LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
CENTRUM FÜR INFORMATIONS- UND SPRACHVERARBEITUNG
STUDIENGANG COMPUTERLINGUISTIK

Bachelorarbeit

im Studiengang Computerlinguistik

an der Ludwig-Maximilians-Universität München

Fakultät für Sprach- und Literaturwissenschaften

Analyzing the Impact of Date Formats on Temporal Reasoning in Language Models

vorgelegt von
Hanna Kulik

Betreuer:	Zeinab Sadat Taghavi
Prüfer:	Prof. Dr. Hinrich Schütze
Bearbeitungszeitraum:	02.04.2025 – 11.06.2025

Abstract

Pretrained Large Language Models often show limitations in temporal reasoning, particularly when processing time-dependent facts. This thesis investigates a potentially overlooked factor influencing their performance: the syntactic format of date expressions. The central research question explores how different date formats affect LLMs' accuracy on temporal reasoning benchmarks.

To address this, the study systematically evaluated the Llama 3.2 3B Instruct model on the date arithmetic tasks of the TEMPReason benchmark. Questions were presented with date and temporal offset information in four distinct syntactic formats, ranging from naturalistic language to structured ISO 8601 representations. Model performance was assessed using Exact Match accuracy, Mean Absolute Error in years and Trend Accuracy.

The findings reveal that date format does impact the model's precision. While the Llama 3.2 3B Instruct model demonstrated robust understanding of temporal directionality across all formats, its accuracy in precise date arithmetic varied. A structured ISO format for base dates ("YYYY-MM") achieved the highest EM accuracy (33.55%). On the other hand, a fully natural language format for both base date and offset resulted in the lowest EM (31.55%). Remarkably, a fully word-based format, despite average EM accuracy, produced the lowest MAE-Year (18.68), suggesting different forms of errors based on input style.

This research concludes that the syntactic representation of dates is a significant factor in LLM temporal reasoning, not merely a superficial detail. The results highlight the need for greater consideration of format diversity in benchmark design and prompt engineering, and highlight ongoing challenges in developing LLMs with a more abstract and robust temporal comprehension.

Contents

1. Introduction

Large Language Models (LLMs) have revolutionized the field of Natural Language Processing (NLP), demonstrating impressive capabilities in tasks such as text generation, summarization, translation, and question answering. Their effectiveness comes from being pretrained on vast amounts of textual data, enabling them to learn complex patterns in language. However, despite their successes, a crucial aspect of human-like understanding where LLMs often struggle is temporal reasoning. Temporal reasoning is a cognitive and computational ability to understand, represent, and reason about temporal information. This includes identifying the order of events, e.g., whether one event occurred before, after, or simultaneously with another, estimating durations, and interpreting temporal expressions (Pustejovsky, 2012). This skill, fundamental for daily tasks such as understanding narratives, formulating plans, and following time-sensitive instructions is intuitive to humans but remains a significant challenge for language models to handle effectively (Vaswani et al., 2017; Raffel et al., 2020; Brown et al., 2020). For example, when someone says "the meeting was delayed by two days" after mentioning "it was planned for Monday," most people understand it now takes place on Wednesday. On the other hand, language models can struggle with this kind of reasoning if the time references aren't clearly spelled out. This gap highlights the challenge in producing models with robust, human-like temporal understanding, which is critical for a wide range of applied tasks.

1.1. Motivation

Temporal reasoning is a fundamental cognitive ability that allows humans to perceive time, interpret the temporal structure of events, and understand everyday communication (Allen, 1983; Nebel and Bürckert, 1995); it is essential for interpreting the temporal aspects of human experiences. In the computational context, this capacity is equally important, forming the basis for diverse applications like historical question answering, automated summarization and schedule generation. Though it is fundamental, current LLMs often struggle in time-sensitive reasoning tasks. The domain's inherent complexity, which demands implicit logical inference and broad world knowledge, has limited the number of models demonstrating strong proficiency. Very large models such as GPT-4 and to a lesser extent, Gemini and larger Llama 2 variants like the 70B model, have shown relatively better performance on some temporal reasoning benchmarks. However, LLMs in general still struggle with tasks that require understanding chronological order, computing durations, or interpreting relative temporal expressions, often producing inconsistent or inaccurate results (Wang and Zhao, 2024; Chu et al., 2024). These inconsistencies can manifest as errors in date arithmetic, misinterpretations of relative temporal adverbs such as "recently", "soon", or an inability to maintain temporal coherence over longer texts. Such failures are not only academic; they can significantly lower the reliability of LLMs in applications like news summarization, historical analysis, or automated planning where temporal accuracy is important. This highlights a major limitation in their ability to replicate human-like reasoning and poses challenges for building more reliable AI systems.

Recent efforts, such as the TEMPReason benchmark introduced by Tan et al. (2023a), have begun to systematically measure and analyze these deficiencies. It is focused on comprehensive coverage of temporal reasoning levels and a wider and more balanced time span, which addresses limitations in earlier Time-Sensitive Question Answering (TSQA) datasets. Although TEMPRea-

son covers multiple reasoning levels, it standardizes date surface forms, implicitly assuming that format does not affect performance.

This oversight points to another potentially important factor influencing LLM performance in temporal reasoning: the variation in the format of the date. In natural language, temporal expressions can take multiple syntactic forms, for example, "January 2020", "2020-01", "the first month of 2020" or "Jan. '20". It is hypothesized that the model's performance may depend significantly on such formatting variations. For instance, common tokenization strategies employed by LLMs, such as Byte Pair Encoding (BPE) or WordPiece, might process certain date formats more effectively than others, potentially due to their statistical prevalence in pretraining corpora (Kassner et al., 2021; Wang and Zhao, 2024). Some date representations could align well with the model's learned patterns, while alternative representations for the same date might introduce ambiguity or tokenization complexities. This can lead to inconsistent token sequences for semantically identical temporal anchors. As a result, an LLM might struggle to apply its temporal reasoning skills consistently when faced with these minor surface-level changes, even if its underlying understanding of the temporal concepts is sound.

The precise impact of date-format variation on tasks like date arithmetic or event ordering remains under-explored in the literature. Gaining a deeper understanding of how date representation influences temporal reasoning can yield important insights into model behavior, help diagnose specific failure points, and inform new strategies for improving temporal comprehension in LLMs. Such improvements would ultimately contribute to more robust and reliable AI systems for time-dependent applications. Therefore, this thesis aims to isolate date-format variation as an independent variable and systematically quantify its effect on LLM accuracy in temporal reasoning tasks, addressing this identified gap.

1.2. Research Question

The main research question for this thesis is: how do different date formats affect LLMs' accuracy on the TEMPReason L1 benchmark? Investigating this question is crucial because understanding the influence of input representation on temporal reasoning can reveal inherent model biases and guide the development of more robust AI systems.

The primary aim is to evaluate the extent to which syntactic variation in temporal expressions, given earlier in Section 1.1, impacts model performance on tasks requiring chronological understanding and date arithmetic. By systematically varying these formats within the established benchmark tasks, the goal is to identify format-specific changes in model accuracy (measured by Exact Match, Mean Absolute Error in years, and Trend Accuracy) and to determine whether specific syntactic forms consistently correlate with lower performance or more frequent inconsistent outputs.

Ultimately, the findings from this thesis aim to provide concrete evidence of such format-related biases, thereby providing practical recommendations for improving both the evaluation and the capabilities of LLMs in temporal tasks.

1.3. Scope and Objectives

This study evaluates the impact of date format variation on the temporal reasoning capabilities of LLMs. Llama 3.2 3B Instruct is used as the primary model. The experiments are conducted exclusively using English text and are specifically limited to the Level 1 (L1) subset of the TEMPReason benchmark, which focuses on date arithmetic tasks. The analysis takes into account the possible impact of instruction fine-tuning.

This research does not extend to:

- Multilingual contexts and multimodal temporal reasoning
- Investigation of pretraining data characteristics or knowledge cut-off effects

Objectives

- To systematically evaluate the performance of the selected LLM on temporal reasoning tasks across diverse date formats, using Exact Match, Mean Absolute Error in years and Trend Accuracy as evaluation metrics.
- To determine if specific date formats correlate with significantly higher or lower accuracy in temporal question answering outcomes.
- To identify and characterize potential weaknesses or biases in the LLMs concerning their sensitivity to superficial syntactic variations in temporal expressions.

This thesis aims to contribute a nuanced understanding of LLM temporal reasoning by demonstrating the impact of date format variation. By exposing performance limitations often hidden or underestimated in standardized evaluations, the findings aim to inform the development of more comprehensive benchmarks and to provide guidance for building LLMs that are more robust to diverse temporal expressions in time-sensitive tasks.

2. Background and Related Work

To effectively analyze the impact of date formats on temporal reasoning in LLMs, this chapter first establishes a foundational understanding of the core concepts and surveys the existing research landscape. First, we define temporal reasoning, outlining its various facets, and discuss the specific challenges LLMs face in this domain. Subsequently, the chapter reviews relevant related work, examining current approaches to evaluating temporal reasoning in LLMs, research into their handling of numerical data, and studies on their sensitivity to input representation. This exploration will provide the necessary context for the methodology and discussion presented in the following chapters and highlight the specific research gap this thesis aims to address.

