Procesamiento del Lenguaje Natural, Revista nº 70, marzo de 2023, pp. 39-51 recibido 12-12-2022 revisado 31-01-2023 aceptado 09-02-2023

RoBERTime: A novel model for the detection of temporal expressions in Spanish

RoBERTime: un nuevo modelo para la detección de expresiones temporales en español

Alejandro Sánchez-de-Castro-Fernández[1], **Lourdes Araujo**[1,2], **Juan Martinez-Romo**[1,2]
[1]Universidad Nacional de Educación a Distancia (UNED), 28040, Madrid
[2]Instituto Mixto UNED-ISCIII IMIENS
{asanchez, lurdes, juaner}@lsi.uned.es

Abstract: Temporal expressions are all those words that refer to temporality. Their detection or extraction is a complex task, since it depends on the domain of the text, the language and the way they are written. Their study in Spanish and more specifically in the clinical domain is scarce, mainly due to the lack of annotated corpora. In this paper we propose the use of large language models to address the task, comparing the performance of five models of different characteristics. After a process of experimentation and fine tuning, a new model called RoBERTime is created for the detection of temporal expressions in Spanish, especially focused in the clinical domain. This model is publicly available. RoBERTime achieves state-of-the-art results in the E3C and Timebank corpora, being the first public model for the detection of temporal expressions in Spanish specialized in the clinical domain.
Keywords: Temporal expressions, TimeML, Language models, Clinical domain.

Resumen: Las expresiones temporales son todas aquellas palabras que refieran temporalidad. Su detección o extracción es una tarea compleja, ya que depende del dominio del texto, del idioma y de la forma de escritura. Su estudio en español y más específicamente en el dominio clínico es escaso, debido principalmente a la falta de corpora anotados. En este trabajo se propone el uso de grandes modelos del lenguaje para abordar la tarea, comparando el rendimiento de cinco modelos de distintas características. Tras un proceso de experimentación y fine tuning, se logra crear un nuevo modelo llamado RoBERTime para la detección de expresiones temporales en español, especialmente centrado en el dominio clínico. Este modelo se encuentra disponible de forma pública. RoBERTime alcanza resultados del estado del arte en los corpus E3C y Timebank, siendo este el primer modelo publico en detección de expresiones temporales en español especializado en el dominio clínico.
Palabras clave: Expresiones temporales, TimeML, Modelos del lenguaje, Dominio clínico.

1 *Introduction*

The detection of time expressions is a task that can be included in the field of information extraction. The extraction of these terms or expressions is necessary in more complex tasks such as: text summarization, (Ng et al., 2014), *question answering* (Pampari et al., 2018), (Sun, Cheng, and Qu, 2018) or creation of temporal lines (Leeuwenberg and Moens, 2018).

Natural language processing models need to be able to temporally locate certain events that are relevant in the text. For example, a model that works as an assistant answering questions needs to know the order of events in order to be able to answer questions like *'Did a occured before b?'* Or in the case of models that work summarizing texts (Barros et al., 2019), they need to know the temporality of events in order to be able to summarize the information in a consistent manner.

Time expressions are terms that express temporality in some form. Expressions such as: *'yesterday'*, *'at 3:00 p.m.'* o *'every eight hours'* can be considered as time expressions. To detect these expressions, two factors are taken into account, the detection of the expression and the normalization of its value.

ISSN 1135-5948 DOI 10.26342/2023-70-3

Expression detection is the same as scope detection. This can be defined as the detection of at least part of a time expression, which is composed of tokens.

Sometimes temporal expressions are easily detected, because they usually follow syntactic patterns that are easily defined under a system of rules or regular expressions. But these patterns are both language-dependent (Lange et al., 2022), (Lange et al., 2020) as well as the domain of the text in question (Strötgen et al., 2014), (Strötgen and Gertz, 2013). This forces rule-based systems to be adapted, having to adjust existing rules and in many cases adding new rules. (Skukan, Glavaš, and Šnajder, 2014), (Li et al., 2014).

The identification of time expressions can be achieved through several methods, one of them and the most popular for years has been rule-based systems. In more recent years these methods have been displaced by large language models (*LLMs*) and the Transformers architecture (Vaswani et al., 2017). These models are capable of delivering good multi-task performance on small data sets by applying a *fine-tuning* process. This process consists of adjusting the weights of the model, fitting them to a new dataset. And due to the scarcity of annotated corpora these models are a strong candidate to replace the classical systems.

This paper proposes the creation of a new model called RoBERTime based on deep learning and LLMs for the extraction of time expressions, specifically for the detection of their extension or scope, in Spanish in general and in the medical domain in particular. This model is a pioneer in the Hispanic community, since to the authors' knowledge there is no other model based on deep learning/similar characteristics for the solution of this task in Spanish. RoBERTime is the result of a process of experimentation with five LLMs of different nature, on which different fine tuning techniques have been applied in order to understand the adaptability of the LLMs to this task. Finally, the findings of each experiment have been applied to maximize the performance of RoBERTime. For the experiments and for training BioRoBERTa, the Timebank corpus and the E3C corpus have been used.

