XLTime: A Cross-Lingual Knowledge Transfer Framework for Zero-Shot Low-Resource Language Temporal Expression Extraction

Anonymous ACL submission

Abstract

Temporal Expression Extraction (TEE) is essential for understanding time in natural language. It has applications in Natural Language Processing (NLP) tasks such as question answering, information retrieval, and causal inference. To date, work in this area has mostly focused on English as TEE for low-resource languages is hindered by a scarcity of training data. We propose XLTime, a novel framework for zero-shot low-resource language TEE. XLTime works on top of pre-trained language models and leverages multi-task learning to prompt cross-language knowledge transfer both from English and within the low-resource languages. It alleviates the problems caused by the shortage in low-resource language training data. We apply XLTime with different language models and show that it outperforms the previous automatic SOTA methods on four low-resource languages, i.e., French, Spanish, Portuguese, and Basque, by large margins. It also closes the gap considerably on the handcrafted HeidelTime tool.

1 Introduction

Temporal Expression Extraction (TEE) refers to the detection of *temporal expressions* (such as dates, durations, etc. as shown in Table 1). It is an important NLP task and has downstream applications in question answering (Choi et al., 2018), information retrieval (Mitra et al., 2018), and causal inference (Feder et al., 2021). Most TEE methods work on English and are rule-based (Strötgen and Gertz, 2013; Zhong et al., 2017). Deep learning-based methods (Chen et al., 2019; Lange et al., 2020) are less common and report results on par with or inferior to the rule-based SOTAs.

Moreover, methods that work on low-resource languages are rare, because of the scarcity of annotated data. We find that that there is considerable room for improving TEE, especially for low-resource languages (e.g., the previous SOTA performance on the English TE3 dataset (UzZaman et al., 2013) is around 0.90 in F1, while that on the Basque TEE benchmark (Altuna et al., 2016) is merely 0.47). Recent deep learning methods, which have shown gains for many tasks, are under-explored for this important area of NLP.

Table 1: Temporal expressions of different types (See Appendix A for the definitions of the types).

In <u>*the last three months*</u> (Duration), net revenue rose 4.3% to $525.8 million from $504.2 million <u>*last year*</u> (Date). The official news agency, which gives the <u>*daily*</u> (Set) tally of inspections, updated on <u>*Friday evening*</u> (Time).

Developing an approach that can learn from a limited amount of training data is crucial for this field because of the efforts required to develop high-quality rules for any language. Thus we propose a cross-lingual knowledge transfer framework for zero-shot low-resource language TEE, namely, XLTime. We base our framework on pre-trained multilingual models (Devlin et al., 2019; Conneau et al., 2020). We then use Multi-Task Learning (MTL) (Liu et al., 2019a) to prompt knowledge transfer both from English and within the low-resource languages. We design primary and secondary tasks. The former leverages the existing data of the other languages. It transfers *explicit knowledge* that explicitly tells the forms of the temporal expressions in a *source language*. The latter constructs its training data in a self-supervised (Liu et al., 2021) manner. It transfers *implicit knowledge* by teaching the model to tell if a sentence in the *target language* contains temporal expressions.

Contributions. 1) We propose XLTime, which prompts cross-lingual knowledge transfer using MTL to address low-resource language TEE. **2)** We show that XLTime outperforms the previous automatic SOTA methods by large margins on four low-resource languages, i.e., French, Spanish, Portuguese, and Basque, in a zero-shot setting. **3)** We show that XLTime also approaches the performance of the heavily handcrafted HeidelTime (Strötgen and Gertz, 2013), and even beats it on two languages (Portuguese and Basque). We make our code and data publicly available [1].

