for this task (0.767 for Gemini E1 natural 10-shot). SaudiBERT also achieved a slightly higher macro-F1-score (0.6417) compared with Gemini's 0.634.

5. Discussion

5.1. Quantitative Results of BERTs and LLMs

Prior work has shown strong performance for deep-learning models on explicit with reported macro-F1-scores reaching approximately 0.88–0.89 for BERT-based causal/non-causal classification, while token-level extraction models such as SpanBERT and DistilBERT have achieved macro-F1 values up to 0.89 [10]. In contrast, the present results were lower, particularly for cause-presence detection and cause-category classification. For example, SaudiBERT achieved a macro-F1 of 0.6417 for cause-presence detection, while the strongest LLM setting achieved a macro-F1 of 0.634 on the same task. This difference is likely explained by the nature of the dataset: unlike benchmark corpora, the present study focuses on informal Saudi dialect tweets, where causal meanings are often implicit and short. Tasks involving clearer classification boundaries, such as cause-presence detection and causal-marker presence, generally produced stronger results than fine-grained cause-category classification (explicit vs. implicit). This was especially evident in the low macro-F1-scores for the cause-category task, where category imbalance and semantic overlap between reasons likely made classification more difficult. In contrast, tasks involving explicit markers were comparatively easier, particularly when the model could rely on visible causal signals in the text. However, the results also show that causal reasoning in informal Arabic social media data cannot be reduced to explicit connectives alone. Many tweets expressed causality indirectly through context or implied motivations, which makes implicit causal reasoning a central challenge.

This challenge is particularly important in Arabic. Compared with prior Arabic causality studies that focused on explicit MSA causality using rule-based or morphology-based methods [2,14], the present results show that dialectal causality is more difficult because causal meaning is often not always signaled by clear connectives. This helps explain why causal-marker detection was more stable than implicit cause interpretation. These systems often achieve high precision, usually above 80%, but lower recall. The present dataset differs from this setting because it contains informal Saudi dialectal tweets, where causal meanings may be expressed through colloquial wording. This partly explains why explicit-marker detection was more stable than cause-category classification and why dialect-specialized modeling with SaudiBERT was beneficial.

The comparison between SaudiBERT and AraBERT supports this point. Although AraBERT achieved slightly higher overall accuracy in cause-presence detection, SaudiBERT performed better on the analytically important minority class, namely tweets where a cause was expressed. SaudiBERT achieved higher recall and F1-score for the cause-present class, as well as a higher macro-F1-score overall. It also outperformed AraBERT in explicit/implicit classification. SaudiBERT appears better able to capture implicit causal expressions that may be less represented in AraBERT, which was trained on MSA. This finding is consistent with the motivation behind SaudiBERT [8], which shows the importance of dialect-specific pretraining for Saudi Arabic NLP. However, the present study extends this line of work from general dialectal classification tasks to event-conditioned causal interpretation.

At the same time, the BERT results show the limits of supervised classification models in this study. The BERT-based models were effective for predefined classification tasks, but they were not used for span extraction because the experimental setup treated them as sequence classifiers. Applying BERT to cause-span extraction or causal marker text extraction would require a different architecture and annotation format. This would also require a separate evaluation pipeline. In contrast, the LLMs were able to perform both classification and free-text span identification through prompting, which make them more flexible across task types.

The LLM results show both the promise and instability of prompting-based causality extraction. Recent studies have examined LLMs for causal reasoning tasks without task-specific supervision. Prior studies found that instruction-based (zero-shot vs. few-shot) and chain-of-thought (step-wise vs. multi-task) prompting can affect causal inference [15,16]. This pattern was also observed in the present study: Gemini-2.5-flash produced the strongest overall LLM performance, particularly under the E1 natural 10-shot single-tweet configuration. This setting performed well for cause-presence detection, cause-span extraction, causal-marker presence, and causal marker text matching. However, cause-category (explicit vs. implicit) classification remained difficult. The best macro-F1 for cause category was obtained not by the strongest overall configuration, but by the E2 balanced 10-shot setting. This may be attributed to the balanced examples, which could have helped the model better recognize underrepresented categories, although the overall accuracy decreased. While few-shot settings improved the performance, this was not entirely significant. The improvement was minimal and could be disregarded. These findings are also consistent with recent Arabic NLP studies showing the growing use of zero-shot and prompt-based LLM methods for classification tasks [10,19]. However, unlike these studies, the present work focuses on causality extraction, which adds another layer of complexity.