The model presented has a dual purpose. The first is to serve as part of the task of extracting temporal lines in the medical domain, which is intended to help medical professionals to more easily understand the patient's history. On the other hand, the model is intended to serve as a baseline for the extraction of time expressions in Spanish.

The rest of the article is structured as follows: Section 2 discusses the state-of-the-art and the works related to the proposal. Section 3 presents the corpora that have been used in the process. Section 4 develops in detail our proposal. The methodology and the experiments carried out are explained in Section 5. The results obtained from the experiments are analysed and compared with the current state-of-the-art. Finally, Section 7 presents the main conclusions together with the lines of future work.

2 Related Work

The current scheme regarding the annotation of time expressions is TimeML 1.2.1 (Saurı et al., 2006), also defined as an ISO standard (Pustejovsky et al., 2010), in which time expressions are defined using the TIMEX3 tag. From this point on, when TimeML is mentioned, it will refer to version 1.2.1.

TimeML defines four types of time expressions: *DATE*, *TIME*, *DURATION*, and *SET*. In this order, dates, dates with a granularity of hour or less, durations and repetitions are defined. For example, *'April 12'* would be an expression of type *DATE*, *'3:15'* would be an expression of type *TIME*, *Two months* would be an expression of type *DURATION*, and *'every 8 hours'* would be an expression of type *SET*.

Two of the best-known systems for time expression extraction are HeidelTime (Strötgen and Gertz, 2010) and TIPSem (Llorens, Saquete, and Navarro, 2010). TIPSem was designed to work in both English and Spanish. It is a system based on the use of *conditional random fields* or CRFs (Lafferty, McCallum, and Pereira, 2001) and *semantic role labeling* or SRL (Gildea and Jurafsky, 2002). HeidelTime was designed to work in English but was eventually adapted to multiple languages, including Spanish. It is a rule-based system and is perhaps the most popular, as it is still available for use today, being one of the few systems with this availability.

Other systems with similar characteristics are: ClearTK (Bethard, 2013), a system based on support vector machines (SVMs)

(Vapnik, 1999), (Cortes and Vapnik, 1995) and SUTime (Chang and Manning, 2012), a rule-based system. Both are designed to operate in English only, show similar performance to HeidelTime, and are publicly available[1][2]. Despite showing similar or even superior performance in some aspects to HeidelTime, HeidelTime has maintained its popularity over time by being adapted to multiple languages (Skukan, Glavaš, and Šnajder, 2014), (Li et al., 2014).

In Clinical TempEval (Bethard et al., 2017), a shared task held in 2017, the organizers proposed time expression extraction changing the subdomain for training and test. Specifically, they proposed to train the systems on a dataset dealing with colon cancer and test their performance on a dataset dealing with brain cancer. The results show a drop in performance of more than twenty points compared to systems trained and tested on colon cancer in detecting the scope of time expressions, thus showing the difficulty in adapting to the domain.

A system called Annotador (Navas-Loro and Rodríguez-Doncel, 2020), based on rules for English and Spanish, has recently been released, which performs better in some aspects than HeidelTime. This system is intended for use on general domain documents but is specialized in the legal domain.

Given this context, the most recent systems are based on LLMs and Transformers, as classical systems have probably reached their limit in time expression extraction and everything seems to indicate that the context understanding capability of LLMs can be applied to this task.

Different approaches can be applied to LLMs. Thus in (Almasian, Aumiller, and Gertz, 2021) two different approaches are proposed for detecting the scope of time expressions, that of *token classification* and that of *seq2seq*. In the former, the text is viewed as a sequence of tokens in which each token may or may not be part of a time expression. Time expressions can be composed of several words so it is necessary to identify which is the beginning and which is not. For example in the expression *'April 12'*, *12* will be marked as the beginning of the expression and *'April'* as part of the expression. In this way, the model can be trained to solve the token classification task. This is a relatively similar approach to the one applied with CRFs and SVMs. On the other hand, the *seq2seq* approach is a text generation problem. The model receives the raw text and has to generate the annotated text in an xml annotation format. In the example of *'April 12'* the expression would be annotated as $< TIMEX3 : "DATE" > 12 \quad of \quad April < /TIMEX3 >$. In this paper they leave aside the value normalization of expressions since applying this approach it is understood as a separate problem from that of expression extraction. This paper shows that these models are capable of outperforming rule-based systems.

In (Chen, Wang, and Karlsson, 2019) the performance of BERT is compared with that of a linear model, a *multi layer perceptron* (MLP), a *bidirectional long short term memory* (BiLSTM) and LSTM. On the one hand, BERT is trained on several datasets and its performance is measured on them. On the other hand, BERT and GloVe (Pennington, Socher, and Manning, 2014) are used as a feature extractor. These features are passed to the models mentioned before to measure their subsequent performance. The results show that retraining BERT on the datasets gives better performance in extracting the time expressions, regardless of normalization, than using BERT and GloVe as a feature extractor to train the rest of the models. Also this retrained version of BERT achieves better results in two of the three corpora used than the baseline systems: Syntime (Zhong, Sun, and Cambria, 2017), TOMN (Zhong and Cambria, 2018) and PTime (Ding et al., 2019). These three systems have superior performance to HeidelTime, ClearTK and SUTime and two of them, Syntime[3] and PTime[4] have their code publicly maintained. Despite this, these models are less popular than HeidelTime.