2 Related Work

While TEE is an important problem in NLP, there is relatively little work in the area, and most of this work focuses on English. Prior art can be divided into two classes: rule/pattern-based and deep learning approaches. In the first class, HeidelTime (Strötgen and Gertz, 2013) is the most commonly used tool and is the top approach to date, even though it is a collection of finely-tuned rules. It covers over a dozen languages. The approach was later extended to more languages with HeidelTime-auto (Strötgen and Gertz, 2015), which leverages language-independent processing and rules. Other approaches include SynTime (Zhong et al., 2017), which is based on heuristic rules, and SUTIME (Chang and Manning, 2012) and PTime (Ding et al., 2019), which leverages pattern learning.

For the second class, Laparra et al. (2018) proposes a model based on RNNs. Chen et al. (2019) uses BERT with a linear classifier. Lange et al. (2020) inputs mBERT embeddings to a BiLSTM with a CRF layer and outperforms HeidelTime-auto on four languages. However, the reported performances of the deep learning-based methods are inferior to the rule-based ones, which is, in part, due to the complexity of the problem and training data paucity. In our work, we propose a new model which outperforms prior deep learning methods but also closes the gap considerably on HeidelTime.

3 Proposed Method

We formalize TEE as a sequence labeling task, similar to named entity recognition (NER) (Lample et al., 2016). Figure 1 shows the architecture of XLTime.

3.1 Pre-trained Multilingual Backbone

We adopt SOTA multilingual models (Devlin et al., 2019; Conneau et al., 2020) as the backbone of XLTime, denoted as: $T(E(X))$. X is the input sequence. E and T are the lexicon and Transformer encoder layers as shown in Figure 1(b). The backbone allows XLTime to acquire semantic and syntactic knowledge of various languages. It is shared by the MTL tasks introduced in Section 3.2.

[1]Github to be added.

3.2 MTL-based Cross-Lingual Knowledge Transfer

XLTime transfers knowledge from multiple *source languages* to the low-resource *target language*. The source languages include English and other languages for which TEE training data is available. We design *primary* and *secondary* tasks on top of the backbone to prompt *explicit* and *implicit* knowledge transfer. The primary task transfers knowledge that explicitly encodes the forms of the temporal expressions in a source language. It is formalized as sequence labeling and directly leverages the training data of the source language to train the backbone along with the primary task head, shown in Figure 1 (b). The primary task minimizes $\mathcal{L}_{sl}$:

$$\mathcal{L}_{sl} = -\sum_{i=1}^{b}\sum_{j=1}^{m_i} \mathbb{1}(y_{ij}, c) log(softmax(\mathbf{W}\cdot\mathbf{x})), \quad (1)$$

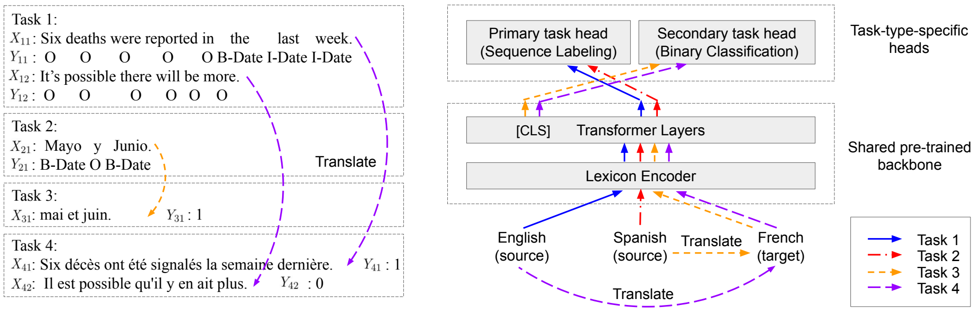
where $\mathbf{x} \in \mathbb{R}^d$ is the embedding of a token output by the backbone. $\mathbf{W} \in \mathbb{R}^{|c|\times d}$ is the primary task head. c and y_{ij} are the predicted and ground-truth labels of the token. b is the total number of input sequences and m_i is the length of the ith sequence.

