ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240940

Generative LLMs for Multilingual Temporal Expression Normalization

Alejandro Sánchez de Castro[a,*], **Lourdes Araujo**[a,b,**] **and Juan Martinez-Romo**[a,***]

[a]Universidad Nacional de Educación a Distancia (UNED), 28040, Madrid
[b]Instituto Mixto UNED-ISCIII IMIENS

Abstract.

Assigning a numerical value to a temporal expression (TE), known as temporal expression normalization, is a crucial process for tasks like timeline creation and temporal reasoning. Rule-based and classical deep-learning normalization systems lack versatility because they are limited to specific domains and languages, while current Large Language Models (LLMs) solutions are relatively unexplored.

To overcome the current limitations in adaptability, we suggest utilizing five of the latest generative Large Language Models (LLMs) - Mistral 7B, Gemma 7B, Gemma 2B, Phi-2, and Llama-3 8B. We have explored various performance enhancement strategies, including using different prompts, contexts, and training techniques like Neftune. Our proposed models demonstrate the ability to adapt to diverse domains (news and biomedical) and multiple languages (Spanish, English, Italian, French, Portuguese, Catalan, and Basque) simultaneously. These models can handle expressions in various domains and languages, making them more versatile and useful for a wide range of applications. As a result, our approach offers significant performance improvements when compared to existing LLM-based and rule-based solutions for TE normalization and a promising solution for the challenges of temporal normalization.

1 Introduction

A temporal expression (TE) denotes a linguistic construction or phrase within a sentence or discourse conveying details regarding date, time, duration, or sets. Each TE corresponds to a specific value; for example, *"25 August 2001"* translates to the value "2001-08-25". The process of determining this value is known as TE normalization. TimeML [29] stands as an ISO standard incorporating the TIMEX3 tag, which outlines the annotation criteria and methodology for TEs.

There are a number of factors that make this a complicated task. On one hand, fundamental constructions like *"today"* or *"yesterday"* are efficiently handled by current systems. On the other hand, there exists a wide array of expressions and various methods of expressing identical concepts, such as *"a while"*, *"some time"*, *"a moment"* or *"a period"*. Additionally, certain TEs necessitate anchoring as they lack adequate information for normalization; for instance, in *Next 25 of August"*, knowledge of the current month is required to identify the subsequent August.

TEs, such as *"5 hours later"*, may have varying interpretations; it could be regarded as denoting a specific time or a duration, depending on the annotator's perspective. Consequently, combining corpora annotated by different interpreters could result in conflicting information.

Context, including instances like the sentence *"On the 3rd and 4th day"* which includes two TEs *3rd* and *4th day*, requires attention. The model in charge of normalizing these TEs necessitates an understanding that *3rd* pertains to *day*. Therefore, systems responsible for normalizing expressions must grasp context complexities, presenting a challenge even for sophisticated linguistic models like ChatGPT [1].

Normalizing TEs proves highly beneficial across various tasks requiring temporal sequencing, including generating timelines [26], text summarization [1], and question answering [6]. Moreover, it holds pivotal significance in enhancing reasoning abilities, as furnishing systems with temporal awareness is an essential stride toward fostering reasoning capabilities. Understanding the temporal sequence of events is fundamental for applying both induction and deduction effectively.

So far, rule-based systems like those referenced in [34, 27] have been dominant in the field of normalization, offering precise adjustments tailored to specific domains. However, these architectures are highly susceptible to changes in domain and language, necessitating labor-intensive manual crafting of new rules for adaptation. Efforts to shift towards machine and deep learning solutions, as mentioned in [28, 8], have faced obstacles hindering their progress. These obstacles include the limited availability of hand-annotated data, particularly for non-English languages, along with a dearth of research conducted for these languages with low resources. Moreover, the representation of date-formatted values complicates the adaptability of deep learning architectures. Furthermore, addressing this task requires significant linguistic ability, as observed, and conventional deep-learning architectures have proven inadequate. Finally, there has been a minimal exploration of the Large Language Models (LLMs) approach, and those like [17, 9] require a post-process for anchoring the TEs. This process is required because these models work with the operations needed to normalize the TEs, in the case of *"tomorrow"*, these models will predict the operation of adding one day to the reference date, but they won't perform this operation. On the other hand, existing research has not extensively explored the recent generative model's capabilities.

The goal of this work is to address the issue of the TE normalization task's lack of adaptability to different languages and domains by utilizing generative LLMs.

To accomplish this goal we will conduct training and explore the

* Corresponding Author. Email: asanchez@lsi.uned.es
** Corresponding Author. Email: lurdes@lsi.uned.es
*** Corresponding Author. Email: juaner@lsi.uned.es

[1] https://chat.openai.com/

adaptability capabilities of five different generative LLMs: Mistral 7B [14], Gemma 7B and 2B [36], Phi-2 [2] and Llama-3 8B [3]. These models will be trained to generate TEs values using two renowned TE corpora—E3C [22] and Timebank [32]. We aim to enhance the models' performance through diverse prompts, contexts, and training techniques like Neftune [13]. Finally, we will demonstrate how our proposed models simultaneously adapt to two different domains: news and biomedical and to multiple languages: Spanish, English, Italian, French, Portuguese, Catalan and Basque, surpassing the performance of current TE normalization LLM-based and rule-based solutions in all the mentioned languages and domains. Additionally, this approach overcomes the need for a post-process anchoring system present in the current LLM and rule-based solutions, greatly simplifying the inference pipeline while minimally impacting its performance. The final models are publicly available in [4].