In (Aumiller et al., 2022) it is proposed a web service that works for the extraction of time expressions using some of the models proposed in (Almasian, Aumiller, and Gertz, 2021).

There are works such as (Almasian, Aumiller, and Gertz, 2022) in which the ELECTRA (Clark et al., 2020) architecture is used for time expression extraction in German, re-

[1] nlp.stanford.edu/software/sutime.shtml

[2] cleartk.github.io/cleartk.html

[3] github.com/xszhong/syntime

[4] ws.nju.edu.cn/ptime/

	Timebank	E3C	Total
Date	1589	241	1830
Time	155	30	185
Duration	757	552	1309
Set	49	64	113
Total	**2550**	**887**	3437
None	64541	27928	92469

Table 1: Number of tokens annotated for each type of time expression to train on the Timebank and E3C corpora.

gardless of normalization, outperforming HeidelTime.

3 Corpora

Two corpora have been used for the development of this work: TimeBank (Nieto, Saurí, and Poveda, 2011) and E3C (Magnini et al., 2020). Timebank is a corpus of annotated journalistic documents. E3C is also a multilingual corpus whose resources will only be used in Spanish. In this case, the corpus includes annotated documents from the medical domain, specifically clinical cases. Both corpora have temporal annotations following the TimeML scheme, however, E3C has an extra type of annotations called *'PRE-POSTEXP'*. These expressions are of the pre/post-operative type, such as *'postoperative'* or *'post-surgical'*. They will not be taken into account because they only appear in E3C and could hinder the detection of other types of expressions.

The sizes of both corpora can be found in Table 1. As we can see, Timebank is approximately three times larger than E3C. Two of the types of expressions are clearly in the minority, *TIME* and *SET*, so the detection of these types of expressions will be more complicated.

4 Proposal

Five models with different characteristics are explored with the goal of comparing performance based on the main characteristics of the models and creating a new model trained specifically for the task in the clinical domain. All models used are publicly available on HuggingFace. The models considered along with their distinguishing features are described below:

- RoBERTa biomedical clinical (Carrino et al., 2021): A RoBERTa-based model trained on a corpus with biomedical and clinical terminology. It is the only model of those considered that has specialized vocabulary in the clinical domain. For short, this model will be referred to as BioRoBERTa.
- BETO-uncased (Canete et al., 2020): A BERT-based model trained on a Spanish corpus for the purpose of solving a wide range of tasks in Spanish.
- BETO-NER: Model based on BETO and trained on different Spanish CONLL corpora (Sang and Buchholz, 2000), (Tjong Kim Sang, 2002), (Nivre et al., 2007) for the task of *Named Entity Recognition.* With BETO-NER we intend to study the impact of pre-training the models on the task to be worked on, comparing their performance with BETO.
- Tiny BETO-NER: A distilled version of BETO-NER was trained at the same time. A distilled or reduced model is obtained from a distillation process, whereby much of the knowledge is transferred from one model to another by reducing its size. This model has a size of approximately 13 % compared to BETO-NER, maintaining a 78 % of the performance in some tasks. The use of this model will allow studying the adaptability of very small models to other domains.
- DistilBERT-m (Sanh et al., 2019): This model is a distilled version of multilingual BERT, with a relative size of 60 % maintains 97 % of the performance. This model has the largest number of parameters of the five considered. The main feature of this model is its multilingual capability, as it will allow us to study how it adapts to the task in Spanish compared to the other models.

5 Methodology

A series of experiments will be carried out in order to compare the different models considered with each other, in addition to a series of training techniques. The experiments will be performed on a batch size of 16, learning rate of $8e^{-5}$, weight decay 0.1 and 24 epoch. This learning rate has been used and not the standard value of $2e^{-5}$ because based on experience working with these corpora and models it has been found that this value gives better results. As stated in (Mosbach, Andriushchenko, and Klakow, 2020), to improve training stability when training on small corpora it is preferable to train over a large number of epochs, until the training loss is close to 0. Therefore, the checkpoint of the model with the best f1 over the 24 training epochs will be the one shown in the experiments.