The secondary task implicitly reveals how the temporal expressions would be expressed in the target language. We translate the sequences in the source language training data into the target language using Google Translate[2] (we also experiment with AWS Translate[3] and observe similar results). The secondary task is formalized as binary classification, where the input samples are the translated sequences and the labels are indicators of whether or not the original sequences contain temporal expressions (can be easily inferred from the original labels). This task tunes the model to learn the characteristics of temporal expressions in the target language in an implicit manner. It is self-supervised and requires no token-level labeling. It trains the backbone along with the secondary task head and minimizes $\mathcal{L}_{bc}$:

$$\mathcal{L}_{bc} = -\sum_{i=1}^{b} \mathbb{1}(y'_i, c') log(softmax(\mathbf{W}'\cdot\mathbf{x}')), \quad (2)$$

where $\mathbf{x}' \in \mathbb{R}^d$ is the sequence embedding output by the [CLS] of the backbone. $\mathbf{W}' \in \mathbb{R}^{2\times d}$ is the secondary task head. c' and y'_i are the predicted and true sequence labels. We train XLTime concurrently on the primary and secondary tasks, further explanation is in Appendix B.

[2]https://translate.google.com/
[3]https://aws.amazon.com/translate/



(a) Sample Input and label

(b) Architecture of the XLTime Framework

Figure 1: The architecture and sample training input of the proposed XLTime framework (*best viewed in color*).

Table 2: Dataset statistics (more details in Appendix C).

Lang	Dataset	# Exprs
FR	Bittar et al. (2011)	425
ES	UzZaman et al. (2013)	1,094
PT	Costa and Branco (2012)	1,227
EU	Altuna et al. (2016)	847
EN	TE3 (UzZaman et al., 2013)	1,830
	Wikiwars (Mazur and Dale, 2010)	2,634
	Tweets (Zhong et al., 2017)	1,128

An Illustrative Example. In Figure 1, Tasks 1 and 4 transfer knowledge from *English* to *French*. Task 1 (primary) transfers knowledge about the exact forms of English temporal expressions using token-level labels (Y_{11} and Y_{12}). Task 4 (secondary) takes the French translations (X_{41} and X_{42}) of X_{11} and X_{12} as input and let Y_{41} and Y_{42} indicate whether or not the original sequences contain temporal expressions (can be inferred from Y_{11} and Y_{12}). Task 4 provides indirect knowledge about French temporal expressions. Similarly, Tasks 2 and 3 transfer from *Spanish* to *French*.

4 Experiments

4.1 Experimental Setup

Datasets. We use the English (EN), French (FR), Spanish (ES), Portuguese (PT), and Basque (EU) TEE benchmark datasets. Table 2 shows dataset statistics (see Appendix C for a more detailed description). For each target language, we split its dataset with 10% for validation and 90% for test. For each source language (applicable to XLTime), we use the whole dataset for training.

Baselines. We evaluate against rule-based as well as deep learning-based methods. We compare to the handcrafted HeidelTime (Strötgen and Gertz, 2013) and its automatically extended version, HeidelTime-auto (Strötgen and Gertz, 2015). We also compare to deep learning methods: BiLSTM+CRF (Lange et al., 2020), mBERT, base and large versions of XLMR (trained on English TEE datasets and evaluated on low-resource languages).

Our Approaches. We test out several variants of our proposed model, which can be broken into two classes: 1) Cross-lingual transfer from EN. We apply XLTime on mBERT, base and large versions of XLMR and use EN as the only source language. 2) Cross-lingual transfer from EN and others. We transfer from other languages in addition to EN. Experimental settings are found in Appendix D.

Evaluation Metrics. We report F1, precision, and recall in *strict match* (UzZaman et al., 2013), i.e., all its tokens must be correctly recognized for an expression to be counted as correctly extracted. We follow the setting in prior work of evaluating "without type" and report the results without considering the types of the temporal expressions (e.g., for 'see you tomorrow', a prediction such as 'O O B-Duration' would be counted as correct, though the proper labeling would be 'O O B-Date').