2 Related Work

The TimeML ISO standard [29], stands as the predominant framework for TEs, delineating them through the employment of the *TIMEX3* tag. This tagging mechanism categorizes TEs into four distinct types: *DATE ("12 April"), DURATION ("2 months"), TIME ("24 hours"), and SET ("each day"*, featuring various attributes, with particular emphasis placed on the *value* attribute.

TimeML exhibits restrictive criteria for delineating TEs, yet it adopts a comparatively more lenient approach in delineating their values, thereby leaving it more open to interpretation.

TEs pose three distinct challenges: detection, classification, and normalization. Detection involves identifying the TE within the text, whereas classification pertains to categorizing it as a date, time, duration, or set. The combination of detection and classification is often denoted as extraction. Normalization, on the other hand, is the task of obtaining the value of the TE. These challenges can be addressed on their own or in some combination with rule-based systems, machine learning or deep learning techniques.

Rule-based systems like [34, 27, 23]employ regular expressions for both extraction and normalization. This approach facilitates the selective annotation of expressions, specifying their types and values following the standards of annotation. Consequently, meticulous transfer of the annotation requirements is achievable with precision. However, it is worth noting that the construction of rules within rule-based systems demands significant manual crafting, rendering them less adaptable to varying linguistic structures, domain contexts and languages. This matter has been addressed, particularly concerning low-resource languages [33, 23], but a costly adaptation period is still required.

Various approaches leveraging machine learning have been explored to resolve the adaptability inherent problem in rule-based systems. While some methodologies, such as those outlined in [28, 18, 8] advocate for integrating machine learning techniques to address extraction and normalization, they are proven to exhibit similar performance to rule-based systems when evaluated against established test datasets and they do not show significantly higher adaptability. A notable limitation lies in the insufficient diversity and volume of training data available for these methodologies.

For deep learning systems, results have improved considerably over time. Works with more classical approaches such as those presented in [15, 16] exhibit similar performance to rule systems in terms of extraction while systems based on LLMs such as [35] present higher performance and versatility, as they require less training data to obtain sufficient performance.

The investigation into normalization has not received as much attention as detection and classification, especially for languages other than English. This imbalance can be attributed to the inherent complexity of normalization tasks within deep learning frameworks, which often present a narrower array of viable solutions. However, recent developments in LLMs have created novel avenues for exploration, as evidenced by the findings presented in [17]. In this work, the authors advocate for the utilization of an XLM-based model employing a fill-mask objective for token prediction, with each token being a part of the TE value. The outcomes underscore the potential efficacy of LLMs, particularly concerning low-resource languages, where they exhibit superior performance compared to HeidelTime.

For its part, the sequence-to-sequence or generative architecture has barely been explored for normalization. Mainly because TimeML is an annotation scheme, which, although it leaves some freedom of interpretation for annotators, presents many formatting constraints. Aligning a generative model with these constraints is not a trivial task as shown in [9]. They propose the use of a T5 model [31] for predicting the sequence of operations that are required to normalize a TE. This sequence of operations has to be resolved in a further step in order to obtain the final value. However, the latest generation of pre-trained generative instruction models such as Mistral [14], Phi-2 or Gemma [36] have proven to have high information retention and processing capabilities, facilitating the alignment of the model with the requirements of the annotation scheme. These models can be adapted to various non-generative tasks such as hate speech detection [5] and have proven great adaptability to multiple languages [24, 11, 10].

There are several ways to align the behavior of pre-trained models to various tasks, domains or languages. The one that requires the least computational effort is prompting. This technique makes use of the model's context window, in which any information can be indicated with natural language, explanation or guide that may be useful to solve a task. There are several techniques to guide the model through the context like *Directional Stimulus Prompting* [19], *Chain of Thought* [39], *Tree of Thoughts* [42, 21], *Retrieval Augmented Generation* [5], *GraphPrompt* [20] or *Self-Consistency* [38]. Other methods for aligning the model behavior require additional training, which is computationally more expensive, but may have a greater impact on the model's behavior, like *Fine-Tuning*, *NefTune* [13], *Reinforcement Learning from Human Feedback* [4] or *Direct Policy Optimization* [30].

Although there is a wide range of possibilities in terms of aligning, there is no general agreement on which techniques are best for each case, and a thorough analysis is necessary to decide which methodology is best to apply.

All in all, current systems lack sufficient adaptability to different domains and languages, which hampers their applicability. But the advent of the current generation of LLMs such as Mistral, Llama-3 or Gemma, together with different prompting and training techniques, offers a potential solution to the current limitations since they have not yet been studied.

[2] https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/

[3] https://ai.meta.com/blog/meta-llama-3/

[4] https://huggingface.co/asdc/[model_name]-multilingual-temporal-expression-normalization. With [model_name]: gemma-8B, gemma-2B, mistral-7B and Llama-3-8B

[5] https://ai.meta.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/

3 Proposed Approach

In this section, we will describe the backgrounds, motivations and justification of each technical and theoretical component. Also, we will describe all the experimentation and the followed methodology.

3.1 Prompt template and context for normalization

One of the key parts of fine-tuning a generative model is the chat template. We have opted to keep it minimal, giving a direct and brief explanation of the task commissioned and the information needed to normalize a TE.