Two metrics will be used to evaluate the models *Seqeval* (Ramshaw and Marcus, 1999), (Nakayama, 2018) and *TempEval-3 toolkit*[5] (UzZaman et al., 2013). Both metrics are designed for the evaluation of *token classification* tasks, but they do not count positive and negative cases in the same way. TempEval-3 toolkit calculates the f1 metric for both fully detected (*strict*) and partially detected (*relaxed*) expressions. A strict match is given when the model predicts the full expression, whereas in a relaxed match the models predicts part of the expression. For example, if the annotated expression is 'El martes 12', and the model predicts 'El martes', it would be counted as a relaxed match. It is the most popular metric for evaluating time expression extraction, so it will be used in the final phase when comparing the performance of RoBERTime with systems from other papers. Seqeval has been used throughout the experimentation and model evaluation phase, as it was much easier to integrate than TempEval-3 toolkit. Seqeval has been used in strict mode and IOB2 scheme.

On the one hand, we are going to test which loss function offers better performance, cross entropy or focal loss (Lin et al., 2017). Both functions are very similar, with the difference that in focal loss a parameter is added to compensate for the most difficult cases to detect. For this purpose, a fine-tuning process is performed on the BioRoBERTa model. For this process we have used the E3C data with a random training/dev split, which has been maintained throughout the experiment. These two functions accept a set of weights representing the importance to be given to each type of expression, since in this case there are many more tokens that are not annotated so this imbalance must be compensated. The weights are calculated with the following formula:

$$W_{n,c} = 1 - \frac{instances_c}{\sum_{c=1}^{n} instances_c}$$

Figure 1: Where n is the total number of classes and c the class for which the weights W are to be calculated.

In order to create a model adapted to the task under consideration, the impact of training the models considered with each of the corpora (or combinations of parts of them) has been studied. In this way it is possible to study the potential of each corpus separately. Cross-folding has been applied with the maximum number of splits allowed for both corpora, two in the case of E3C and three in the case of Timebank with a fixed seed equal to 42, in order to be able to replicate the splits in each experiment. The maximum number of splits is marked by the maximum number of splits that can be made from the data without leaving any of the expression types unrepresented in the split.

To study the impact of merging the two corpora, two experiments have been performed on the BioRoBERTa model. One of them proposes a layer freezing method and the other is based on maximizing the training data of one of the corpora:

- *Join both corpora in a single fine-tuning process:* In this case, one of the two corpora is partially used for evaluation using cross-folding splits, while the other is used in its entirety as training data. In this way we seek to maximize the training set while maintaining a sufficiently representative validation set. Moreover, since the validation sets are the same as those used when training the models on each corpus separately, the results can be directly compared.
- *Train first with Timebank and retrain with E3C freezing different layers of the model:* In this way, the model is trained on the majority corpus, Timebank, performing cross-folding and choosing the best split. Subsequently, this model is trained on the minority corpus, E3C, with cross-folding while freezing different layers of the model. This layer freezing method has been studied in several works such as (Lee, Tang, and Lin, 2019), (Eberhard and Zesch, 2021) for

[5] github.com/naushadzaman/tempeval3_toolkit

language and domain adaptation.

Finally, once the methods described above have been compared, the one that maximizes the performance on E3C is selected and the rest of the models are trained with it.

After completion of the experimentation process, the best performing model is selected for publication and to compare its performance with HeidelTime and Annotador. The results of these systems on Timebank have been obtained from (Navas-Loro and Rodríguez-Doncel, 2020), while the results on E3C have been obtained using the TempEval-3 toolkit by ourselves. To obtain the HeidelTime annotations on E3C, the Philip Hausner repository was used[6] and for Annotador the María Navas repository[7].

All experiments have been performed in blocks of five iterations to minimize the random factor in model training. And to favor reproducibility each block has the same set of seeds: 42, 52, 62, 72, 82. The results shown as f1 metric have been calculated as the arithmetic mean of the five experiments.

6 *Results*

This section will summarize and analyse the results obtained in the experiments proposed in the Section 5 of this work.

6.1 Focal loss versus Cross entropy

The results of comparing the focal loss function against the cross entropy function can be seen in Table 2. Focal loss is slightly superior in three of the four types of time expressions and in the weighted average. This may be mainly due to the fact that this function gives more importance to the cases that are more difficult for the model to detect, which at the same time are usually the minority cases. This difference is noticeable in the *SET* expressions. As for the *TIME* expressions, it is possible that the difference in favor of the cross entropy function is due to the random factor in the training of the model. This difference will not be taken into account since the performance on the *TIME* expressions is too close to zero. Given these results, focal loss is chosen over cross entropy.

	Cross entropy	Focal loss
Date	0.5972	**0.6174**
Time	**0.0173**	0
Duration	0.6756	**0.678**
Set	0.191	**0.2579**
Mean	0.5966	**0.6099**

Table 2: Comparison of the F1 measure results for each class of the E3C corpus on the two loss functions used to train BioRoBERTa.

	Timebank	Mean	Split
brob	0.8029	0.79716	2
beto	0.766	0.7377	2
btn	**0.8137**	**0.7986**	2
tbtn	0.1938	0.1817	3
mbr	0.748	0.7431	3

	E3C	Mean	Split
brob	**0.5831**	**0.58045**	1
beto	0.4643	0.4482	1
btn	0.5146	0.4978	1
tbtn	0.0617	0.0511	2
mbr	0.5157	0.5057	1

Table 3: F1 measure results of each model for the best Timebank (*tb*) and E3C split, along with the average f1 of all splits. Each abbreviation corresponds, from top to bottom, with BioRoBERTa, BETO, BETO-NER, Tiny BETO-NER, and DistilBERT-m.