2.1. Foundational Concepts

Before reviewing existing research, it is important to define the key terms and concepts that will be used throughout this thesis.

2.1.1. Large Language Models (LLMs)

Large Language Models are a class of neural networks, mainly based on the Transformer architecture (Vaswani et al., 2017), that are pretrained on exceptionally large and diverse text corpora. This pretraining enables them to learn patterns, acquire a broad range of world knowledge, and generate coherent, contextually relevant text. Two key characteristics of LLMs are particularly relevant to their capacity for temporal reasoning and therefore to the focus of this thesis:

- **Implicit Knowledge Representation**: Unlike traditional knowledge bases that store facts in an explicit structured format, LLMs encode knowledge implicitly within their vast network of parameters (Petroni et al., 2019). This means that temporal facts, such as the date of an event or the holder of a position at a specific time, are not stored as discrete entries but are learned as statistical correlations within the training data. This implicit encoding can make precise retrieval and consistent application of specific temporal information difficult, since it relies on the model's ability to correctly activate and synthesize relevant learned patterns based on the input query.
- **Knowledge Cutoff**: The knowledge assimilated by an LLM is static, reflecting the state of the world up to its last training data snapshot. Consequently, LLMs are typically unaware of events or changes to factual information that have occurred since their pretraining was completed (Touvron et al., 2023). This limitation is particularly relevant for tasks involving time-dependent facts, as the model may provide outdated information if asked about periods beyond its available data.

2.1.2. Temporal Reasoning

Temporal reasoning is the fundamental cognitive ability to understand, represent, and infer relationships concerning time, events, durations, and correlations (Allen, 1983). It ranges from simple event ordering to comprehending complex expressions, e.g., "next Tuesday", and is crucial for both human cognition and intelligent systems that process narratives or time-dependent data.

Levels of Temporal Understanding

Temporal reasoning involves a range of tasks, from simple date comparisons to complex event sequencing, each presenting different levels of difficulty. As shown in Figure 2.1, Tan et al. (2023a) structure human understanding of temporal reasoning into three levels: L1 — time-time, L2 — time-event, and L3 — event-event relations.

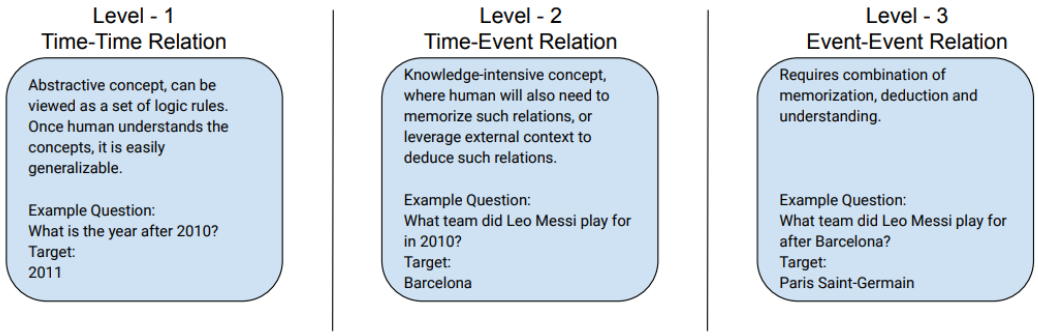

Figure 2.1.: Illustration of three levels of understanding towards time (adapted from Tan et al. (2023a).

Based on this categorization, the **Level 1 (L1)** or **Time-Time Relation** primarily involves the ability to "determine the relation between two timestamps t1 and t2 on the time axis". This level covers abstract, rule-based temporal logic, such as understanding sequential order, calendar mechanics, e.g., knowing February has fewer days than March, or simple arithmetic operations on dates. This level tests understanding of the structural logic of time, which tends to generalize well after the core ideas are grasped, with minimal dependence on real-world knowledge. For example, determining the year after 2010 to be 2011, as shown in Figure 2.1, relies only on understanding the numerical progression of years.

Moving to more complex forms, the **Level 2 (L2)** or **Time-Event Relation** shifts from abstract temporal logic to knowledge-intensive reasoning. This level involves connecting specific events or facts to particular points or intervals in time. The example provided by Tan et al. (2023a) and illustrated in Figure 2.1, "What team did Leo Messi play for in 2010?", exemplifies this, where the correct answer ("Barcelona") requires retrieving a piece of factual knowledge based on the specified year. This is different from L1 as it demands not just an understanding of time, but the recall and application of time-dependent information from a knowledge base,whether that be in human memory or an LLM's parameters. Humans often need to memorize such associations or use contextual information to deduce these relations.

The most advanced layer, **Level 3 (L3)** or **Event-Event Relation**, involves reasoning about the temporal relationships between different events. This often requires coordinating memory retrieval, logical deduction, and understanding of how events unfold, connect causally, and progress narratively. As illustrated by Tan et al. (2023a) with the question, "What team did Leo Messi play for after Barcelona?", answering correctly ("Paris Saint-Germain") involves not just knowing individual time-event facts but understanding the sequence of Messi's career moves. This might need multi-step reasoning: knowing when he played for Barcelona, when it ended, and what event (joining another team) followed. This level tests the ability to construct temporal narrative or timeline from discrete pieces of information.

Understanding these distinct levels helps identify specific challenges for LLMs. This thesis will concentrate on challenges related to Level 1 (Time-Time Relation), investigating how an LLM's ability to execute basic temporal logic and date arithmetic is influenced by the format in which temporal information is presented. Proficiency in these fundamental calculations is essential

before more complex time-event (L2) or event-event (L3) reasoning can be reliably achieved.

Types of Temporal Information

Temporal information in natural language is expressed through a variety of forms, each presenting unique challenges for computational understanding. The widely adopted TimeML annotation standard (Pustejovsky et al., 2005) provides a comprehensive taxonomy for these expressions. The most fundamental types include:

- **Absolute/Explicit Temporal Expressions:** These expressions specify an exact point or interval on a timeline, often connected to a calendar or clock system. Examples include full dates, e.g., "2020-01-15", partial dates, e.g., "March 2021", and specific times, e.g., "14:30 UTC".
- **Relative Temporal Expressions:** Unlike absolute expressions, these represent time in relation to a reference point, which is often the speaking time or another mentioned event. Common examples include "yesterday," "next week," "three years ago," "the following day," or "two months prior to the event." Interpreting these requires correctly identifying and resolving the temporal anchor.
- **Durations:** These expressions specify the length of a time interval, indicating how long an event or state lasts, e.g., "for three hours," "a two-week period," or "all day". Understanding durations is essential for reasoning about event extent and persistence.
- **Frequencies and Set-Based Temporal Expressions:** These describe how often an event occurs or refer to sets of times. Examples include "daily," "every Tuesday," "twice a month," or "on weekends."

Achieving full temporal competence requires LLMs to engage with all these categories, but this thesis narrows its focus to the processing of explicit calendar dates. This targeted approach allows for a controlled examination of how variations in the input representation of these common temporal markers influence model performance in tasks requiring time-dependent factual recall.

2.2. Related Work

This section reviews the existing literature to provide context for the present research. It focuses on studies evaluating temporal reasoning in Large Language Models, research into LLM sensitivity to input formats and prompting, and their processing of numerical and structured data. By examining these areas, this review will highlight the specific contributions of this thesis, particularly in addressing the under-explored impact of date format variations on LLMs' temporal reasoning accuracy.

2.2.1. Evaluating Temporal Reasoning in LLMs

Evaluating the temporal reasoning capabilities of LLMs is a growing research area, predominantly utilizing specialized benchmarks. For instance, TimeQA (Chen et al., 2021) challenges models by requiring them to process complex news texts where temporal details are scattered, demanding synthesis for understanding event chronology and duration. SituatedQA (Zhang and Choi, 2021) and its temporal subset test the model's ability to use provided temporal context to retrieve time-sensitive facts. Benchmarks like TimelineQA (Tan et al., 2023b) assess an LLM's knowledge of events associated with specific years, while diagnostic datasets such as TempLAMA (Dhingra et al., 2022) probe the understanding of factual knowledge that changes over time, particularly facts with defined start/end dates.

More recently, TEMPReason (Tan et al., 2023a) was designed to specifically evaluate multi-step, compositional temporal reasoning, including diverse question types covering date arithmetic, event ordering, and duration understanding. Pushing these boundaries further, ComplexTempQA (Gruber et al., 2024) focuses on even more intricate temporal question answering scenarios, often requiring models to perform multiple, complex reasoning steps over indirectly linked events or nuanced temporal relations.

Other efforts, such as MTEB (Muennighoff et al., 2023), while not solely focused on temporal reasoning, also include tasks with temporal components, highlighting the pervasive nature of temporal understanding in NLP. Collectively, these benchmarks assess a wide array of temporal reasoning skills, including temporal question answering, event ordering, duration understanding, fact verification over time, and date arithmetic.

Studies utilizing these benchmarks have revealed both strengths and significant weaknesses in current LLMs. While models often show some capacity for retrieving explicitly stated temporal information and performing simple ordering tasks (Kassner et al., 2021; Dhingra et al., 2022), they frequently struggle with more complex scenarios like sequential reasoning, implicit temporal relations, and accurately tracking factual changes over time, especially beyond their knowledge cutoff (Chen et al., 2021; Tan et al., 2023a). Furthermore, LLMs can be prone to "temporal misattribution" or exhibit biases towards more recent information (Jang et al., 2023).