We do note that the temporal expression field should ultimately evaluate on the more complex task of identifying temporal expressions as well as their types. This is in the spirit of the annotations and is in line with other sequence labeling tasks, such as NER. Therefore, we also experiment with the "with type" setting and show results in Appendix F. In both settings, the observations made in Section 4.2 hold and XLTime outperforms the previous automatic SOTAs by large margins.

4.2 Evaluation Results

We evaluate XLTime on zero-shot low-resource language TEE (see Table 3). We observe: **1)** XLTime-XLMRlarge outperforms the strongest automatic baseline by up to 13%, 14%, and 18% in F1, precision, and recall on all languages. It even out-

Table 3: Zero-shot low-resource language TEE results (w/o type).

Model	FR F1	FR Pr.	FR Re.	ES F1	ES Pr.	ES Re.	PT F1	PT Pr.	PT Re.	EU F1	EU Pr.	EU Re.
`Automatic Baseline Models`												
HeidelTime-auto	0.55	0.65	0.47	0.42	0.58	0.33	0.50	0.67	0.39	0.17	0.66	0.10
BiLSTM+CRF(temp)	0.64	0.73	0.57	0.62	0.68	0.56	0.64	0.66	0.63	0.47	0.58	0.40
mBERT	0.63	0.70	0.58	0.62	0.69	0.56	0.66	0.63	0.69	0.65	0.71	0.60
XLMR-base	0.69	0.75	0.64	0.54	0.61	0.48	0.63	0.64	0.62	0.46	0.64	0.36
XLMR-large	0.75	0.78	0.73	0.72	0.75	0.69	0.75	0.74	0.76	0.70	0.74	0.67
`Cross-Lingual Transfer from EN (Ours)`												
XLTime-mBERT	0.73	0.73	0.72	0.71	0.77	0.66	0.67	0.64	0.71	0.76	0.81	0.71
XLTime-XLMRbase	0.78	*0.79*	0.78	0.66	0.70	0.63	0.68	0.67	0.70	0.71	0.76	0.66
XLTime-XLMRlarge	0.76	*0.79*	0.73	0.72	**0.79**	0.67	0.77	0.74	0.81	*0.78*	*0.85*	0.71
`Cross-Lingual Transfer from EN and Additional Source Languages (Ours)`												
XLTime-mBERT	0.80	0.77	*0.82*	**0.77**	**0.79**	**0.74**	*0.80*	*0.77*	*0.83*	0.77	0.82	0.72
XLTime-XLMRbase	*0.82*	*0.79*	**0.86**	0.72	*0.78*	0.68	0.73	0.72	0.75	**0.79**	**0.86**	*0.73*
XLTime-XLMRlarge	**0.84**	**0.82**	**0.86**	*0.75*	**0.79**	*0.71*	**0.84**	**0.82**	**0.87**	**0.79**	0.84	**0.74**
`Handcrafted Method`												
HeidelTime	0.86	0.87	0.85	0.86	0.91	0.81	0.60	0.64	0.57	/	/	/

Table 4: Zero-shot low-resource language TEE with additional source languages (F1 scores w/o type). The blue cells are expected to, while the underlined cells actually outperform (by $\geq 4\%$) using EN as the only source language.

Target Language	FR				ES			
Source Language(s)	EN	EN, EU	EN, PT	EN, ES	EN	EN, EU	EN, PT	EN, FR
XLTime-mBERT	0.73	0.76	0.72	0.80	0.71	0.72	0.72	0.77
XLTime-XLMRbase	0.78	0.76	0.78	0.82	0.66	0.68	0.71	0.72
XLTime-XLMRlarge	0.76	0.81	0.80	0.84	0.72	0.72	0.75	0.73
Target Language	**PT**				**EU**			
Source Language(s)	**EN**	**EN, FR**	**EN, ES**	**EN, EU**	**EN**	**EN, PT**	**EN, ES**	**EN, FR**
XLTime-mBERT	0.67	0.80	0.70	0.80	0.76	0.73	0.75	0.77
XLTime-XLMRbase	0.68	0.73	0.63	0.56	0.71	0.74	0.75	0.79
XLTime-XLMRlarge	0.77	0.82	0.84	0.74	0.78	0.79	0.79	0.77