6.2 Performance of the models on each corpus

The f1 metric results using the Seqeval approach for each model can be found in Table 3. The results of the best split are presented, which are quite homogeneous.

As for the corpora, it can be seen that the models perform better with Timebank than with E3C. This may be mainly due to the sizes of both corpora. To support this idea, a data augmentation technique based on duplicating the records of the E3C training set while keeping the same test set of Table 3 has been tested. This resulted in improving the performance of the f1 metric by 6.44 %.

The model results show that BETO-NER is the best option for Timebank, while BioRoBERTa is the best for E3C. There are multiple factors that can explain this behaviour. On the one hand E3C is composed of documents from the clinical domain, so a mo-

[6] github.com/PhilipEHausner/python_heideltime

[7] github.com/mnavasloro/Annotador

del that has a specialized vocabulary for it should be able to provide better performance. Similarly, BETO and BETO-NER are two models trained in part with documents and newspaper articles and Timebank is built on news documents. BETO-NER also outperforms BETO, so pre-training the models on the task seems to carry relevant weight.

It can be seen that DistilBERT-m performs better in E3C than BETO and BETO-NER, while the opposite is the case with Timebank. So a model with a larger number of parameters can maintain good results in different tasks and domains. But working on a specific task or domain, a model with fewer and specialized parameters can achieve better performance if the amount of data is sufficient.

Finally, Tiny BETO-NER shows a much lower performance than the other models. Being approximately nine times smaller, Tiny BETO-NER shows approximately four times lower performance in Timebank and nine times lower performance in E3C.

In order to maximize the performance of the model on E3C, two more experiments have been performed. In the next one, the layer freezing technique is tested, to try to make the model fit better to the changes introduced by E3C to a version of the model trained on Timebank. The second one follows the strategy of maximizing the data from one of the two corpora.

6.3 Timebank + E3C with freeze layers

The results of training BioRoBERTa on Timebank and training on Timebank and E3C freezing different layers of the model can be found in Tables 4 and 5. In order to enhance reproducibility, this model is available in a HuggingFace repository.

	E3C	
brob pre E3C	0.53361	
brob post E3C	Split 1	Split 2
0 layers	0.7075	0.7234
3 layers	**0.7202**	**0.7365**
6 layers	0.7129	0.7269
9 layers	0.6936	0.6803
Mean	0.7085	**0.7164**

Table 4: F1 measure results for the BioRoBERTa model on E3C test set. The model is first trained on Timebank (brob pre E3C row), selecting the best split. Subsequently, the trained model is retrained on the other E3C splits (split 1 and split 2) and freezing different layers.

	Timebank	
brob pre E3C	0.8159	
brob post E3C	Split 1	Split 2
0 layers	0.7551	0.7508
3 layers	0.7573	0.7456
6 layers	0.7602	0.7525
9 layers	**0.7699**	**0.7587**
Mean	**0.7606**	0.7519

Table 5: F1 measure results for the BioRoBERTa model on Timebank test set. The model is first trained on Timebank (brob pre E3C row), selecting the best split. Subsequently, the trained model is retrained on the other E3C splits (split 1 and split 2) and freezing different layers.

As we can see in the first row of table 4, when training the model solely on Timebank, the performance on E3C is similar to training with E3C in isolation (see Table 3). Again, it can be seen that there is a consistent difference between the splits. As can be seen in the column *Split 2* of both tables, this split boosts the E3C results the most, while it is the one that most impairs Timebank results and vice versa. The same is true for the number of frozen layers. Timebank results are increased as more layers are forzen, while the opposite is true for E3C. This behaviour can be explained by the performance when freezing layers, since freezing more layers will cause the model to retain more information, whereas freezing few layers will cause the model to update more information. However the results in Timebank were not expected to worsen when retraining on E3C, since both corpora are ultimately composed of time expressions of the same type and in the same language. Therefore, the model seems to present difficulties in generalizing to both domains, giving a trade-off situation between the two corpora. This is also evident when the model is trained with Timebank alone. In Table 5 BioRoBERTa achieves an f1 metric of 0.8159 on the Timebank test set while if trained first with Timebank and subse-

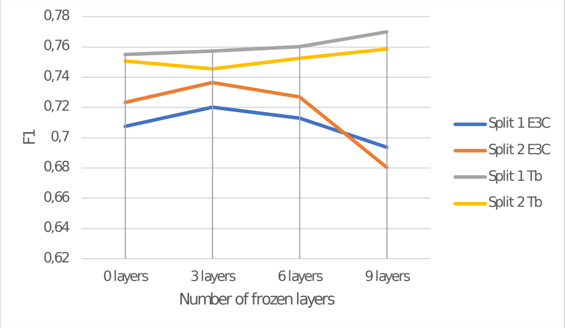

Figure 2: Evolution of the f1 metric according to the number of frozen layers on TimeBank and E3C. Each split corresponds to both E3C evaluation splits. Increasing the number of frozen layers increases TimeBank´s performance, while decreasing the E3C and vice versa.

quently with E3C with freezing, the average performance drops by 0.0553 points for split 1 and 0.064 for split 2. With E3C the opposite happens, the average performance improves by 0.1749 for split 1 and 0.1828 points for split 2.