Crucially, while these benchmarks and the resulting research provide valuable insights into the overall temporal reasoning performance of LLMs, they typically do not systematically investigate the influence of lower-level input characteristics, such as the syntactic format of explicit dates as a primary variable affecting accuracy. The focus has largely been on the semantic complexity of the temporal reasoning task itself or the model's architectural capabilities. As a result, there is a gap in understanding whether and how these fundamental variations in representing temporal anchors directly impact an LLM's ability to process queries and retrieve or compute time-dependent information. This thesis aims to address this specific gap by isolating date format as an independent variable in temporal reasoning tasks.

2.2.2. LLM Sensitivity to Input Representation and Prompting

Beyond the semantic complexity of temporal reasoning tasks themselves, the performance of Large Language Models is known to be highly sensitive to the precise formulation of their input. This phenomenon underscores the importance of input representation, a broad area that includes the more specific practice of prompt engineering, which has emerged as a critical field in effectively generating the expected behavioral outcomes and knowledge from LLMs (Liu et al., 2023).

LLMs, particularly instruction-tuned models, are designed to follow instructions and complete tasks based on the textual prompts they receive. However, extensive research has demonstrated that even subtle variations in prompt wording, structure, or formatting can lead to significant differences in output quality, factual accuracy, and task success rates (Brown et al., 2020; Zhao et al., 2021; Shin et al., 2020). For example, studies have shown that rephrasing a question, adding few-shot examples, or specifying the desired output format can drastically improve performance on tasks ranging from question answering to code generation and logical reasoning (Wei et al., 2023; Kojima et al., 2023; Lu et al., 2022). This sensitivity suggests that LLMs do not possess a perfectly robust or abstract understanding of tasks but are instead significantly influenced by the surface patterns and cues present in the input. Webson and Pavlick (2022) further highlighted that prompt-based fine-tuning can lead models to rely on superficial cues in the prompt rather than learning the underlying task. The PromptRobust benchmark (Zhu et al., 2024), for instance, evaluates LLM resilience against prompts perturbed by character, word, sentence, and semantic level changes, demonstrating that even seemingly minor deviations mimicking user errors can affect outcomes, reinforcing the impact of surface-form variations.

The challenge of finding optimal prompts has led to various automated and semi-automated

prompt engineering techniques (Shin et al., 2020; Zhou et al., 2023). However, the underlying principle remains: the way information is presented to an LLM matters profoundly. While much of prompt engineering focuses on crafting elaborate instructional text or selecting effective few-shot exemplars, the core observation of input sensitivity is highly relevant to more constrained forms of input variation. Indeed, the syntactic format of a date represents such a specific instance. Although not 'prompt engineering' in the traditional sense of crafting natural language instructions, different date formats present identical temporal information via distinct token sequences and structural patterns. Given LLMs' established sensitivity to subtle input changes, it is plausible that these differences in date representation could influence how the model tokenizes, processes, and ultimately reasons with the temporal information conveyed. For example, a more structured, numeric format like "2020-01" might be processed differently by the model's numerical or symbolic reasoning capabilities compared to a more natural language-like format such as "January 2020." This section, therefore, provides a theoretical underpinning for the hypothesis that date formats, as a fundamental aspect of input representation for temporal queries, can significantly impact LLM performance on time-dependent tasks.

2.2.3. LLM Processing of Numerical and Structured Data

The challenge of temporal reasoning, particularly when involving explicit dates, is significantly connected with how LLMs process numerical and structured information. Dates, at their core, are numerical constructs, e.g., day, month, year, presented in a structured format. Therefore, understanding LLM capabilities and limitations in these fundamental areas is crucial for contextualizing their performance on temporal tasks.

Research into the numerical reasoning abilities of LLMs has revealed a mixed picture. While pretrained models can perform some basic arithmetic operations, particularly those frequently encountered in their training corpora, their accuracy often degrades with increasing complexity, number of digits, or the need for multi-step calculations (Wallace et al., 2019). Considerable research effort is dedicated to improving LLM performance on complex reasoning tasks that inherently involve such multi-step numerical or logical operations, for instance, through techniques like chain-of-thought finetuning and adaptive reasoning mechanisms (Chen et al., 2024). Studies have shown that LLMs may struggle with robust "number sense," failing to grasp the magnitude or relational properties of numbers in a way that humans do, sometimes treating numbers more like opaque tokens rather than symbols with inherent mathematical properties (Spithourakis and Riedel, 2018; Nogueira et al., 2021). This sub-optimal numerical processing could directly impact tasks like date arithmetic or duration calculation, where precise manipulation of numerical date components is essential.

Furthermore, dates represent a specific type of structured data embedded within natural language. The ability of LLMs to interpret and utilize structured information presented textually is an ongoing area of investigation. While LLMs can sometimes extract and reason over tabular data presented in serialized formats (Herzig et al., 2020), their handling of less explicitly demarcated structures within free-form text, like addresses, version numbers, or indeed dates, can be less consistent. The internal representation and processing of such structures are not yet fully understood. Different date formats, e.g., "2020-01-15" vs. "January 15, 2020", inherently vary in their textual structure and the explicitness of their numerical components. A format like "2020-01-15" is highly regular and tokenizes into numerical segments, whereas "January 15, 2020" mixes words and numbers.

Given these considerations, if LLMs exhibit general difficulties with precise numerical operations or with consistently parsing and utilizing structured information from text, it is plausible that the specific format of a date could influence their performance. A date format that aligns more closely with how LLMs effectively process numbers, e.g., as distinct numerical tokens, or parse structure might lead to more accurate temporal reasoning compared to formats that are more ambiguous or require more complex string-to-number-to-structure mappings by the model. This

section highlights that numerical and structural processing represents another dimension through which input date format could be a significant factor in LLM temporal reasoning accuracy.

Building on these identified gaps, this thesis systematically investigates how controlled variations in date formats, an aspect largely overlooked in prior evaluations, affect the accuracy of Large Language Models in temporal reasoning tasks.

3. Methodology

This chapter presents the experimental methodology used to evaluate how variations in date formats affect LLM performance on Level 1 temporal reasoning tasks. The approach involved generating diverse syntactic representations of dates within standardized queries and evaluating a target LLM's accuracy. The following sections will present: the base dataset and the procedure for creating format variations; the selected language model and its experimental configuration, including prompt design; and the metrics used for evaluating model outputs. This systematic approach allows for a controlled investigation into the research question: How do different representations of explicit calendar dates and temporal offsets, e.g., "January 2020" vs. "2020-01", or "3 years" vs. "thirty-six months" affect an LLM's accuracy in calculating target dates?

3.1. Dataset and Task

This section presents the foundational dataset selected for the experiments and explains the controlled generation of diverse date format inputs. It also defines the specific task posed to the language model.

3.1.1. Base Dataset: TempReason L1

The primary dataset for this research was selected based on four key criteria: a direct focus on foundational arithmetic, the presence of explicit dates and offsets suitable for manipulation, a controlled level of task complexity and its status as an established benchmark. The Level 1 (L1) subset of the TempReason benchmark (Tan et al., 2023a) uniquely satisfies these requirements.

TempReason is a benchmark specifically designed to evaluate temporal reasoning in language models. Its L1 subset, which focuses on "Time-Time Relations," provides the ideal testing ground for this study. The tasks at this level primarily involve date arithmetic, directly addressing the need for a focus on foundational arithmetic. This allows for a clear analysis of a model's core computational abilities when processing temporal information. An example of an original L1 question is: "What is the time 3 year and 3 month after Jul, 1699?", with the expected answer being a specific date, e.g., "Oct, 1702".

Furthermore, the questions in this subset inherently contain explicit dates and temporal offsets, making them perfectly suited for the systematic format manipulation that is central to this thesis. By concentrating on these L1 tasks, the experiment benefits from a controlled complexity, reducing interference from complex knowledge retrieval (L2) or multi-event reasoning (L3). This controlled environment enables a more direct assessment of how syntactic date format variations impact the arithmetic task itself. Finally, by using an established benchmark, this study leverages a standardized set of questions and answers, which facilitates comparability with other research and grounds the findings in existing evaluation practices.

The L1 test data, specifically the file `test_l1.jsonl`, was sourced directly from the official TempReason repository hosted on Hugging Face Datasets. This file contains 4,000 unique questions, each forming the basis for the generated variants used in this study.

3.1.2. Generation of Date Format Variations

To investigate the impact of date representation, the original L1 questions were systematically transformed into multiple variants, each differing in the syntactic format of its temporal compo-

nents. This process involved parsing the original question and then reformatting its offset and base date components.

Question Parsing

Original L1 questions were first parsed using a regular expression, the specific pattern is provided in Appendix A.1. It is designed to identify and extract key temporal elements:

- The numerical values and units of the temporal offset, e.g., "3" and "year", "3" and "month".
- The direction of the offset, e.g., "after" or "before".
- The month and year of the base date, e.g., "Jul" and "1699".

This structured information then served as the input for the reformatting functions.