A TE is composed of four parts: the string, the type, the value and the reference date as can be seen in figure 1. It is necessary to know three of them to predict the fourth. For example, for the TE *"Yesterday"*, type *DATE* and reference date *2001-08-25*, the value can be calculated by subtracting one from the reference date. For the same TE, if the type is unknown, it can be predicted by looking at the format of the value and the string. The string can also be predicted but with a much smaller confidence interval, since for the above example, the string can be either *"Yesterday"* or *"24th of August"*. The same applies to predicting the reference date.

Sometimes the reference date is not available, and cases like the example above cannot be fully normalized. Such cases are assigned a generic value *XXXX-XX-XX (year-month-day)* for dates and *XXXX-XX-XX-XXTXX (year-month-dayThour)* for times. It may be the case that only part of the value is known, as for the expression *"August this year"*. If the reference date is unknown, the expression shall be assigned the value *XXXX-08-XXXX*. The corpora used to take this feature into account, which makes it easier to adjust the behavior of the model. In other cases, reference information can be found in the context of the expression. For example, in the sentence *"The patient will visit on days 3 and 4"* there are two expressions *"days 3"* and *"4"*. If one tries to normalize the expression *"4"* without taking into account the context, it is impossible to know what it is referring to.

Type	*String*	*Ref. Date*	*Value*
Date	Yesterday	20-04-1990	19-04-1990
Time	8 p.m.	03-12-2021	03-12-2021T20:00
Duration	Two days	25-08-2001	P2D

Figure 1. The Figure shows three examples of the four components of a TE.

With all this in mind, we have constructed a prompt that contains all the information necessary to normalize a time expression, as shown in Figure 2. Along with all the necessary parts for normalization, we have included a short description of the task, so that the model can better adapt to the instructions, both for training and for inference.

3.2 Temporal expression anchoring

TEs often lack complete information regarding their temporal value. For example, the expression *"25th August"* does not specify the year,

```
Given the temporal expression type, the reference date, the temporal
expression and the context phrase in which the time expression
appears, generate the value according to the TIMEX3 scheme. In the
context phrase the temporal expression appears, sometimes it will be
necessary to pay attention to the surrounding context in order to resolve
the expression.

### Context phrase:
{sentence}

### Temporal expression type:
{type}

### Temporal expression:
{expression}

### Reference date:
{dct}

### Value:
{value}
```

Figure 2. Prompt template used for training and inference with context phrase.

necessitating supplementary data known as a reference date. This reference date may be implicit, as seen in phrases like *"the patient was admitted on 25 August...the following day..."*, where *"the following day"* refers to the 26th of August. However, in many cases, textual context lacks implicit temporal information, with the document creation time (DCT) being the sole reference date. For instance, if the document was created on 3rd December 2021, the phrase *"the following day"* would be interpreted as referring to 4th December 2021.

Normalization systems typically integrate an anchoring mechanism, such as in [27, 34] or the systems proposed at [17, 9], where they extract an incomplete TE value solely from the TE text, termed as the context intermediate representation (CIR). For instance, for *"25th of August"*, the CIR could be "XXXX-08-25". Subsequently, this value is anchored to the reference date, whether it be the DCT or another TE. Moreover, the anchoring mechanism must be capable of performing date operations, such as the addition or subtraction of days, weeks, months, years, hours, seasons, etc. These anchoring mechanisms are often costly to develop and are specific to each normalization system and how operations are defined, making them poorly reusable.

Works such as [40, 25] have shown that the previous generation of LLMs (BERT family and T5) struggle to perform mathematical arithmetic operations. They need to be fine-tuned and adjusted in order to obtain proper behavior on addition and subtraction, while [17, 9] have shown that these LLMs need an anchoring system to properly solve the TE normalization adequately. However, recent generative LLMs have proven to have better abilities to perform basic mathematical arithmetic operations as shown in [3, 41].

On the other hand, there are TEs that need access to a calendar in order to be normalized. For example, the TE *"the next week"* needs an auxiliary calendar to know to which week corresponds the reference date, or the expression *"the next Monday"* requires a calendar to know when the current Monday is.

After conducting some preliminary training and testing, we have found that the chosen LLMs are generally capable of solving these kinds of expressions, for which previous LLMs would require an auxiliary system. For instance, the models accurately solve *"the next Friday"*, *"Sunday"* or *"the past summer"*. This suggests that these

models have seen and memorized the calendar in their pre-training process. However, they are not infallible, and we have noticed some errors such as *"this weekend"* or *"first week of the month"*, although they are in the minority compared to the successes on the test sets. Therefore we have dispensed with an anchoring system.

3.3 Multilingual adaptability

While recent generative LLMs like Mistral, Phi-2 or Gemma are trained primarily in English, they show amazing multilingual capabilities. As studied in [24], where they evaluate a great variety of recent LLMs on a silver standard benchmark for basic open-ended question answering with 27.4k test questions through 137 languages, finding that exclusively English trained models answered faithfully to other languages questions. On [11], the authors explore the potential and limitations of GPT-3 across three tasks: extractive Question-Answering, text summarization, and natural language generation in five distinct languages: German, Spanish, Russian, Turkish and Catalan. The findings reveal that GPT-3 demonstrates utility beyond English, proving effective even for languages with limited training data. As shown in [10], monolingual models can undergo adaptation to other languages by leveraging the source language. The source language enriches the syntactic and semantic understanding of the target language.

Hence, the multilingual adaptability capacity inherent in recent LLMs has the potential to mitigate the deficiency in adaptability observed in existing TE normalization solutions. Adapting to these models does not necessitate an in-depth examination of language features and TE structures, unlike rule-based systems, making adaptability significantly more straightforward.