The difference in performance can be clearly noticed depending on the number of layers that are frozen.

One would expect that the best performance over E3C would be achieved by not freezing any layers, since the model should completely adapt to the new data set. However, this does not occur and the best performance on E3C is given by freezing three layers, even freezing six layers improves performance over not freezing any in both splits. This behaviour may be due to the information shared between the two corpora. It may be the case that when training without any frozen layer the model loses relevant information acquired from Timebank stored in the initial layers. The trade-off is shown in Figure 2. It can be clearly seen how the performance of E3C decreases as more layers are frozen, while for Timebank it increases.

6.4 Timebank + E3C complete versus E3C + Timebank complete

Tables 6 and 7 shows the results of training the BioRoBERTa model on both corpora, using only one of them to perform the training and validation split.

From the mean values it can be seen that maximizing the Timebank set for training does not bring as much benefit as maximizing the E3C set does. When the entire E3C set is used for training, Table 6, there is a 3 % performance improvement over E3C and a 1 % drop in Timebank as compared to using the entire Timebank for training, Table 7. This may be because since E3C is a corpus with more complex cases for the model, if its representation is maximized in training, the model extrapolates those cases better to Timebank. On the contrary, if Timebank representation is maximized, the model is not able to use those extra cases to extrapolate to E3C.

Comparing these results with those of Tables 4 and 5 it can be seen that maximizing the amount of data for training is better than freezing layers of the model and training separately. The results on both corpora are, on average, better when this strategy is taken. Given these results and since the objective is to maximize the performance of the models for the clinical domain, we will choose the option of training the rest of the models using full E3C for training and Timebank to perform cross-folding splits for the evaluation set.

			brob
Timebank split	Timebank	split 1	0.8036
		split 2	0.8028
		split 3	0.7922
		Mean	**0.7995**
	E3C	split 1	0.7406
		split 2	0.7201
		split 3	0.7249
		Mean	**0.7285**

Table 6: F1 measure results for BioRoBERTa model on both test sets, trained on: Timebank (without the validation split), and the whole E3C training sets.

			brob
E3C split	Timebank	split 1	0.8127
		split 2	0.8031
		Mean	**0.8079**
	E3C	split 1	0.7076
		split 2	0.7081
		Mean	**0.7079**

Table 7: F1 measure results for BioRoBERTa model on both test sets, trained on: E3C (without the validation split), and the whole Timebank training sets.

Tb	split 1	split 2	split 3	Mean
brob	0.8036	0.8028	0.7922	0.7995
beto	0.7384	0.7559	0.7209	0.7383
btn	0.8069	0.8233	0.798	**0.8093**
tbtn	0.2676	0.2612	0.2823	0.2702
mbr	0.7533	0.7604	0.7495	0.7544

E3C	split 1	split 2	split 3	Mean
brob	0.7406	0.7201	0.7249	**0.7285**
bt	0.6349	0.6242	0.6099	0.6229
btn	0.7123	0.6768	0.6987	0.6958
tbtn	0.2024	0.2246	0.2185	0,215
mbr	0.6683	0.6367	0.6629	0.6558

Table 8: Comparison of the different models considered for each Timebank split (*Tb*). The models were trained on Timebank split into validation and training, concatenating to the latter the full E3C training set.

6.5 Final results

The performance of all models on both corpora can be seen in Table 8. BETO-NER manages to have the best performance on Timebank, outperforming BioRoBERTa by 1,2 % on average. On the other hand, BioRoBERTa achieves better results over E3C, being superior by 4,7 %. Significant differences again stand out between BETO and BETO-NER, the latter being better by 11 % over E3C and 8 % over Timebank. DistilBERT-m, on the other hand, performs better than expected. Being a multi-language model and without any specialization in either task or domains, lower performance was expected. Tiny BETO-NER shows similar performance to those seen above. The performance differences based on splits again show the impact of finding a good split for the training data.

Given these results, BioRoBERTa has been chosen as the best model for this task. Because it is the model with the best performance on E3C and with a performance very close to BETO-NER on Timebank. As a result of the whole experimentation process, a model has been trained on the best options found. This new model based on BioRoBERTa, which we have called RoBERTime, is available at[8].

Table 9 shows the comparison of the best version of RoBERTime with HeidelTime and Annotador. It can be seen how the rule-based systems outperform RoBERTime over Timebank. This is mainly due to the fact that these systems were created with the purpose of giving good results on this corpus. This behaviour is shown by measuring the performance of HeidelTime and Annotator over E3C. As can be seen, the performance is considerably reduced with RoBERTime standing out above both.