Offset Formatters

The extracted temporal offset was reformatted using three distinct approaches, hereafter referred to by their code identifiers:

- **Original (`original`):** The offset string was kept identical to its appearance in the original TempReason L1 question, e.g., "3 year and 3 month".
- **Total Months Numeric (`total_months_numeric`):** The offset was converted into a total number of months, expressed numerically, e.g., "39 months". This involved calculating $(\text{years} \times 12) + \text{months}$.
- **Total Months Word (`total_months_word`):** The offset was converted into a total number of months, with the number spelled out in words, e.g., "thirty-nine months". The `inflect` Python library was utilized for converting numbers to words.

Base Date Formatters

The extracted base date was reformatted using four distinct approaches, hereafter referred to by their code identifiers:

- **Original (`original`):** The base date string was kept identical to its appearance in the original question, e.g., "Jul, 1699".
- **ISO Format (`iso`):** The base date was converted to the YYYY-MM ISO 8601 format, e.g., "1699-07".
- **Ordinal Month (`ordinal_month`):** The base date was expressed with the month as an ordinal number followed by the year as numerals, e.g., "7th month of 1699". The `inflect` library was used for generating ordinal numbers.
- **Full Words (`full_words`):** The base date was expressed with the month as an ordinal number and the year spelled out in words, e.g., "seventh month of one thousand six hundred and ninety-nine". The `inflect` library was used for both ordinal numbers and number-to-word conversion for the year.

Combinatorial Generation of Question Variants

By combining each of the three offset formats with each of the four base date formats, a total of 12 unique question variants could theoretically be generated for each original L1 question (3 offset types × 4 base date types). Each such generated question variant retains the original question's underlying arithmetic problem and gold-standard answer but presents the temporal information to the LLM in a different syntactic structure.

While twelve unique format variations were generated, for the primary experimental evaluation, a focused subset of four distinct format combinations was strategically selected. This selection was not arbitrary, it was designed to cover a spectrum of linguistic properties, from highly structured and machine-readable formats to highly naturalistic and verbose ones. The goal was to create distinct experimental conditions that could effectively probe the model's sensitivity to both structure and linguistic complexity. The four chosen variations, which form the core independent variables in this analysis, are defined as follows:

- **Original Offset, Original Date** (original_offset_original_date.jsonl): This serves as a baseline, using the unmodified format from the TempReason L1 dataset, e.g., "3 year and 3 month after Jul, 1699". It represents a standard, semi-naturalistic format and establishes the model's performance before any controlled disruptions are introduced.
- **Original Offset, ISO Date** (original_offset_iso_date.jsonl): Combines the original offset with a structured ISO base date, e.g., "3 year and 3 month after 1699-07". It was chosen to directly test the impact of a highly structured and unambiguous date format. Standardizing the month to YYYY-MM isolates structural regularity effects, enabling direct comparison with the baseline to assess parsing performance.
- **Total Months Numeric Offset, Ordinal Month Date** (total_months_numeric_offset _ordinal_month_date.jsonl): Features a numeric total month offset and a base date with an ordinal month, e.g., "39 months after 7th month of 1699". This condition was selected to challenge the model's numerical processing in two ways. First, it requires the model to handle an aggregated numerical offset rather than a pre-parsed year/month structure. Second, it uses a different natural language format for the base date, testing adaptability to varied linguistic cues for the same temporal concept.
- **Total Months Word Offset, Full Words Date** (total_months_word_offset_full_words_date.jsonl): Representing the opposite end of the spectrum from the ISO format, this highly verbose condition was chosen to assess the model's ability to reason when all numerical information is fully lexicalized. It tests the model's "number sense" when numbers are presented as words rather than digits, pushing the boundaries of its natural language understanding in a numerical context.

Task Definition

For all experiments, the task presented to the language model was to answer the provided question in one of the four chosen formats by calculating the target date. Crucially, the model was explicitly instructed via the prompt (detailed in Section 3.2.2) to provide its final answer strictly in the YYYY-MM ISO format, regardless of the input question's format. This standardized output format simplifies automatic evaluation and ensures that performance differences are attributable to the model's processing of the input format, not its ability to generate varied output date strings.

3.2. Model Selection and Experimental Setup

This section outlines the specific Large Language Model chosen for the experiments, the design of the prompt used to obtain responses, and the relevant implementation details regarding software, hardware, and the prediction generation process.

3.2.1. Language Model

The primary model investigated in this thesis is Llama 3.2 3B Instruct, an advanced instruction-tuned decoder-only Transformer model developed by Meta AI (Touvron et al., 2023; Llama Team,

2024). It was chosen based on its proven ability to perform well in reasoning and instruction-following tasks, making it well-suited for analyzing complex aspects of temporal understanding. As an instruction-tuned model, Llama 3.2 3B Instruct is specifically optimized to understand and respond to user queries and prompts, a crucial attribute for the experimental setup of this study. The choice of Llama 3.2 3B Instruct was motivated by several factors:

- Instruction-Following Capabilities: As an instruction-tuned model, it is designed to understand and follow specific user directives and output formats, which is crucial for the controlled experimental setup of this study, particularly the requirement for a YYYY-MM date output.
- Reasoning Abilities: The Llama family of models has demonstrated competent reasoning skills, which are relevant for the date arithmetic tasks included in the TempReason L1 dataset.
- Accessibility and Manageable Size: The 3-billion-parameter version offers a balance between strong performance and computational feasibility for academic research, being accessible via platforms like Hugging Face and manageable for experimentation on available GPU resources, e.g., Google Colab.
- Contemporary Relevance: Utilizing a recent and widely recognized open-access model ensures the findings are relevant to current discussions in LLM research.

The model and its corresponding tokenizer (AutoTokenizer) were accessed and utilized through the Hugging Face `transformers` library.

3.2.2. Prompting Strategy

A standardized prompting method helps to confirm that observed performance differences are due to date format, not prompt inconsistency. The following prompt structure, implemented in the `generate_batch_response_llama` function, was used for querying the Llama 3.2 model:

```
System: "You are a Helpful assistant"
User: "Answer the following question:
{prompt}
Explain how you arrive at the result briefly. Then, on the next
   line, output **only** the final date in YYYY-MM format, with
   no extra words"
```

Where `{prompt}` is the placeholder for each generated question variant. This prompt design is motivated by two key considerations.

Firstly, achieving a standardized output format was deemed critical. The explicit instruction, "Then, on the next line, output only the final date in YYYY-MM format, with no extra words," aimed to standardize the model's output. This approach was intended to make the extraction and evaluation of the final date answer more reliable and to minimize parsing errors. As a result, the evaluation metrics could more accurately reflect the model's temporal reasoning capabilities, rather than its ability to follow varied output instructions.

Secondly, the prompt is crafted to guide the model through a reasoning process. To this end, the request for a brief explanation before the final answer was included. While the primary focus of evaluation remains the final date, it is assumed that this instruction will lead the model to perform more structured reasoning, comparable to chain-of-thought strategies. Such a process could potentially improve the accuracy of the final temporal calculation, although the explanation itself was not directly evaluated in this study.

3.2.3. Experimental Setup and Implementation

The experiments were conducted using Python 3.11 and a set of open-source libraries to support a reliable and reproducible workflow. The following subsections detail the software, hardware, model configuration, and data handling procedures employed in this study.

Software and Model Configuration

The experimental framework was built upon the Hugging Face ecosystem, leveraging Python 3.11 and PyTorch as the underlying deep learning framework. The `datasets` library was used for loading and managing the TempReason benchmark and its generated variants.

The text-generation `pipeline` from the `transformers` library was used to instantiate the Llama 3 model. To optimize performance on the available CUDA-enabled GPU hardware, several specific configurations were applied. The model was loaded using a `bfloat16` floating-point data type to balance memory usage with computational speed. Automatic model parallelization across available GPU devices was enabled via the `device_map="auto"` parameter. Finally, as a standard procedure for causal language models, the tokenizer's padding token ID was explicitly set to its end-of-sequence (EOS) token ID to ensure consistent handling of batched inputs during inference.

Generation Parameters and Reproducibility

To control the model's output and ensure the reproducibility of the experiments, a fixed set of generation parameters was used for all inference tasks.

- **Maximum New Tokens:** The output length was constrained by setting `max_new_tokens` to 128. This value was chosen to provide sufficient length for the model's brief explanation and the required 'YYYY-MM' date output, while avoiding excessively verbose or truncated responses.
- **Decoding Strategy:** The default decoding strategy of the Hugging Face `pipeline` was employed for text generation. By default, for this model, the pipeline uses a deterministic greedy decoding approach. Consequently, sampling-related parameters were ignored. The generation parameters, such as `temperature`, `top_p`, and `repetition_penalty`, were left at their default values of 1.0.
- **Batch Size:** Questions were processed in batches of 8 (`batch_size=8`). This size offered a practical balance between maximizing GPU utilization for faster processing and operating within typical GPU memory constraints for a model of this scale.

Crucially, no global random seed was set for the generation process, which has direct implications for the reproducibility of the results, as discussed in the Limitations section (5.2).

Response Extraction and Data Handling

Following the prompt's instructions, the final date answer was extracted from the model's generated output for each item. This was achieved by parsing the content of the last message in the conversation, splitting it by newline characters, and taking the final line as the predicted date. This extraction method directly corresponds to the instruction given to the model to place the YYYY-MM formatted date on the last line of its response.