3.4 Corpora

For training and testing our models we have used two well-known public multilingual corpora: European Clinical Case corpus (*E3C*) [6] from the biomedical domain, specifically built from clinical narratives, and Timebank (*TB*) [7] from the news domain. These corpora have manually annotated TEs each one with its corresponding value together with other TimeML entities such as events and temporal relations. Table 1 shows the distribution of TEs in both corpora. As can be seen, some languages overlap in both corpora.

	Language	Training	Test	Total
Timebank	English	1053	156	1209
	French	205	81	286
	Italian	522	126	648
	Spanish	1093	198	1291
	Catalan	1295	125	1420
	Portuguese	1082	145	1227
E3C	English	153	174	327
	French	118	157	275
	Italian	110	177	287
	Spanish	146	200	346
	Basque	222	404	626

Table 1. TE distribution in E3C and Timebank corpora based on each language

For its part, Figure 3 shows the distribution by type of TE on both corpora. As can be seen, the frequencies are inverted. In Timebank the type *DATE* is more frequent than the *DURATION* type and the type *TIME* is more frequent than the *SET* type. However, in E3C, this pattern is reverted. The reason for this difference could be the distinct domains of both corpora. In clinical narratives, *DATES* and *TIMES* may be less frequent because specific timestamps are often not given when describing the patient's clinical condition. Instead, the durations of treatments, recoveries, and each process are provided.

On the other hand, Figure 3 shows the distribution by type of TE through all languages. As can be seen, although the frequencies of the types are inverted when looking just at the domain, since the Timebank corpus is denser than E3C the *DATE* type is the most frequent in all languages, followed by the *DURATION* type. While *TIME* and *SET* are the minority. It should be noted that Basque is only present in the E3C corpus and it can be seen that *DURATIONS* are more frequent than *DATES*, as well as *SETS* are more frequent than *TIMES*. The opposite is true for Catalan and Portuguese, which are only available in the Timebank corpus.

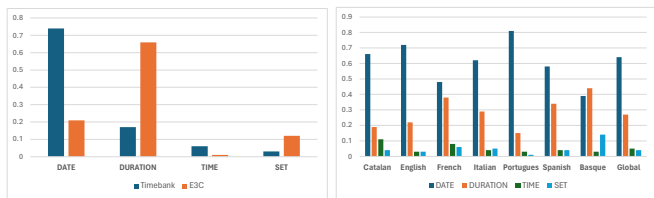

Figure 3. Distribution of TE type by corpora (left) and distribution of TE type by language through both corpora (right)

3.5 Experiments

A series of experiments have been set up to maximize the performance of the models, exploring their multi-domain and multi-language capabilities. Experimentation with 4 models has been considered: Mistral 7B, Gemma 7B, Gemma 2B and Phi-2.

1. First, we compare the domain adaptability of the models, testing whether the corpora complement or interfere with each other. We will do so by comparing the performance of the models on E3C and Timebank test sets after training them on separate corpora versus training them on both corpora. Showing how the models adapt to multiple domains simultaneously. The motivation behind this experiment is that as shown in [35], the performance of the model on a domain reaches its maximum when it is trained exclusively for that domain. When training on two domains, performance increases considerably in the domain that was not considered in the original training while it decreases in the domain that was trained on.
2. Second, we apply a well-known technique called *Neftune* [13]. This technique involves fine-tuning by introducing noise to the embedding vectors during training, thereby enhancing the overall performance of the model. As demonstrated by the authors, this straightforward yet impactful augmentation method has shown significant improvements in language model performance. The authors propose a tunable parameter α for adjusting the introduced noise into the embedding vectors. Their experimentation has shown three possible values for α, 5, 10 and 15, on which we will base our experiments. In this way, we will try to improve the baseline performance of the models and see if this technique can be useful for this purpose.

[6] https://github.com/hltfbk/E3C-Corpus
[7] https://github.com/AntonFagerberg/Temporal-Information-Extraction/tree/master/tempeval2-data

Both of these experiments have been done with the training data from all languages. This way we can evaluate and focus from the beginning on the multilingual capacities of the models.

3. In both of the exposed experiments, we don't give the models any context further from the TE itself, using the prompt shown in Figure 4. This approach allows us to more effectively isolate the impact of each experiment, as the introduction of a context may confuse the model's attention. However, as detailed in Section 3.1, incorporating the contextual phrase of the TE may prove essential for normalizing the value. Hence, we conduct an experiment where the contextual phrase is introduced both at the beginning and at the end of the prompt, as shown in Figure 2. The intuition behind this is that introducing the context phrase beforehand would enable the model to grasp the context prior to encountering the TE. Conversely, introducing the context phrase afterward would guide the model in identifying the TE's placement within the context and prompt it to allocate greater attention accordingly.
4. Finally, we propose an experiment where we train each model in multiple languages. The aim is to explore the multilingual capabilities of the models for the normalization of TEs, so we propose a comparison between training the models in a single language versus training them in all languages together, accumulating the techniques that have worked best in the previous experiments. The models will be evaluated on the languages in which they have been trained.

```
Given the temporal expression type, the reference date and the
temporal expression, generate the value according to the TIMEX3
scheme.

### Temporal expression type:
{type}

### Temporal expression:
{expression}

### Reference date:
{dct}

### Value:
{value}
```

Figure 4. Prompt template used for the first two experiments without context phrase.