Timebank	Strict	Relaxed	Type
RoBERTime	0.8152	0.8798	0.8504
Heideltime	**0.8533**	0.8907	0.8363
Annotador	0.8513	**0.9179**	**0.8923**

E3C	Strict	Relaxed	Type
RoBERTime	**0.7606**	**0.9108**	**0.8357**
Heideltime	0.5945	0.7558	0.6083
Annotador	0.6006	0.7347	0.5598

Table 9: Comparison on the f1 metric of TempEval-3 toolkit.

Regarding the performance of RoBERTime over Timebank, it can be seen how this model outperforms HeidelTime in detecting the time expression type. RoBERTime also shows good performance in detecting the type of expression in E3C. RoBERTime excels in this section over HeidelTime and Annotator.

About E3C, the big difference between strict detection and relaxed detection stands out, being the difference between both much bigger than in Timebank. This may be due to the fact that the expressions in E3C are composed of more tokens or are formulated in more varied ways than in Timebank, being more difficult to detect completely.

RoBERTime fails to positively detect some expressions such as *“actualmente”*, *“recientemente”*, four-digit numbers that do not correspond to dates such as *“2006”* or ages such as *”6 años”*. These expressions are particularly difficult to detect since sometimes it is necessary to take into account a large part of the text. There are also other expressions that are not annotated in E3C such as *“Ácido clavulánico 125 mgr, 1 comp. / 8 horas”* or *“isoniazid 300 mgr al día”* that RoBERTime detects.

In Timebank some cases have been detected in which the corpus has annotated age expressions as time expressions, as in the document with identifier *11033_20000817*: *“27 años”*, *“35 años”*, *“53 años”*. It is therefore possible that the model has at least partially acquired this behaviour from the annotations. There are other cases of ambiguities

[8] huggingface.co/asdc/Bio-RoBERTime

in the Timebank annotations that may confuse the model, such as annotating the expression *"hace un año"* in *"Al menos, hace un año, los camiones circulaban en la misma dirección"*, but not doing so in *"Hace un año los camiones te adelantaban a 70 o 80 kilómetros por hora"*.

These ambiguities are hard to treat, because they imply to make changes to the original annotations, and there might be some reason on why those expressions are or are not annotated. So it has to be accepted that this results are limitated by the annotators accuracy in both corpora.

7 Conclusions and future work

In this work we have presented a new model called RoBERTime which achieves state-of-the-art results in the detection of time expressions in Spanish in the journalistic domain with the Timebank corpus and in the clinical domain with E3C. In particular, E3C outperforms some of the most popular systems in all aspects. It has been proved that, unlike other systems, RoBERTime is able to adapt to both domains, showing a balanced behaviour on both corpora. This shows the great potential of LLMs to solve the task of time expression extraction. All this has been achieved through a series of experiments, which have allowed us to make decisions based on empirical results to maximize the performance of RoBERTime.

It has been observed in the performance of BETO-NER and BioRoBERTa that pretraining the models on the task and domain can considerably improve performance. The ability of LLMs to adapt to different tasks and domains has also been shown to be easier to shape than classical rule-based systems.

Although the main objective of the work was to achieve a time expression detection model for the clinical domain, the proposed model has turn out to perform better outside the clinical domain, in spite of the performance in E3C has been prioritized over that of Timebank. This may be due to the quality and quantity of data.

As for future work, the possibility of exploring the multilingualism of the corpora is being considered. On the one hand, this would make it possible to create multilingual models and, on the other hand, to translate annotations from other languages into Spanish, in order to increase the size of the available data. We are also considering exploring data augmentation techniques, since it has been observed that doubling the E3C records improves performance. We also consider the task of obtaining the normalized value of the expressions detected by RoBERTime, exploring rule-based system solutions and LLM-based solutions. For example, seq2seq models in which the expression would be taken as the input and the normalized value as the output. In line with this, the use of this model is proposed for the extraction of time lines, a task that requires the extraction of time expressions.

Acknowledgments

This work has been funded by the following projects DOTT-HEALTH (MCI/AEI/FEDER, UE with identification PID2019-106942RB-C32), OBSERMENH(MCIN/AEI/10.13039/501100011033 and NextGenerationEU"/PRTR with identification TED2021-130398B-C21) and by the project RAICES (IMIENS 2022).

References

Almasian, S., D. Aumiller, and M. Gertz. 2021. Bert got a date: Introducing transformers to temporal tagging. *arXiv preprint arXiv:2109.14927*.

Almasian, S., D. Aumiller, and M. Gertz. 2022. Time for some german? pre-training a transformer-based temporal tagger for german. In *Text2Story@ ECIR*, pages 83–90.

Aumiller, D., S. Almasian, D. Pohl, and M. Gertz. 2022. Online dateing: A web interface for temporal annotations. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3289–3294.