The comparison between zero-shot, step-wise, and few-shot prompting further suggests that more structured prompting does not automatically lead to better results. Although step-wise prompting was intended to decompose the annotation process into simpler stages, the results indicate that separating the task into sequential prompts did not improve extraction accuracy. One likely explanation is error propagation: if the model made an incorrect decision in an early step, such as failing to detect a cause, later steps such as span extraction or category classification were affected. Therefore, it can be concluded that step-wise prompting can introduce additional inconsistency.

Overall, the findings suggest a trade-off between BERT-based classification and LLM-based prompting. SaudiBERT performed better for supervised classification, particularly when macro-F1 and minority-class recall were important. In contrast, LLMs were more flexible for both classification and span extraction, but their performance was more sensitive to prompt design.

These findings should be interpreted within the scope of the present dataset and experimental design. The results reflect Saudi-dialect Twitter data. Broader generalization to other Arabic dialects, genres, or causal extraction settings requires further evaluation.

5.2. Qualitative Interpretation of LLMs Output

To complement the quantitative evaluation, a qualitative analysis was conducted to examine how Gemini and ChatGPT behaved under zero-shot prompting. The qualitative observations were organized into five main areas of concern:

1. Prediction stability across repeated and reformulated tweets.
2. Cause presence errors and downstream error propagation.
3. Span boundary variation in causal text extraction.

4. Cause category confusion and schema violations.
5. Linguistic/dialectal challenges and causal marker extraction.

The first area concerns prediction stability across repeated and reformulated tweets. In several cases, highly similar tweets received different labels for cause expressed, with some instances labeled as *yes* and others as *no*. This inconsistency was observed in the outputs of both Gemini and ChatGPT. At the same time, the models showed some ability to recognize reformulated ideas. For example, when a non-causal tweet was repeated in several different forms, both models consistently labeled the reformulated versions as *no* for causal presence. This suggests that the models were not relying only on exact wording and could sometimes identify equivalent meanings across different formulations. However, this stability was weaker when the reformulated tweets contained causal meaning. In such cases, changes in wording sometimes confused the models and led to incorrect labeling across all labels.

The second area relates to cause presence errors and downstream error propagation. In several cases where the correct gold label for cause expressed was *no*, the models incorrectly predicted *yes*. Once this initial error occurred, the models often proceeded to generate additional incorrect annotations. This pattern was observed in both Gemini and ChatGPT and shows that cause-presence detection functions as a gatekeeping task in the annotation process. When the model incorrectly identifies a tweet as causal, the error is likely to propagate to the remaining subtasks.

The third area concerns span boundary variation in causal text extraction. The qualitative review showed that model-generated causal spans often differed from the gold annotations in length and boundary selection. In many cases, the models captured the general causal meaning but extracted either more or fewer words than the human annotator. The most common pattern was span expansion, where the model included extra surrounding context in addition to the core cause. In other cases, the model extracted only part of the intended cause, resulting in a reduced span. These observations help explain why exact-match scores alone may underestimate model performance for span extraction. A prediction may fail to match the gold span exactly while still capturing the main causal meaning. Therefore, partial-matching measures are important for evaluating causal span extraction.