Throughout the experiments, the performance of our proposed models will be compared with various state-of-the-art solutions such as *HeidelTime* [34], SuTime [2], UWTime [18], CogCompN [28], ARTime [8], DNPTime [9] and the model proposed at [17], denoted as *XLM_Bosch*, due to the absence of a designated name by the authors.

Regarding the preprocessing of the training corpora, the text has been split into sentences, choosing only those with at least one TE to minimize the dataset's size. We have trained for 1200 steps for the unique language experiments a 5400 for the multi-language experiments.

3.6 Methodology

In this Section, we will discuss all the details related to the training implementation, inference and evaluation.

For optimizing the computational and memory cost of training these models we have used *"low ranking adaptation" LoRA* [12, 7] together with 4 bits quantization with parameters $r = 8$, $alpha = 16$, $dropout = 0.05$, none *bias*, together with double quantization in *"nf4"*. For the decoding algorithm, we have chosen *beam search*, since it prioritizes sequences with higher probabilities. As the model has to predict the sequence of tokens that forms the value, we deem this option as the most suitable. We have tried to adjust the size of the input as much as possible, on the one hand, to save memory and computational costs and on the other hand not to impair the performance of the model with useless information. Therefore, a length of 410 tokens has been used as input for training based on the lengths of the constructed prompts, with padding on the left. We have used the same hyperparameters throughout all the conducted experiments. A $learning\_rate = 2.5e-5$, 5 $warmup\_steps$, $batch\_size = 1$, $seed = 42$ and *paged adamw 8 bit optimizer*.

The prompt used for inference is the same as the ones used for training, as shown in Figures 2 and 4 without passing the value. The predictions were generated individually rather than in batches to prevent prompts from interfering with each other.

In previous experimentation, we have tried *self-consistency* [38] to improve the quality of the predictions but we found that the use of $temperature = 1$ offered an equal performance for TE normalization. This may be due to the precision needed to normalize expressions, as the models have no room for improvisation.

We have repeated each experiment three times and taken the mean value to reduce randomness. For evaluation we have used the widespread metric from Tempeval-3 [37]. We report the accuracy of TE normalization, which in this case is equivalent to the F1-score. All the mean values are weighted based on the number of expressions of each corpus so that the performance of the largest corpus does not affect the performance of the smaller ones. All the conducted experiments have been run on a configuration of two *NVIDIA RTX 3090 24GB GPUs*.

4 Results

In this Section all the results from the experiments proposed in Section 3.5 will be presented.

4.1 Domain adaptability

The results of training five models on each corpus separately and on both corpora are presented in Table 2. It can be observed that training with both corpora enhances the performance compared to training with the corpora separately. This improvement is greater for the Timebank corpus than for the E3C, with a corresponding intra-model mean improvement of 3 and 1.95 points. This indicates that Timebank draws more from E3C's expressions than E3C draws from Timebank's expressions. On the other hand, it can be observed that the Gemma models perform better at E3C but worse at Timebank than Mistral 7B, while Llama-3 falls in between.

These findings suggest that the new generation of LLMs is more domain-independent than the RoBERTa model described in [35], as it can better understand and use a wide variety of expressions.

4.2 Neftune

The data presented in Table 3 demonstrates that the performance of the Gemma models is significantly enhanced with the use of Neftune, while Llama-3 8B, Mistral 7B and Phi-2 models show a decrease

	E3C		Timebank	
	Merged	Separated	Merged	Separated
Mistral 7B	58.89	57.36	**84.53**	81.73
Gemma 7B	**67.07**	65.75	81.67	79.44
Gemma 2B	62.93	60.33	74.49	70.84
Phi-2	56.86	53.82	61.58	57.57
Llama-3 8B	63.83	62.58	83.56	81.27

Table 2. Results on E3C and Timebank comparing the performance of five models when training over E3C and Timebank merged or separated. The best results for each row are underlined and the best overall result for each corpus is marked in bold.

in performance. This suggests that there are some architectural differences between these models, which cause the Gemma family to benefit from Neftune.

It is interesting to note that $\alpha = 5$ seems to be the best option for both Gemma models, Llama-3 8B and Phi-2, but in the case of Mistral 7B, $\alpha = 15$ seems to be a better option for applying Neftune.

In conclusion, the findings suggest that the use of Neftune could be beneficial for some models, even for downstream tasks such as TE normalization. However, it cannot be concluded when to apply it and which α parameter is better.

	Steps	$\alpha = 5$	$\alpha = 10$	$\alpha = 15$	No α
Mistral 7B	800	60.55	62.09	63.43	**71.64**
	1000	63.86	62.75	64.07	71.42
	1200	63.64	62.76	64.08	71.42
Gemma 7B	800	73.87	73.87	73.65	73.87
	1000	**75.03**	74.34	74.57	73.65
	1200	74.57	74.57	74.34	74.33
Gemma 2B	800	67.52	64.30	64.30	67.29
	1000	68.22	64.29	65.63	68.68
	1200	**68.92**	66.08	65.42	68.68
Llama-3 8B	800	71.51	71.87	71.36	72.96
	1000	72.24	72.38	72.45	72.96
	1200	73.13	72.91	73.06	**73.65**
PHI-2	800	50.33	39.88	41.85	**59.21**
	1000	52.38	39.49	43.42	58.31
	1200	52.16	43.03	44.60	58.51

Table 3. Results comparing the performance over training five different models on 800, 1000 and 1200 steps with the Neftune parameter $\alpha = 5, 10, 15$ and without Neftune. The best results for each row are underlined and the best model performance is underlined and marked in bold. The models have been trained on both Timebank and E3C multilingual training sets. The results are the weighted mean based on the number of TEs in Timebank and E3C test sets for each language.