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

Bethard, S. 2013. Cleartk-timeml: A minimalist approach to tempeval 2013. In *Second joint conference on lexical and computational semantics (* SEM), volume 2: proceedings of the seventh interna-*

tional workshop on semantic evaluation (SemEval 2013), pages 10–14.

Bethard, S., G. Savova, M. Palmer, and J. Pustejovsky. 2017. SemEval-2017 task 12: Clinical TempEval. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 565–572, Vancouver, Canada, August. Association for Computational Linguistics.

Canete, J., G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez. 2020. Spanish pre-trained bert model and evaluation data. *Pml4dc at iclr*, 2020:1–10.

Carrino, C. P., J. Armengol-Estapé, A. Gutiérrez-Fandiño, J. Llop-Palao, M. Pàmies, A. Gonzalez-Agirre, and M. Villegas. 2021. Biomedical and clinical language models for spanish: On the benefits of domain-specific pretraining in a mid-resource scenario. *arXiv preprint arXiv:2109.03570*.

Chang, A. X. and C. D. Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In *Lrec*, volume 3735, page 3740.

Chen, S., G. Wang, and B. Karlsson. 2019. Exploring word representations on time expression recognition. Technical report, Technical report, Microsoft Research Asia.

Clark, K., M.-T. Luong, Q. V. Le, and C. D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Cortes, C. and V. Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Ding, W., G. Gao, L. Shi, and Y. Qu. 2019. A pattern-based approach to recognizing time expressions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6335–6342, Jul.

Eberhard, O. and T. Zesch. 2021. Effects of layer freezing on transferring a speech recognition system to under-resourced languages. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*, pages 208–212.

Gildea, D. and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Lafferty, J. D., A. McCallum, and F. C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lange, L., A. Iurshina, H. Adel, and J. Strötgen. 2020. Adversarial alignment of multilingual models for extracting temporal expressions from text. *arXiv preprint arXiv:2005.09392*.

Lange, L., J. Strötgen, H. Adel, and D. Klakow. 2022. Multilingual normalization of temporal expressions with masked language models. *arXiv preprint arXiv:2205.10399*.

Lee, J., R. Tang, and J. Lin. 2019. What would elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*.

Leeuwenberg, A. and M.-F. Moens. 2018. Temporal information extraction by predicting relative time-lines. *arXiv preprint arXiv:1808.09401*.

Li, H., J. Strötgen, J. Zell, and M. Gertz. 2014. Chinese temporal tagging with heideltime. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 133–137.

Lin, T.-Y., P. Goyal, R. Girshick, K. He, and P. Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Llorens, H., E. Saquete, and B. Navarro. 2010. Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 284–291.

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

Mosbach, M., M. Andriushchenko, and D. Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines.

Nakayama, H. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Navas-Loro, M. and V. Rodríguez-Doncel. 2020. Annotador: a temporal tagger for spanish. *Journal of Intelligent & Fuzzy Systems*, 39(2):1979–1991.

Ng, J. P., Y. Chen, M.-Y. Kan, and Z. Li. 2014. Exploiting timelines to enhance multi-document summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 923–933.

Nieto, M. G., R. Saurí, and M. A. B. Poveda. 2011. Modes timebank: a modern spanish timebank corpus. *Procesamiento del lenguaje natural*, 47:259–267.

Nivre, J., J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 915–932.

Pampari, A., P. Raghavan, J. Liang, and J. Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. *arXiv preprint arXiv:1809.00732.*

Pennington, J., R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Pustejovsky, J., K. Lee, H. Bunt, and L. Romary. 2010. Iso-timeml: An international standard for semantic annotation. In *LREC*, volume 10, pages 394–397.

Ramshaw, L. A. and M. P. Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*. Springer, pages 157–176.

Sang, E. F. and S. Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. *arXiv preprint cs/0009008.*

Sanh, V., L. Debut, J. Chaumond, and T. Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Saurı, R., J. Littman, B. Knippen, R. Gaizauskas, A. Setzer, and J. Pustejovsky. 2006. Timeml annotation guidelines version 1.2. 1.

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

Strötgen, J., T. Bögel, J. Zell, A. Armiti, T. V. Canh, and M. Gertz. 2014. Extending HeidelTime for temporal expressions referring to historic dates. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2390–2397, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

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

Strötgen, J. and M. Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

Sun, Y., G. Cheng, and Y. Qu. 2018. Reading comprehension with graph-based temporal-casual reasoning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 806–817.

Tjong Kim Sang, E. F. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002).*

UzZaman, N., H. Llorens, L. Derczynski, J. Allen, M. Verhagen, and J. Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings* *of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9.

Vapnik, V. 1999. *The nature of statistical learning theory*. Springer science & business media.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zhong, X. and E. Cambria. 2018. Time expression recognition using a constituent-based tagging scheme. In *Proceedings of the 2018 world wide web conference*, pages 983–992.

Zhong, X., A. Sun, and E. Cambria. 2017. Time expression analysis and recognition using syntactic token types and general heuristic rules. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 420–429.