performs the handcrafted HeidelTime method by a large margin (24% in F1) in PT. **2)** Applying XLTime improves upon the vanilla language models, even by transferring knowledge only from EN. E.g., XLTime-XLMRbase outperforms XLMR-base by 13%, 22%, 8%, and 54% in F1 on FR, ES, PT, and EU. **3)** Introducing additional source languages to XLTime further improves the performance: the F1 improves by up to 19%, 11%, and 11% for XLTime-mBERT, XLTime-XLMRbase, and XLTime-XLMRlarge. **4)** HeidelTime is a very hard baseline to beat given the time and care that went into developing language-specific rules. However, XLTime approaches its performance for FR and ES, outperforms it for PT, and makes predictions for EU (where HeidelTime has no rules).

We also study the effect of transferring additional knowledge from low-resource language(s), see Table 4 and Appendix E. Our assumption is, similar languages (FR, ES, and PT) would help each other (one exception is PT, as its dataset is translated from the EN dataset and we, therefore, don't expect it to provide a benefit beyond what EN already provides). We observe: **1)** In most cases, transferring additional knowledge from similar languages does help (the blue cells overlap with the underlined cells), and improves the F1 by up to 13%. **2)** In some rare cases, negative transfer (Wu et al., 2020) occurs as adding source languages hurts performance (e.g., EN, ES $\rightarrow$ PT scores lower than EN $\rightarrow$ PT for XLTime-XLMRbase). We hypothesize this is related to the quality of the datasets and plan to address this in the future (Appendix H).

5 Conclusion

We propose XLTime for zero-shot low-resource language TEE. XLTime is based on language models and leverages MTL to prompt cross-language knowledge transfer. It greatly alleviates the problems caused by the shortage in low-resource language data and shows results superior to the previous automatic SOTA methods on four languages. In addition, it approaches the performance of a highly engineered rule-based system.

References

Begoña Altuna, María Jesús Aranzabe, and Arantza Díaz de Ilarraza. 2016. Adapting timeml to basque: Event annotation. In *Proceedings of CICLing 2016*, pages 565–577. Springer.

André Bittar, Pascal Amsili, Pascal Denis, and Laurence Danlos. 2011. French timebank: an iso-timeml annotated reference corpus. In *Proceedings of ACL-HLT 2011*, pages 130–134.

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

Sanxing Chen, Guoxin Wang, and Börje Karlsson. 2019. Exploring word representations on time expression recognition. Technical report, Tech. rep., Microsoft Research Asia.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of EMNLP 2021*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL 2020*, pages 8440–8451.

Francisco Costa and António Branco. 2012. Timebankpt: A timeml annotated corpus of portuguese. In *LREC*, volume 12, pages 3727–3734.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186.

Wentao Ding, Guanji Gao, Linfeng Shi, and Yuzhong Qu. 2019. A pattern-based approach to recognizing time expressions. In *Proceedings of AAAI 2019*, volume 33, pages 6335–6342.

Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2021. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT 2016*, pages 260–270.

Lukas Lange, Anastasiia Iurshina, Heike Adel, and Jannik Strötgen. 2020. Adversarial alignment of multilingual models for extracting temporal expressions from text. In *Proceedings of Workshop on Representation Learning for NLP at ACL 2020*, pages 103–109.

Egoitz Laparra, Dongfang Xu, and Steven Bethard. 2018. From characters to time intervals: New paradigms for evaluation and neural parsing of time normalizations. *Transactions of the Association for Computational Linguistics*, 6:343–356.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. In *Proceedings of ACL 2019*, pages 4487–4496.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR 2019*.