Due to the poor results of Phi-2 in comparison to the other models, it will not be contemplated for the rest of the experiments.

4.3 Context phrase

Table 5 shows a comparison between training both Gemma and Mistral 7B models with the context phrase at the end of the prompt versus at the beginning. It can be seen how introducing the context phrase at the beginning offers a 2.92 point mean performance increase across models.

When looking at the errors we can see that the model properly solves TEs where the context is necessary. For example the Spanish sentence *"obligan a los obreros a trabajar hasta 12 o 14 horas"* (*"force the workers to work up to 12 or 14 hours"*) includes two TEs *"12"* and *"14 hours"*. With both the context phrase at the beginning and the end, the models are capable of normalizing both TEs, relating *"12"* with *"hours"*. This kind of context-dependent TE is properly solved by passing both the context at the beginning and the end.

The difference in performance between both options comes from non-context-dependent expressions. The intuition behind this behavior may be explained by the fact that generative models pay more attention to the end of the prompt. As such, prompting the context phrase at the end adds too much noise to the prompt. However, when the context phrase is introduced at the beginning, the model is better able to understand the context while retaining the rest of the prompt.

It is worth mentioning that with the context phrase at the beginning, Mistral 7B and Gemma 2 B's performance worsens when compared to not using the context phrase. This can be seen when comparing the best performance in Table 3 with the performance of the first column in Table 5.

Finally, it should be noted that the models have been trained on the best neftune parameter founded on the previous experiment. That is $\alpha = 5$ for Gemma 7B and 2B and no neftune for Mistral 7B and Llama-3 8B.

4.4 Multilingual adaptation

Table 4 compares the performance of the models trained in a single language versus models trained in all languages together. The Table also shows how the models perform when trained in Spanish and tested on all languages, highlighting how well the models adapt to unseen languages. Additionally, the Table compares the models with the current state-of-the-art multilingual normalization solution proposed in [17], which trains an XLM-RoBERTa model on multiple languages at once.

As can be seen, the best overall model is Gemma 7B, while Mistral 7B outperforms in five of seven languages. It's worth noticing the bad performance of Mistral 7B on Basque, with around 20 points of difference from Gemma 7B and the same peak performance as Gemma 2B.

On the other hand, training on multiple languages at one time improves the performance over training on each language separately. Except for Basque, where training only with Basque on Gemma 2B, Mistral 7B, and Llama-3 8B yields substantial improvements.

Moreover, the models trained only in Spanish adapt well to other languages, with Gemma 7B being the best-performing model, staying close to XLM_Bosch. However, Gemma 2B shows the greatest difference between training only in Spanish versus training with all languages, indicating that smaller models have a worse adaptation to unseen languages. Interestingly, the performance of the model is worse when trained only in Italian than when trained only in Spanish, possibly due to a lack of variability or quantity in the training annotations. Additionally, Basque shows a greater difference between training only in Spanish and training with Basque data, demonstrating that the models are better adapted to similar languages.

Finally, as mentioned earlier, the normalization task has been extensively studied in English. As a result, most of the currently available normalization systems are available only in English. We, therefore, compare our best English model with other well-known monolingual models in Table 6, where it can be seen how our proposed solution is 4.58 points above the second best system. This shows how our proposed solution outperforms current normalization systems not only in multilingual form but also in monolingual form.

		Catalan	English	Italian	French	Spanish	Basque	Portugues	Mean
Mistral 7B	Together	75.3	72.61	82.18	73.53	88.19	52.48	84.83	74.31
	Separate	67.2	68.84	48.52	68.90	86.43	59.9	82.07	68.31
	Sp-Zero-Shot	70.04	61.90	79.54	68.10	86.43	29.7	68.97	64.79
Gemma 7B	Together	73.6	69.26	80.20	72.81	85.93	78.22	82.07	**77.91**
	Separate	73.6	67.33	44.89	68.07	86.68	75.99	80	71.03
	Sp-Zero-Shot	65.6	60.67	75.91	67.22	86.68	45.05	64.14	66.51
Gemma 2B	Together	64.8	65.58	77.23	69.75	79.90	56.19	75.17	69.55
	Separate	55.2	63.05	51.48	63.02	76.46	59.9	71.72	63.48
	Sp-Zero-Shot	45.6	47.60	44.75	55.46	76.46	20.79	40.69	47.81
Llama-3 8B	Together	76.8	70.74	82.18	70.58	83.92	71.52	82.76	76.65
	Separate	71.2	67.62	47.85	64.29	80.65	74.26	82.07	69.49
	Sp-Zero-Shot	71.2	55.24	73.45	65.96	80.65	55.45	55.17	65.66
XLM_Bosch	Together	67.2	65.60	76.36	62.87	76.25	55	79.48	68.06

Table 4. Results comparing the performance of models trained in a single language (Separate) versus models trained in all languages together (Together), and models trained only in Spanish and tested on all languages (Sp-Zero-Shot). These results are compared with the baseline system XLM_Bosch. The best mean result is marked in bold, and the best results for each language are underlined.

	Context phrase end	Context phrase beginning
Mistral 7B	71.50	**74.31**
Gemma 7B	76.07	**77.91**
Gemma 2B	65.44	**69.55**
Llama-3 8B	74.83	**76.65**

Table 5. Results comparing the performance of the three models over training with the context phrase at the beginning and at the end. The best results for each row are marked in bold and underlined for each column. The models have been trained on both Timebank and E3C multilingual training sets. The results are the weighted mean based on the number of TEs in Timebank and E3C test sets for each language.