The fourth area includes cause category confusion and schema violations. Overall, both models were able to assign many tweets to reasonable broad cause categories. However, several errors were observed. A recurring issue was the failure to distinguish between medical condition and well-being. Although some tweets were labeled in the gold standard as well-being-related causes, the models did not assign the well-being category to any of the tweets. Instead, these cases were consistently classified as medical conditions. This means that the models interpreted well-being-related expressions as direct medical conditions and were not successful in separating physical illness from broader well-being explanations. In addition, both models occasionally assigned categories outside the predefined label set. These hallucinated labels indicate a failure to fully follow the annotation schema in the zero-shot setting.

The fifth area relates to linguistic/dialectal challenges and causal marker extraction. Some errors appeared to result from the models' difficulty in interpreting informal Arabic, especially when the dialectal expression was not a common or easily recognizable Saudi form. This is a challenge of applying general-purpose LLMs to dialectal social media text. Causal marker extraction was also affected by the rarity of explicit causality in the dataset. Since explicit causal markers appeared in only a minority of tweets, the models had relatively few clear linguistic cues to rely on. For the cases where a causal marker was present, Gemini was generally more successful in extracting the correct marker and showed closer

alignment with the gold standard. ChatGPT was less consistent in this task and sometimes produced words that were not causal markers.

Overall, the qualitative findings show that both Gemini and ChatGPT were capable of identifying many causal meanings in the tweets, particularly when the cause was explicit or clearly expressed. However, their outputs were affected by inconsistency across repeated and reformulated tweets, false positive cause detection, downstream error propagation, span boundary mismatch, category confusion, hallucinated labels, dialectal misunderstanding, and causal marker extraction errors. These findings help explain the quantitative results and support the use of both exact-match and partial-match measures when evaluating causal information extraction from informal Arabic social media text.

Together, these observations point to a broader set of qualitative concerns that can guide the interpretation of LLM-based annotation results. These concerns can be generalized into five categories:

1. Consistency: referring to whether models produce stable annotations across repeated or reformulated inputs;
2. Task dependency, referring to how errors in one annotation decision affect subsequent outputs;
3. Extraction granularity, referring to differences between model and human span boundaries;
4. Label interpretation, referring to confusion between overlapping categories or failure to follow the predefined schema; and
5. Linguistic sensitivity, referring to the model's ability to handle dialectal variation and implicit meanings.

6. Conclusions

This study shows that causality extraction in dialectal Arabic social media text requires models that can handle both linguistic variation and implicit causal meaning. The work addressed both classification and text-span identification tasks. The findings contribute to the broader literature on causality extraction by examining causal-reason annotation in a relatively underexplored setting which is informal Saudi Arabic social media text.

The results show that dialect-specific pretraining matters: SaudiBERT was more effective than AraBERT when minority-class and macro-level performance were prioritized. These findings suggest that Arabic causality extraction should move beyond explicit connective detection and MSA-focused resources toward dialect-trained models that can capture implicit reasons in informal text.

The LLM results show that prompting can extend causality extraction beyond classification to span and marker identification, but this flexibility comes with reduced stability and greater sensitivity to prompt design.

Possible directions for future work would be to combine the strengths of both approaches. For example, dialect-specific transformer models could be used for robust classification, while LLMs could support span proposal, explanation generation, or weak supervision. Future work could also explore token-level fine-tuning of SaudiBERT for cause-span and causal-marker extraction, allowing a more direct comparison between BERT-based and LLM-based approaches across all subtasks. In addition, repeated-run experiments over the full test set would provide a more comprehensive assessment of LLM prediction stability across prompting conditions. Expanding the dataset and improving the representation of minority cause categories would also support more reliable evaluation of causal interpretation.

Author Contributions: Conceptualization, M.E. and S.B.; methodology, M.E. and S.B.; validation, M.E.; formal analysis, M.E. and S.B.; investigation, M.E. and S.B.; resources, R.O. and S.E.; data curation, R.O. and S.E.; writing—original draft preparation, M.E. and S.B.; writing—review and editing, R.O. and S.E. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Mount Royal University Open Access Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study is openly available in Zenodo at <https://doi.org/10.5281/zenodo.20278150> (accessed on 19 May 2026).