Experimental artifacts were stored systematically to ensure traceability and facilitate analysis. The datasets representing each date format combination were saved as JSON Lines (`.jsonl`) files. Model predictions for each dataset variant were also stored in a `.jsonl` format, augmenting the original data with the model's generated response. Finally, the computed evaluation metrics

for each condition were aggregated and saved in structured JSON (`.json`) files for subsequent reporting.

3.3. Evaluation

To assess the language model's performance on the temporal reasoning tasks with varying date formats, a systematic evaluation procedure was implemented. This involved normalizing the model's predictions and the gold standard answers, followed by the application of specific quantitative metrics.

3.3.1. Evaluation Set Policy

The evaluation was conducted using the official test set of the TempReason L1 benchmark, specifically the file `test_l1.jsonl` containing 4,000 examples. This file was loaded as the sole data split for the experiment. While the code refers to this split as "validation" for loading purposes with the `datasets` library, it is in fact the official test set and was used in its entirety for the final evaluation without any re-splitting or model tuning. The primary goal of this thesis is to analyze the performance of a pre-trained model under different input conditions, rather than to perform hyperparameter tuning or model selection, thus a separate validation set was not required.

3.3.2. Answer Normalization

A critical preparatory step in the evaluation process was the normalization of both the gold standard answers from the TempReason dataset and the predictions generated by the Llama 3.2 model. This normalization was performed by a custom function, `_normalize_text`, the implementation details of which are provided in Appendix A.2. The primary purpose of this normalization step was to ensure a fair and consistent comparison between the predicted and target answers. LLMs can produce outputs with minor syntactic variations that do not necessarily reflect a substantive error in temporal reasoning. Normalization is considered for such variations, including differences in capitalization, e.g., "Mar" vs. "mar", punctuation, e.g., "Mar, 1789" vs. "Mar 1789", the presence of leading or trailing whitespace, or irrelevant characters like asterisks that might be artifacts of the generation process. By standardizing these aspects, the evaluation can focus more directly on the semantic correctness of the temporal calculation rather than superficial formatting differences. For instance, without normalization, a prediction like "mar, 1789" might be incorrectly penalized when compared against a gold answer of "1789-03", even if the underlying temporal understanding is correct.

The `_normalize_text` function operates by converting various common date string representations into a canonical 'YYYY-MM' ISO format. This includes parsing dates like "March, 1789", "mar 1789", or even "1789-03-15" into "1789-03". If a direct conversion to the 'YYYY-MM' format is not successfully achieved, e.g., if the input string does not conform to a recognized date pattern, the function defaults to a more general cleaning process. This fallback involves converting the text to lowercase, stripping leading and trailing whitespace, and removing certain common non-alphanumeric characters. This ensures that even if a specific date pattern isn't matched, a basic level of text standardization is still applied for comparison.

3.3.3. Evaluation Metrics

Following the normalization of answers, the model's performance was estimated using a set of specific evaluation metrics. Each metric was chosen to provide a distinct perspective on the model's temporal reasoning capabilities and its sensitivity to date format variations.

- **Exact Match (EM):**

- *Definition:* Exact Match is defined as the percentage of predictions where the normalized predicted date in 'YYYY-MM' format precisely matches the normalized gold standard date.
- *Relevance:* EM serves as a strict measure of correctness. A high EM score indicates that the model is not only reasoning correctly about the temporal offset but is also able to produce the answer in the exact target format. This metric directly addresses the accuracy component of the primary research question concerning the influence of date formats on model performance.

- **Mean Absolute Error (MAE) - Year:**
  - *Definition:* The Mean Absolute Error in years is calculated as the average absolute difference between the year component of the normalized predicted date and the year component of the normalized gold standard date. This metric is computed only for instances where valid year components can be successfully extracted from both the prediction and the gold answer.
    The formula is: $MAE_{Year} = \frac{1}{N_{valid}} \sum_{i=1}^{N_{valid}} |Year_{pred,i} - Year_{gold,i}|$, where N_{valid} is the count of examples with valid extractable years in both prediction and gold answer.
  - *Relevance:* MAE-Year provides insight into the magnitude of error when the model's prediction is not an exact match. A lower MAE indicates that even when the model is incorrect, its predictions are chronologically closer to the correct year. This metric is less strict than EM and helps to understand if certain date formats lead to larger or smaller deviations in the predicted timeline, offering a more nuanced view of performance beyond binary correctness.

- **Trend Accuracy:**
  - *Definition:* Trend Accuracy measures the percentage of predictions where the temporal relationship of the predicted date relative to a reference year in the question aligns with the temporal relationship of the gold standard date relative to the same reference year. The reference year is typically the year mentioned in the base date of the input question, for example, for a question "...after July, 1699", the reference year is 1699.
    Let $N_{\text{valid\_trend}}$ be the number of examples that meet the validity conditions described below, and $N_{\text{correct\_trend}}$ be the count of those examples where the predicted trend matches the gold trend. The accuracy is then calculated as:

$$\text{Trend Accuracy} = \frac{N_{\text{correct\_trend}}}{N_{\text{valid\_trend}}} \times 100\% \tag{3.1}$$

    This metric is calculated only for examples where: (a) valid year components can be extracted from the prediction, the gold answer, and the reference point in the question, and (b) the gold answer's year is different from the reference year, to ensure a clear "before" or "after" trend.
  - *Relevance:* Trend Accuracy assesses whether the model correctly understands the fundamental temporal direction implied by terms like "before" or "after," even if it fails to calculate the precise target date. This is particularly relevant for understanding if specific date formats might confuse the model about the directionality of the temporal reasoning task, which is a core aspect of temporal understanding.

4. Analysis

This chapter presents the experimental results from evaluating the Llama 3.2 3B Instruct model on the TempReason L1 benchmark under four distinct date format conditions. The primary objective is to analyze how these syntactic variations in presenting temporal information influence the model's performance. The analysis focuses on three key metrics: EM accuracy, MAE in years, and Trend Accuracy. The four conditions evaluated are: 1) Original Offset, ISO Date; 2) Original Offset, Original Date; 3) Total Months Numeric Offset, Ordinal Month Date; and 4) Total Months Word Offset, Full Words Date. All evaluations were conducted on a consistent set of 4000 questions derived from the TempReason L1 test set.

4.1. Performance Evaluation

The performance of the Llama 3 model was evaluated across the four date format conditions, with each condition containing 4000 test items. The Exact Match (EM) metric was calculated on all 4000 items. The Mean Absolute Error (MAE) and Trend Accuracy metrics, however, rely on the successful extraction of a year from the model's output. For these two metrics, the effective sample size varied per condition, with an average of [ZZZ] items per condition being excluded per condition due to unparsable year information in the generated response.

A detailed summary of these performance metrics is presented in Table 4.1. Subsequent sections will delve into the analysis of each metric individually.

Table 4.1.: Overall Performance Metrics Across Date Format Conditions (N=4000)

Date Format Condition	Exact Match (%)	MAE (Year)	Trend Accuracy (%)
Original Offset, ISO Date	33.55	21.19	90.80
Original Offset, Original Date	31.55	23.87	90.89
Total Months Numeric Offset, Ordinal Month Date	32.87	23.85	91.53
Total Months Word Offset, Full Words Date	32.97	18.68	90.94

4.1.1. Exact Match (EM)

Exact Match accuracy measures the percentage of instances where the model's normalized YYYY-MM output precisely matched the gold standard. Figure 4.1 illustrates the EM scores for each condition.

As presented in Table 4.1 and Figure 4.1, the "Original Offset, ISO Date" condition achieved the highest EM score at 33.55%. The "Original Offset, Original Date" condition showed the lowest EM score at 31.55%. The other two conditions, "Total Months Numeric Offset, Ordinal Month Date" (32.87%) and "Total Months Word Offset, Full Words Date" (32.97%), performed very similarly to each other, falling between the highest and lowest scores. The overall range of EM scores is relatively narrow, from 31.55% to 33.55%, indicating that while date format does have an impact, the model's performance on this task remains modest across all tested formats.

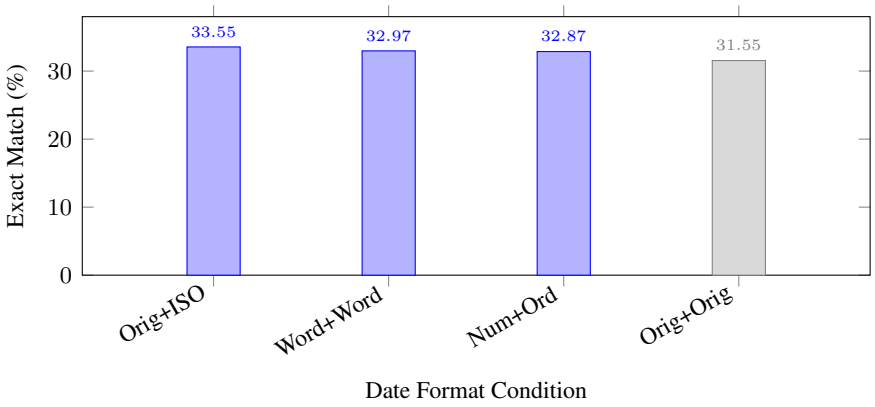

Figure 4.1.: EM Scores by Date Format Condition (Sorted Best to Worst). ‘Orig+Orig‘ is the baseline.