	English Timebank
HeidelTime	76.1
SuTime	70.3
UWTime	82.6
CogCompN	83.4
ARTime	75.4
DNPTime-Large	80.4
XLM_Bosch	71.8
Ours Best (Mistral 7B)	**87.18**

Table 6. Comparison on English Timebank of our best English model versus other monolingual normalization systems. The best result is marked in bold and the second one is underlined.

5 Conclusions and Limitations

In this work, we achieve the best results for multilingual TE normalization on two different corpora, E3C and Timebank. To accomplish this, we have adapted the latest generative LLMs to the normalization task. We have conducted a series of experiments and proved that this kind of architecture can be directly used for monolingual, multilingual and multidomain normalization, overcoming the current adaptability limitations of rule-based systems. After analyzing the results, it can be concluded that no single model is superior in all aspects of language and domain. To get the best outcome, it is recommended to use the model that suits the target language and domain. However, if there is a need to use any of these models for a non-target domain or language, it is recommended to use Gemma 7B, as it has shown the best overall performance.

We have considered certain limitations. First, the proposed solution does not include an anchoring system. This simplifies the normalization system but limits its performance for certain expressions where a calendar is required, as exposed in Section 3.2. For future work, we plan to include an anchoring system along with the proposed LLMs to explore the capabilities of this architecture and study its performance.

Also, we explore the use of the context phrase. However, there may be certain TEs that need from a longer context to be normalized. From the studied corpora, this is a very rare condition, and from the results presented in Section 4.3 it can be concluded that a larger context may worsen the performance. Therefore, we propose to study how to increase the available context for normalizing TEs without degrading the performance. We also plan to use a Retrieval Augmented Generation (RAG) system to retrieve the most relevant context for the temporal expression. In this way, the context phrase can be shortened and optimized.

Finally, each experiment was run only three times to minimize randomness while minimizing environmental impact.

Acknowledgments

This work has been funded by the following projects DOTT-HEALTH (MCI/AEI/FEDER, UE with identification PID2019-106942RB-C32), OBSER-MENH (MCIN/AEI/10.13039/501100011033 and NextGenerationEU"/PRTR with identification TED2021-130398B-C21), SICAMESP (with identification 2023-VICE-0029) and by the project EDHER-MED (with identification PID2022-136522OB-C21)".

References

[1] C. Barros, E. Lloret, E. Saquete, and B. Navarro-Colorado. Natsum: Narrative abstractive summarization through cross-document timeline generation. *Information Processing & Management*, 56(5):1775–1793, 2019.

[2] A. X. Chang and C. Manning. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3735–3740. European Language Resources Association (ELRA), May 2012.

[3] V. Cheng and Z. Yu. Analyzing ChatGPT's mathematical deficiencies: Insights and contributions. In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP), Oct. 2023. URL https://aclanthology.org/2023.rocling-1.22.

[4] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences, 2023.

[5] C. Christodoulou. Nlpdame at climateactivism 2024: Mistral sequence classification with peft for hate speech, targets and stance event detection. In *CASE*, 2024. URL https://api.semanticscholar.org/CorpusID:268417305.

[6] J. R. Cole, A. Chaudhary, B. Dhingra, and P. Talukdar. Salient span masking for temporal understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3052–3060. Association for Computational Linguistics, May 2023. doi: 10.18653/v1/2023.eacl-main.222. URL https://aclanthology.org/2023.eacl-main.222.

[7] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.

[8] W. Ding, J. Chen, J. Li, and Y. Qu. Automatic rule generation for time expression normalization. *Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021*, pages 3135–3144, 8 2021. doi: 10.48550/arxiv.2108.13658. URL https://arxiv.org/abs/2108.13658v3.

[9] W. Ding, J. Chen, L. E, J. Li, and Y. Qu. Time expression as update operations: Normalizing time expressions via a distantly supervised neural semantic parser. *Knowledge-Based Systems*, 278: 110870, 2023. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2023.110870. URL https://www.sciencedirect.com/science/article/pii/S0950705123006202.

[10] E. Gogoulou, A. Ekgren, T. Isbister, and M. Sahlgren. Cross-lingual transfer of monolingual models. In *International Conference on Language Resources and Evaluation*, 2021. URL https://api.semanticscholar.org/CorpusID:237513421.

[11] C. Holtermann, P. Röttger, T. Dill, and A. Lauscher. Evaluating the elementary multilingual capabilities of large language models with multiq. *ArXiv*, abs/2403.03814, 2024. URL https://api.semanticscholar.org/CorpusID:268253307.

[12] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models, 2021.

[13] N. Jain, P. yeh Chiang, Y. Wen, J. Kirchenbauer, H.-M. Chu, G. Somepalli, B. R. Bartoldson, B. Kailkhura, A. Schwarzschild, A. Saha, M. Goldblum, J. Geiping, and T. Goldstein. Neftune: Noisy embeddings improve instruction finetuning, 2023.

[14] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b, 2023.

[15] Z. M. Kim and Y.-S. Jeong. Timex3 and event extraction using recurrent neural networks. In *2016 International Conference on Big Data and Smart Computing (BigComp)*, pages 450–453, 2016. doi: 10.1109/BIGCOMP.2016.7425968.