Acknowledgments: Generative AI was also used as part of the experimental design and evaluation in this study. Specifically, GPT-4.1-mini and Gemini-2.5-flash were evaluated as prompting-base large language models for causal-reason annotation. The models were used to experiment under different prompting settings, including zero-shot, step-wise, and few-shot prompting. Their outputs were compared against the gold-standard annotated dataset using classification metrics and text-matching measures such as exact match and token overlap.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Prompting Templates

The following prompt templates were employed for LLM evaluation. Prompts were issued in English for control and consistency. All prompts explicitly stated that the sick-leave effect was implied by the hashtag and did not require extraction.

Appendix A.1. Structured Multi-Task Prompt

System Instruction:

You will be given a tweet written in Saudi dialect Arabic. The presence of a sick-leave-related hashtag means that the user intends to obtain a sick-leave. Your task is to analyze the tweet text.

Tweet:

[insert tweet text here]

Please perform the following tasks:

1. Decide whether the tweet expresses a cause for requesting the sick-leave. Answer “Yes” or “No”.
2. If Yes, copy the exact text span that expresses the cause.
3. Assign the cause to one general category from the following list:
 - ☐ Health/illness/medical condition
 - ☐ Psychological or emotional well-being
 - ☐ Caregiving/responsibility for another person
 - ☐ Lifestyle/social/entertainment/personal activity
 - ☐ General/other/not clearly specified
4. Decide whether the causality is explicit (a causal word is used) or implicit (implied without a causal word).
5. If explicit, identify the causal marker used (for example: “بسبب”, “علشان”, “عشان”, “كذا”, “عشان”, or similar).

Return the output in the following JSON structure:

```
{
  "cause_present": Yes/No,
  "cause_span": "...",
```

```

"cause_category": "...",
"causality_type": Explicit/Implicit,
"causal_marker": "...
}

```

If no cause is expressed, set all other fields to null.

Appendix A.2. Step-Wise Prompting Strategy

Step 1: Cause-Relation Detection

Instruction:

You will be given a tweet written in Saudi dialect Arabic. The hashtag means the user intends to obtain a sick-leave. Decide whether the tweet expresses a cause for this sick-leave request.

Tweet:

[insert tweet text]

Answer only with "Yes" or "No".

Step 2: Cause-Span Extraction (if Yes)

Instruction:

The tweet expresses a cause for requesting a sick-leave. Copy the exact text span that expresses the cause.

Tweet:

[insert tweet text]

Step 3: Cause-Category Classification

Instruction:

Assign the cause to one of the following general categories:

- Health/illness/medical condition
- Psychological or emotional well-being
- Caregiving/responsibility for another person
- Lifestyle/social/entertainment/personal activity
- General/other/not clearly specified

Output only the category label.

Step 4: Causal-Marker Identification

Instruction:

Decide whether the cause is expressed using an explicit causal marker (for example: "عشان كذا", "بسبب", "عشان", "علشان", or similar).

If explicit, name the marker.

If implicit, answer "Implicit".

References

1. Elhussein, M.; Brahimi, S. Clustering as Feature Selection Method in Spam Classification: Uncovering Sick-Leave Sellers. *Appl. Comput. Inform.* **2025**, *21*, 232–243. [CrossRef]
2. Sadek, J.; Meziane, F. Extracting Arabic Causal Relations Using Linguistic Patterns. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2016**, *15*, 14. [CrossRef] [PubMed]
3. Blanco, E.; Castell, N.; Moldovan, D. Causal Relation Extraction. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*; Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Tapias, D., Eds.; European Language Resources Association (ELRA): Marrakech, Morocco, 2008.
4. Chen, P.; Liu, K.; Chen, Y.; Wang, T.; Zhao, J. Probing into the Root: A Dataset for Reason Extraction of Structural Events from Financial Documents. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 2042–2048.