4.1.2. Mean Absolute Error (MAE) - Year

The Mean Absolute Error (MAE) in years is defined as the average deviation of the predicted year from the gold standard year when an exact match is not achieved. A lower MAE indicates that the model's year predictions are, on average, closer to the correct year. Referring to Figure 4.2, a notable finding appears for the MAE-Year metric. The "Total Months Word Offset, Full Words Date" condition, despite not having the highest EM, achieved the lowest MAE-Year of 18.68. This suggests that when this highly verbose format led to an incorrect YYYY-MM answer, the year component of the prediction was, on average, less distant from the true year compared to other formats. On the opposite, the "Original Offset, Original Date" (MAE 23.87) and "Total Months Numeric Offset, Ordinal Month Date" (MAE 23.85) conditions exhibited the highest MAE-Year scores, indicating larger average deviations in the predicted year. The "Original Offset, ISO Date" condition had an MAE-Year of 21.19. The notably better MAE-Year performance of the "Full Words" format is promising and should be examined further in the deeper error analysis and discussion to uncover the types of errors responsible for this result.

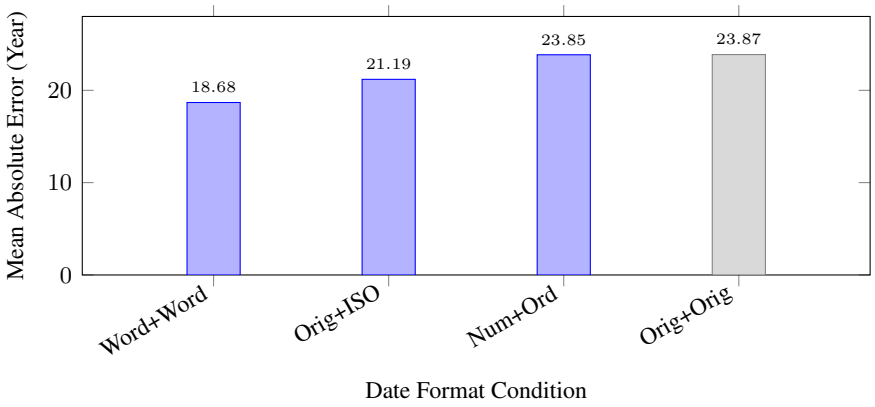

Figure 4.2.: MAE - Year by Date Format Condition (Sorted Best to Worst). ’Orig+Orig’ is the baseline.

4.1.3. Trend Accuracy

Trend Accuracy assesses whether the model correctly determined if the target date was before or after the reference date, irrespective of calculating the exact YYYY-MM. This metric provides insight into the model's high-level directional understanding.

As shown in Table 4.1, the model's Trend Accuracy was consistently high across all four conditions, ranging from 90.80% to 91.53%. The "Total Months Numeric Offset, Ordinal Month Date" condition achieved the highest score at 91.53%, while the "Original Offset, ISO Date" condition had the lowest at 90.80%. These strong results, calculated according to the method defined in Section 3.3.3 (Equation 3.1), suggest that the Llama 3 model effectively comprehends the directional aspect (before/after) of the temporal reasoning task, regardless of the specific input date format. The small variation between conditions indicates that the choice of date format had a minimal impact on this fundamental component of temporal understanding for the L1 tasks.

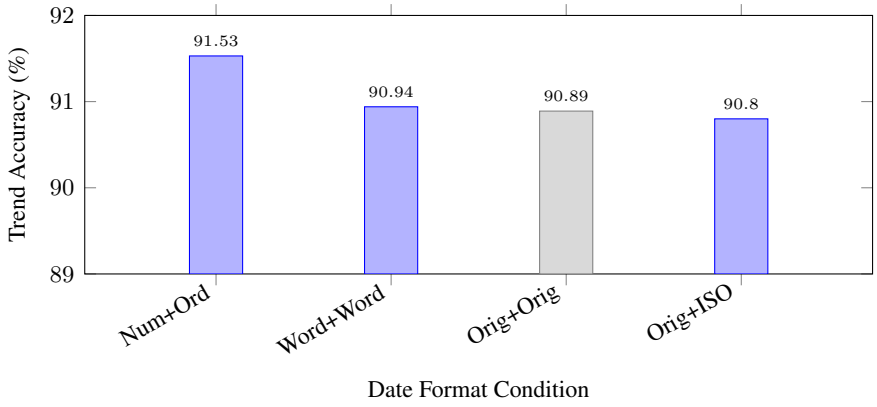

Figure 4.3.: Trend Accuracy by Date Format Condition (Sorted Best to Worst). 'Orig+Orig' is the baseline

4.2. Qualitative Error Analysis

To better understand the quantitative results, this section examines specific examples of errors the model made for each format condition. These qualitative examples reveal distinct failure patterns that help explain the differences observed in the EM and MAE-Year metrics.

Errors in Structured vs. Naturalistic Formats

The most common errors involved incorrect month calculations or misinterpretations of the year, with the input format often influencing the type of mistake.

Format: Original Offset, ISO Date (High EM, Higher MAE): When the model failed on the structured ISO input, it often made significant arithmetic errors, suggesting a failure in the calculation step rather than the parsing step.

- **Question:** "What is the time 8 year and 7 month after 1821-06?"
- **Gold Answer:** ‘1830-01‘
- **Model Prediction:** ‘1829-06‘
- **Error Type:** Incorrect Year Calculation. The model appears to have correctly parsed ‘1821-06‘ but failed the arithmetic, possibly by only adding the 8 years and ignoring the 7 months, or by making a simple off-by-one error in the year.

Format: Original Offset, Original Date (Low EM, High MAE): With the semi-naturalistic input, errors sometimes stemmed from parsing issues, compounding the arithmetic challenge.

- **Question:** "What is the time 11 year and 2 month before Apr, 1954?"

- **Gold Answer:** ‘1943-02‘
- **Model Prediction:** ‘1943-04‘
- **Error Type:** Incorrect Month Calculation. The model correctly calculated the year (‘1954 - 11 = 1943‘) but seems to have ignored the month offset, simply retaining the original month "Apr". This suggests a failure to fully integrate all parts of the parsed natural language offset.

Errors in Verbose Word-Based Formats

The fully word-based format led to unique error patterns, highlighting the model’s different handling of lexicalized numbers.

Format: Total Months Word Offset, Full Words Date (Low MAE): This format’s errors were often small in magnitude, typically miscalculating the month while keeping the year correct or very close. This pattern is consistent with the low MAE-Year score for this condition.

- **Question:** "What is the time one hundred and thirty-four months after third month of one thousand nine hundred and ninety?"
- **Gold Answer:** ‘2001-05‘ (11 years, 2 months after 1990-03)
- **Model Prediction:** ‘2001-03‘
- **Error Type:** Small-Magnitude Month Error. The model correctly grasped the large year offset but failed the final, precise month calculation. It appears to have successfully added 11 years to 1990 but then defaulted to the original month. This supports the hypothesis that the model handles the semantic magnitude of large numbers well but struggles with the symbolic arithmetic of complex offsets.

Directional Errors

While rare, as indicated by the >90% Trend Accuracy, errors in temporal direction did occur across all formats. These represent a more fundamental failure of comprehension.

- **Question:** "What is the time 5 year and 1 month before Aug, 1642?"
- **Gold Answer:** ‘1637-07‘
- **Model Prediction:** ‘1647-09‘
- **Error Type:** Incorrect Direction. The model completely misunderstood the keyword "before" and instead calculated the date 5 years and 1 month *after* the reference date. This type of error, though infrequent, is critical as it shows a breakdown in basic semantic understanding, independent of the numerical format.

4.3. Summary of Analytical Findings

The quantitative analysis presented in this chapter highlights several distinct patterns regarding the Llama 3.2 3B Instruct model’s sensitivity to date format variations.

Firstly, the model’s precision, as measured by Exact Match accuracy, showed modest but discernible differences across formats, with the structured "Original Offset, ISO Date" condition demonstrating a slight advantage.

Secondly, an unexpected finding emerged for Mean Absolute Error in years, where the highly verbose "Total Months Word Offset, Full Words Date" format led to the smallest average year deviations when errors occurred.

Finally, the model's understanding of temporal directionality, reflected in Trend Accuracy, remained consistently high and robust across all tested formats.

These observations underscore a complex interplay between input syntax and model performance, which will be explored and interpreted in detail in the subsequent Discussion chapter.

5. Discussion

The experimental investigation presented in this thesis revealed a nuanced relationship between input representation and model accuracy. While the Llama 3 model displayed a consistent grasp of temporal directionality, its capacity for precise date arithmetic was sensitive to the format of the input. Structured base dates appeared beneficial for exactness, yet a fully word-based format led to smaller errors in year magnitude when exact matches were missed. These key findings form the basis for the subsequent interpretation and discussion regarding the implications for LLM temporal reasoning.

5.1. Interpretation of Findings and Answering the Research Question

The experimental results presented in Chapter 4 provide a clear, affirmative answer to the central research question: the syntactic format of dates significantly affects the Llama 3 model's accuracy on temporal reasoning tasks. The data reveals a complex relationship between input structure, calculative precision, and error characteristics, indicating that the model has not yet developed a fully abstract, format-independent representation of time. Instead, its performance appears to be a composite of distinct and sometimes competing processing mechanisms.