[16] L. Lange, A. Iurshina, H. Adel, and J. Strötgen. Adversarial alignment of multilingual models for extracting temporal expressions from text. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 103–109. Association for Computational Linguistics, July 2020. doi: 10.18653/v1/2020.repl4nlp-1.14. URL https://aclanthology.org/2020.repl4nlp-1.14.

[17] L. Lange, J. Strötgen, H. Adel, and D. Klakow. Multilingual normalization of temporal expressions with masked language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1174–1186. Association for Computational Linguistics, May 2023. doi: 10.18653/v1/2023.eacl-main.84. URL https://aclanthology.org/2023.eacl-main.84.

[18] K. Lee, Y. Artzi, J. Dodge, and L. Zettlemoyer. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1135. URL https://aclanthology.org/P14-1135.

[19] Z. Li, B. Peng, P. He, M. Galley, J. Gao, and X. Yan. Guiding large language models via directional stimulus prompting. *ArXiv*, abs/2302.11520, 2023. URL https://api.semanticscholar.org/CorpusID:257079124.

[20] Z. Liu, X. Yu, Y. Fang, and X. Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks, 2023.

[21] J. Long. Large language model guided tree-of-thought, 2023.

[22] B. Magnini, B. Altuna, A. Lavelli, M. Speranza, and R. Zanoli. The e3c project: Collection and annotation of a multilingual corpus of clinical cases. In *CLiC-it*, 2020.

[23] B. Mansouri, M. S. Zahedi, R. Campos, M. Farhoodi, and M. Rahgozar. Parstime: Rule-based extraction and normalization of persian temporal expressions. *Lecture Notes in Computer Science*, 10772 LNCS:715–721, 2018. ISSN 16113349. doi: 10.1007/978-3-319-76941-7_67/TABLES/3. URL https://link.springer.com/chapter/10.1007/978-3-319-76941-7_67.

[24] J. R. G. Mendoncca, P. Pereira, J. P. Carvalho, A. Lavie, and I. Trancoso. Simple llm prompting is state-of-the-art for robust and multilingual dialogue evaluation. *ArXiv*, abs/2308.16797, 2023. URL https://api.semanticscholar.org/CorpusID:261395306.

[25] M. Muffo, A. Cocco, and E. Bertino. Evaluating transformer language models on arithmetic operations using number decomposition. *ArXiv*, abs/2304.10977, 2023. URL https://api.semanticscholar.org/CorpusID:251406206.

[26] M. Najafabadipour, M. Zanin, A. Rodríguez-González, M. Torrente, B. Nuñez García, J. L. Cruz Bermudez, M. Provencio, and E. Menasalvas. Reconstructing the patient's natural history from electronic health records. *Artificial Intelligence in Medicine*, 105: 101860, 2020. ISSN 0933-3657. doi: https://doi.org/10.1016/j.artmed.2020.101860. URL https://www.sciencedirect.com/science/article/pii/S0933365719311467.

[27] M. Navas-Loro, V. Rodríguez-Doncel, D. Pinto, V. Singh, and F. Perez. Annotador: a temporal tagger for spanish. *J. Intell. Fuzzy Syst.*, 39(2): 1979–1991, jan 2020. ISSN 1064-1246. doi: 10.3233/JIFS-179865. URL https://doi.org/10.3233/JIFS-179865.

[28] Q. Ning, B. Zhou, Z. Feng, H. Peng, and D. Roth. CogCompTime: A tool for understanding time in natural language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 72–77. Association for Computational Linguistics, Nov. 2018. doi: 10.18653/v1/D18-2013. URL https://aclanthology.org/D18-2013.

[29] J. Pustejovsky, K. Lee, H. Bunt, and L. Romary. ISO-TimeML: An international standard for semantic annotation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA), May 2010. URL http://www.lrec-conf.org/proceedings/lrec2010/pdf/55_Paper.pdf.

[30] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model, 2023.

[31] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020. ISSN 1532-4435.

[32] R. Saurí, E. Saquete, and J. Pustejovsky. Annotating time expressions in spanish. timeml annotation guidelines (version tempeval-2010). Technical report, Barcelona Media Technical Report 2010-02, 2010.

[33] L. Skukan, G. Glavaš, and J. Šnajder. Heideltime. hr: extracting and normalizing temporal expressions in croatian. In *Proceedings of the 9th Slovenian Language Technologies Conferences (IS-LT 2014)*, pages 99–103, 2014.

[34] J. Strötgen and M. Gertz. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 321–324, 2010.

[35] A. Sánchez-de Castro-Fernández, L. Araujo Serna, and J. Martínez Romo. Robertime: A novel model for the detection of temporal expressions in spanish, 2023.

[36] G. Team. Gemma: Open models based on gemini research and technology, 2024.

[37] N. UzZaman, H. Llorens, L. Derczynski, J. Allen, M. Verhagen, and J. Pustejovsky. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9. Association for Computational Linguistics, June 2013. URL https://aclanthology.org/S13-2001.

[38] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models, 2023.

[39] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. H. hsin Chi, F. Xia, Q. Le, and D. Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022. URL https://api.semanticscholar.org/CorpusID:246411621.

[40] P. Yang, Y. Chen, Y. Chen, and D. M. Cer. Nt5?! training t5 to perform numerical reasoning. *ArXiv*, abs/2104.07307, 2021. URL https://api.semanticscholar.org/CorpusID:233240851.

[41] Z. Yang, M. Ding, Q. Lv, Z. Jiang, Z. He, Y. Guo, J. Bai, and J. Tang. Gpt can solve mathematical problems without a calculator, 2023.

[42] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.