5. Al-Saif, A.; Markert, K. The Leeds Arabic Discourse Treebank: Annotating Discourse Connectives for Arabic. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*; European Language Resources Association (ELRA): Valletta, Malta, 2010.
6. El-Naggar, N.; El-Sonbaty, Y.; El-Nasr, M.A. Sentiment Analysis of Modern Standard Arabic and Egyptian Dialectal Arabic Tweets. In *Proceedings of the 2017 Computing Conference*; IEEE: New York, NY, USA, 2017; pp. 880–887.
7. Moussa, H.N.; Mourhir, A. DarNERcorp: An Annotated Named Entity Recognition Dataset in the Moroccan Dialect. *Data Brief* **2023**, *48*, 109234. [[CrossRef](#)] [[PubMed](#)]
8. Qarah, F. SaudiBERT: A Large Language Model Pretrained on Saudi Dialect Corpora. *arXiv* **2024**. [[CrossRef](#)]
9. Doan, S.; Yang, E.W.; Tilak, S.S.; Li, P.W.; Zisook, D.S.; Torii, M. Extracting Health-Related Causality from Twitter Messages Using Natural Language Processing. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 79. [[CrossRef](#)] [[PubMed](#)]
10. Yang, J.; Han, S.C.; Poon, J. A Survey on Extraction of Causal Relations from Natural Language Text. *Knowl. Inf. Syst.* **2022**, *64*, 1161–1186. [[CrossRef](#)]
11. Khetan, V.; Ramnani, R.; Anand, M.; Sengupta, S.; Fano, A.E. Causal BERT: Language Models for Causality Detection between Events Expressed in Text. In *Intelligent Computing: Proceedings of the 2021 Computing Conference*; Springer International Publishing: Cham, Switzerland, 2021.
12. Li, Z.; Ding, X.; Liao, K.; Qin, B.; Liu, T. CausalBERT: Injecting Causal Knowledge Into Pre-Trained Models with Minimal Supervision. *arXiv* **2021**. [[CrossRef](#)]
13. Gopalakrishnan, S.; Chen, V.Z.; Dou, W.; Hahn-Powell, G.; Nedunuri, S.; Zadrozny, W. Text to Causal Knowledge Graph: A Framework to Synthesize Knowledge from Unstructured Business Texts into Causal Graphs. *Information* **2023**, *14*, 367. [[CrossRef](#)]
14. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Adv. Neural Inf. Process. Syst.* **2023**, *35*, 24824–24837.
15. Wang, Z. CausalBench: A Comprehensive Benchmark for Evaluating Causal Reasoning Capabilities of Large Language Models. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*; Wong, K.-F., Zhang, M., Xu, R., Li, J., Wei, Z., Gui, L., Liang, B., Zhao, R., Eds.; Association for Computational Linguistics: Bangkok, Thailand, 2024; pp. 143–151.
16. Li, G.; Wang, P.; Ke, W. Revisiting Large Language Models as Zero-Shot Relation Extractors. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 6877–6892.
17. Tan, F.A.; Hettiarachchi, H.; Hürriyetoglu, A.; Caselli, T.; Uca, O.; Liza, F.F.; Oostdijk, N. Event Causality Identification with Causal News Corpus—Shared Task 3, CASE 2022. In *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-Political Events from Text (CASE)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022.
18. Youssef, L.; Elhoussaine, Z.; Soufiane, N.; Noureddine, M. Enhancing Arabic Aspect Category Detection Using Large Language Models (LLMs). *Results Eng.* **2025**, *26*, 105049. [[CrossRef](#)]
19. Fairouz Chowdhury, A.; Sagor Chowdhury, M. Quasar at AraHealthQA Track 1: Leveraging Zero-Shot Large Language Models for Question and Answer Categorization in Arabic Mental Health. In *Proceedings of the Third Arabic Natural Language Processing Conference: Shared Tasks*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 155–163.
20. Sadek, J.; Meziane, F. Learning Causality for Arabic—Proclitics. *Procedia Comput. Sci.* **2018**, *142*, 141–149. [[CrossRef](#)]

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.