The most direct impact of format was on precision, where a clear trade-off between structure and ambiguity emerged. The "Original Offset, ISO Date" condition, which provided a standardized and unambiguous YYYY-MM base date, consistently yielded the highest EM accuracy at 33.55% (Table 4.1). Conversely, the fully naturalistic "Original Offset, Original Date" format produced the lowest EM score of 31.55%, demonstrating that even minor linguistic variability introduces processing hurdles that impair precise calculation. The format of the temporal offset itself appeared to be a secondary factor, as performance on these variants was comparable. This suggests the primary challenge for precision lies in grounding the calculation to a base date, a task that is significantly easier when that date is unambiguously structured.

Beyond simple precision, the most nuanced finding relates to the nature of the errors. The highly verbose "Total Months Word Offset, Full Words Date" format, despite its modest EM score, produced a remarkably low MAE in years of 18.68, significantly better than the baseline's 23.87 (Figure 4.2). The observation that a format yields higher error rates but lower error magnitudes highlights divergent types of model failures. Finally, across all syntactic variations, the model demonstrated a consistently high and robust Trend Accuracy (Figure 4.3), correctly identifying the temporal direction over 90.8% of the time. The findings show that the model maintains robust semantic reasoning about temporal direction but struggles with delicate computational tasks affected by surface input variations.

5.1.1. Possible Explanations for Observed Phenomena

Based on the performance patterns, we outline speculative hypotheses about the model's mechanisms to guide future investigations.

We hypothesise that the superior performance of the ISO format stems from two primary factors: tokenization alignment and pretraining prevalence. Structured formats like 2020-01 are often tokenized into predictable, discrete units, e.g., 2020, -, 01, which may align better with the model's numerical processing pathways. In contrast, natural language dates like "January 2020" can be

split into less consistent tokens by subword algorithms like Byte-Pair Encoding, potentially complicating interpretation (Kassner et al., 2021). Furthermore, the high prevalence of ISO-formatted dates in technical documents, code, and structured data within the pretraining corpus likely leads to more specialized and efficient processing routines for this specific format.

The low MAE-Year for the fully word-based format may be explained by a gap between semantic number interpretation and symbolic arithmetic operations. The model may be more adept at grounding the magnitude of large, spelled-out year numbers, e.g., "one thousand seven hundred", which are common in narrative text, than it is at executing precise, multi-step arithmetic with complex verbalized offsets, e.g., "thirty-nine months". Recent work has shown that while LLMs can be prompted to perform arithmetic, their underlying number sense can be fragile (Huang et al., 2024; Nogueira et al., 2021). When exact calculations fail on word-based inputs, the model's strong semantic grasp of the base year's approximate time period can still guide it to a roughly correct year, reducing year-level error. This would explain why the model gets the month wrong but the year is closer to the target.

5.1.2. Connection to Existing Literature

This thesis confirms and expands prior work on Large Language Models' temporal reasoning abilities and their sensitivity to input formats. The central finding provides strong evidence for the widely observed sensitivity of LLMs to input representation (Brown et al., 2020; Zhao et al., 2021; Liu et al., 2023). The shifts in EM accuracy based only on formatting differences in otherwise semantically identical queries underscore that models like Llama 3 do not yet operate with a fully abstracted, format-independent grasp of temporal information.

The observed sensitivity correlates with input format, as shown by enhanced EM performance on ISO 8601 base dates. This aligns with observations that LLMs can benefit from more structured and unambiguous inputs (Herzig et al., 2020), likely due to easier parsing and consistent tokenization. The study also adds a layer of nuance to discussions of LLM numerical reasoning (Wallace et al., 2019; Nogueira et al., 2021). While the fully word-based date format did not achieve peak EM accuracy (32.97%), its remarkably low Mean Absolute Error in year calculations (18.68) suggests the model might be better at estimating overall number size in natural language than performing precise calculations with complex verbal inputs, hinting at different processing behaviors.

Crucially, these findings highlight an underexplored dimension in current temporal reasoning benchmarks which, as noted in Section 2.2.1, rarely treat date format as a primary variable. By demonstrating the noticeable impact of such variations, this thesis suggests that existing benchmark scores may reflect not only reasoning competence but also a model's alignment with prevalent date formats. Future evaluations should incorporate more heterogeneous formatting to disentangle these factors.

5.2. Limitations of the Study

While this thesis provides valuable insights, it is important to acknowledge its limitations:

- **Single Language Model:** The experiments were conducted exclusively using the Llama 3 model. This specificity means that the observed sensitivities and performance patterns may not directly generalize to other Large Language Models. Performance characteristics could differ significantly for models of a substantially larger scale, e.g., Llama 3 70B, GPT-4o, those belonging to different architectural families, or base models that have not undergone instruction fine-tuning.
- **Constrained Dataset and Task Type:** The empirical investigation was concentrated on the Level 1 (Time-Time Relation) subset of the TempReason benchmark. As a result, the

findings may not be directly transferable to more complex temporal reasoning challenges, such as those requiring understanding of event-time (L2) or event-event (L3) relations, or to tasks demanding the retrieval of absolute date knowledge.

- **Limited Range of Date Format Variations:** The experiments reported here were performed on a focused subset of four distinct conditions. While chosen to represent a diverse array of syntactic structures, they do not encapsulate the full spectrum of possible date representations found in natural language or structured data.
- **Limited Evaluation Set Size:** The study utilized 4,000 questions derived from the TempReason L1 test set. While this provides a solid basis for analysis, it may not be large enough to fully capture the statistical diversity of all possible date formats. Consequently, the performance observed on the more complex or less common generated formats could be subject to sampling noise.
- **Potential Artificiality of Certain Generated Formats:** Some of the input formats created through systematic generation may occur with less frequency in naturally produced text. The model's observed performance on these potentially more artificial constructs might not perfectly mirror its capabilities when processing more common, real-world temporal expressions.
- **Dependency on a Specific Prompting Strategy:** The experimental results are contingent on the specific prompting style used, which instructed the model to provide an explanation before the final answer. Performance could change with a different prompt; for instance, a zero-shot query might yield lower accuracy, while allowing a free-form answer could reveal different error types.
- **Reliance on Specific Question Parsing Methodology:** The initial transformation of original TempReason L1 questions relied on a regular expression-based parser. Any inherent limitations in this parser could theoretically have influenced the characteristics of the generated date format variations.
- **Absence of Pretraining Corpus Analysis:** As defined by the scope of this research, an analysis of the Llama 3 pretraining corpus was not conducted. Therefore, suggestions that frequent date formats in training data influence model behavior remain theoretical.
- **Lack of Formal Statistical Significance Testing:** The comparison of performance metrics across the different format conditions was based on the direct observation of numerical results, e.g., differences in Exact Match percentages and Mean Absolute Error scores. This study did not employ formal statistical tests, such as a paired bootstrap resampling or a McNemar's test, to determine whether the observed differences in performance are statistically significant. Consequently, while the magnitude of the observed effects provides indicative evidence of the impact of date formats, definitive claims about the statistical significance of these variations cannot be made. The conclusions drawn, therefore, rely on the apparent trends and the extent of the numerical differences rather than on formal hypothesis testing.
- **Lack of Full Reproducibility:** The experiments were conducted without explicitly setting a global random seed. Furthermore, key decoding parameters (`temperature`, `top_p`) were left to their library defaults. Consequently, due to the stochastic nature of the sampling-based decoding, the exact text of the model's predictions may vary slightly if the inference script is run again. While the primary evaluation metrics are expected to be broadly stable against these minor variations, the results are not bit-for-bit reproducible, which is a key limitation of this study.

5.3. Future Work

The findings of this thesis, demonstrating that syntactic format is an active variable in temporal reasoning, carry practical implications for the research community and propose a structured plan for future study. This work points toward concrete improvements in how temporal reasoning is evaluated and modeled in LLMs.

5.3.1. Implications for the Benchmark Designers

Based on the evidence that format significantly impacts performance, we propose the following recommendations:

- **Incorporating Format Diversity:** Datasets should intentionally include a wide variety of representations, expanding from the variations in this study to include common international formats, e.g., DD-MM-YYYY, MM/DD/YY, dates with contextual elements, e.g., "Tuesday, January 15, 2020", and more complex natural language expressions, e.g.,"the second Friday of next month".
- **Reporting Per-Format Scores:** Evaluation reports on temporal benchmarks should include a breakdown of performance across different date format categories. This would allow for a more nuanced understanding of a model's strengths and weaknesses, separating reasoning competence from reliance on syntactic form.

5.3.2. For Model Developers

To address the observed format sensitivity, a key practical objective is to develop strategies that actively train more robust models. This involves:

- **Robustness through Data Curation:**Actively fine-tuning models with datasets that incorporate a balanced variety of challenging date formats is a direct way to address format sensitivity.
- **Exploring Advanced Training Strategies:** More advanced techniques should be investigated to foster a more abstract temporal representation. This includes experimenting with adaptive tokenizers that learn to handle date-specific patterns more effectively, or using contrastive fine-tuning, where the model is trained to recognize that different syntactic formats are semantically equivalent.

5.3.3. Directions for future research

In direct continuation of this work, we propose the following prioritized agenda for upcoming experiments:

1. **Replicate at Scale and Across Architectures:** The immediate next step is to systematically assess whether the observed sensitivities are specific to the Llama 3 model or represent a more general phenomenon. This involves replicating this study using larger, state-of-the-art models, e.g., Llama 3 70B, GPT-4o, to see if sensitivity diminishes with scale. Furthermore, the experiment should be extended to models from diverse architectural families and to non-instruction-tuned base models to differentiate the effects of architecture and fine-tuning from fundamental pretraining characteristics.
2. **Extend to Diverse and Realistic Contexts:** To assess performance in more practical scenarios, the experimental scope must be broadened to include more varied linguistic and data conditions. This includes:

- **Investigating Multilingual Contexts:** The analysis should be extended to non-English languages, testing common date formatting conventions like 15.03.2020 (German) to evaluate the cross-lingual generalizability of the findings.
- **Evaluating on Noisy and Real-World Data:** The experiment should be repeated with noisy, OCR-style date formats, e.g., Ianuary 2O20, 2020-O1. This would test the model's resilience to the kinds of imperfect data encountered in real-world document processing applications.

3. **Deepen the Analysis of Temporal Reasoning Processes:** Beyond varying the input, future work must investigate the reasoning process itself. This involves:
   - **Engaging with More Complex Temporal Tasks:** A vital next step is to investigate whether format sensitivity persists when LLMs engage with more complex reasoning. The analysis must be broadened to include the higher complexity tiers L2 and L3 of TempReason and additional benchmarks like TimeQA, which challenge models to merge factual knowledge with sequential event reasoning.
   - **Exploring Processing Mechanisms:** To transition from black-box observations to a mechanistic understanding, future work should leverage interpretability techniques. Examining attention mechanisms or mapping activation flows during date processing could reveal precisely where errors originate, e.g., in tokenization, numerical conversion, or arithmetic, and explain the performance differences, such as the low MAE of the "Full Words" format.

5.4. Conclusion

This thesis intends to answer the question of how different date formats affect Large Language Models' accuracy on temporal reasoning benchmarks. The experimental investigation, using Llama 3.2 3B Instruct and the TempReason L1 dataset, provides a clear affirmative: the syntactic format of date expressions is not a trivial detail for current LLMs but a significant factor influencing their accuracy in temporal reasoning tasks. While the model demonstrated a generally robust understanding of temporal directionality across all tested formats, its precision in date arithmetic was sensitive to the input representation. Structured, unambiguous formats for base dates, such as the YYYY-MM ISO standard, showed the highest EM accuracy, suggesting potential benefits from reduced parsing complexity or alignment with prevalent formats in training data. On the opposite, more naturalistic formats caused a slight decrease in outcomes. Notably, a highly verbose, fully word-based format, while not top-performing in exact matches, resulted in the smallest average error in the year component when mistakes were made, hinting at different error modalities based on input style. These findings emphasize that an LLM's performance on temporal tasks is a composite of its core reasoning abilities and its sensitivity to the surface form of the input. This offers practical relevance for prompt engineering, the design of more comprehensive evaluation benchmarks, and the ongoing development of LLMs towards a more abstract and robust understanding of time-dependent information. Ultimately, this research contributes to a deeper understanding of LLM behavior and highlights the continued need to bridge the gap between statistical pattern matching and true, human-like temporal comprehension.

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*.

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions.

Xiaoshu Chen, Sihang Zhou, Ke Liang, and Xinwang Liu. 2024. Distilling reasoning ability from large language models with adaptive thinking.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2024. TimeBench: A comprehensive evaluation of temporal reasoning abilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1204–1228, Bangkok, Thailand. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Raphael Gruber, Abdelrahman Abdallah, Michael Färber, and Adam Jatowt. 2024. Complextempqa: A large-scale dataset for complex temporal question answering.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. LARGE LANGUAGE MODELS CANNOT SELF-CORRECT REASONING YET.

Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023. Exploring the benefits of training expert language models over instruction tuning.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. BeliefBank: Adding memory to a pre-trained language model for a systematic notion of belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8849–8861, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

AI @ Meta Llama Team. 2024. The llama 3 herd of models.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Bernhard Nebel and Hans-Jürgen Bürckert. 1995. Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. *Journal of the ACM*, 42(1):43–66.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2021. Investigating the limitations of transformers with simple arithmetic tasks.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *CoRR*, abs/1909.01066.

James Pustejovsky. 2012. Reasoning about temporal and event information in clinical texts: (615572012-008).

James Pustejovsky, Robert Knippen, Jessica Moszkowicz, Roser Saurí, and Jessica Littman. 2005. Temporal and event information in natural language text. *Language Resources and Evaluation*, 39(2-3):123–164.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 21(140):1–67.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts.

Georgios Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2104–2115, Melbourne, Australia. Association for Computational Linguistics.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023a. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.

Wang-Chiew Tan, Jane Dwivedi-Yu, Yuliang Li, Lambert Mathias, Marzieh Saeidi, Jing Nathan Yan, and Alon Halevy. 2023b. TimelineQA: A benchmark for question answering over timelines. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 77–91, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Long Beach, CA, USA.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Yuqing Wang and Yun Zhao. 2024. TRAM: Benchmarking temporal reasoning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6389–6415, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Michael Zhang and Eunsol Choi. 2021. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Gong, and Xing Xie. 2024. Promptrobust: Towards evaluating the robustness of large language models on adversarial prompts. In *Proceedings of the 1st ACM Workshop on Large AI Systems and Models with Privacy and Safety Analysis*, LAMPS '24, page 57–68, New York, NY, USA. Association for Computing Machinery.

List of Figures

List of Tables

A. Supplementary Material

Code for Question Parsing

A.1. Regular Expression Pattern

```
pattern = re.compile(r'''
    (?P<offset_str>
        (?P<num1>\d+)\s*(?P<unit1>years?|months?)
        (?:\s*and\s*(?P<num2>\d+)\s*(?P<unit2>years?|months?))?
    )
    \s*
    (?P<direction>after|before)\s*
    (?P<date_str>
        (?P<month>[A-Za-z]+),\s*(?P<year>\d{4})
    )
''', re.IGNORECASE | re.VERBOSE)
```

A.2. Answer Normalization Function Code

The following Python code snippet presents the `_normalize_text` function used for standardizing model predictions and gold answers. The function relies on the `re` (regular expression) and `calendar` modules. The `_month_lookup` dictionary, which maps month name prefixes to month numbers, is pre-populated globally as shown.

Listing A.1: Python implementation of the `_normalize_text` function and its dependencies.

```
import re
import calendar

# Globally defined dictionary for month name/abbreviation to
   month number lookup
_month_lookup = {}
for month_idx in range(1, 13):
    month_num_str = f"{month_idx:02d}"
    month_forms = [
        calendar.month_name[month_idx].lower(),
        calendar.month_abbr[month_idx].lower().rstrip('.')
    ]
    for form in month_forms:
        # Create keys for prefixes of month names (e.g., "jan",
   "janu", ...)
        for prefix_len in range(3, len(form) + 1):
            _month_lookup[form[:prefix_len]] = month_num_str

def _normalize_text(txt: str) -> str:
```

```
    """
    Canonicalise various date strings to ISO 'YYYY-MM' where
possible,
    otherwise fallback to lowercased / whitespace-collapsed text
.

    Examples:
        "Mar, 1789"   -> "1789-03"
        "march 1789"  -> "1789-03"
        "1789-03-12"  -> "1789-03"
        "1789-03"     -> "1789-03"
    """
    if not txt or not isinstance(txt, str):
        return ""

    s = " ".join(txt.strip().lower().split()).replace("*", "")

    # 1) ISO patterns: YYYY-MM or YYYY-MM-DD
    m_iso = re.match(r"^(?P<year>\d{4})-(?P<month>\d{2})(?:-\d
{2})?$", s)
    if m_iso:
        return f"{m_iso.group('year')}-{m_iso.group('month')}"

    # 2) Month name patterns
    month_pattern = "|".join(re.escape(month) for month in
_month_lookup.keys())

    # Pattern: YYYY month_name [YYYY]?
    pattern = rf"^(?P<year1>\d{{4}})\s+(?P<month_name>{
month_pattern})[\.,]?\s*(?P<year2>\d{{4}})?$"
    m_name = re.match(pattern, s)
    if m_name:
        month_str = m_name.group("month_name")
        year_str = m_name.group("year1")
        month_num = _month_lookup.get(month_str)
        if month_num:
            return f"{year_str}-{month_num}"

    # Pattern: month_name YYYY
    m_month_year = re.match(rf"^(?P<month_name>{month_pattern})
[\.,]?\s+(?P<year>\d{{4}})$", s)
    if m_month_year:
        month_num = _month_lookup.get(m_month_year.group("
month_name"))
        if month_num:
            return f"{m_month_year.group('year')}-{month_num}"

    # 3) If no conversion matched, return the cleaned text
    return s
```

B. Submitted Software and Data Files

All software and data files submitted for this thesis can be found in the accompanying GitHub repository: https://github.com/hk-dv/thesis.git.

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, den 11.06.2025

. . . . . .

Hanna Kulik