Pawel Mazur and Robert Dale. 2010. Wikiwars: A new corpus for research on temporal expressions. In *Proceedings of EMNLP 2010*, pages 913–922.

Bhaskar Mitra, Nick Craswell, et al. 2018. *An introduction to neural information retrieval*. Now Foundations and Trends.

James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. Iso-timeml: An international standard for semantic annotation. In *LREC*, volume 10, pages 394–397.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

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

Jannik Strötgen and Michael Gertz. 2015. A baseline temporal tagger for all languages. In *Proceedings of EMNLP 2015*, pages 541–547.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics, Volume 2: Proceedings of SemEval 2013*, pages 1–9.

Sen Wu, Hongyang R Zhang, and Christopher Ré. 2020. Understanding and improving information transfer in multi-task learning. In *Proceedings of ICLR 2020*.

Xiaoshi Zhong, Aixin Sun, and Erik Cambria. 2017. Time expression analysis and recognition using syntactic token types and general heuristic rules. In *Proceedings of ACL 2017*, pages 420–429.

Algorithm 1: Training XLTime

```
//Initialize model.
Load the parameters of E and T from a
    pre-trained multilingual model.
Initialize W and W' randomly.
// Prepare task data.
for t in {primary, secondary} do
    Split the data of task t into
      mini-batches B_t
B = B_primary ∪ B_secondary
for e in 1, ..., epoch do
    Randomly shuffle B
    //b_t is a mini-batch of task t
    for b_t in B do
        if t is a primary task then
            L_sl = Equation 1
        else
            L_bc = Equation 2
        Compute gradient and update
          model parameters
```

A Types of the Temporal Expressions

According to ISO-TimeML (Pustejovsky et al., 2010), the TEE dataset annotation guideline, there are four types of temporal expressions, i.e., *Date*, *Time*, *Duration*, and *Set*. *Date* refers to a calendar date, generally of a day or a larger temporal unit; *Time* refers to a time of the day and the granularity of which is smaller than a day; *Duration* refers to the expressions that explicitly describe some period of time; *Set* refers to a set of regularly recurring times (Pustejovsky et al., 2010).

B The Training Procedure

We adopt mini-batch-based stochastic gradient descent (SGD) to train XLTime, as shown in Algorithm 1. To concurrently train on the primary and secondary tasks, we split the training data of both tasks into mini-batches and randomly take one at each step. We then calculate loss using that mini-batch and update the parameters of the shared backbone (including E and T) as well as the task-type-specific head. The head of the other task type is unaffected.

C Detailed Statistics of the Datasets

Table 5 shows the detailed statistics of the datasets used in this study.

Table 5: The statistics of the datasets.

Lang	Dataset	Domain	#Docs	#Exprs	#Dates	#Times	#Durations	#Sets
FR	Bittar et al. (2011)	News	108	425	227	130	52	16
ES	UzZaman et al. (2013)	News	175	1,094	749	57	251	37
PT	Costa and Branco (2012)	News	182	1,227	998	41	176	12
EU	Altuna et al. (2016)	News	91	847	662	22	151	12
	TE3 (UzZaman et al., 2013)	News	276	1,830	1,471	34	291	34
EN	Wikiwars (Mazur and Dale, 2010)	Narrative	22	2,634	2,634	0	0	0
	Tweets (Zhong et al., 2017)	Utterance	942	1,128	717	173	200	38

D Experimental Setting

We set d, the embedding dimension, to be 768 when applying on the base version language models and 1024 on large versions. We use AdamW (Loshchilov and Hutter, 2019) with a learning rate of $7e^{-6}$ and warm-up proportion of 0.1. We train the models for 50 epochs and use the best model as indicated by the validation set for prediction. All datasets are transformed into IOB2 format to fit the sequence labeling setting. For BiLSTM+CRF, we use the hyperparameters as suggested in the original paper. We repeat all experiments for 5 times and report the mean results.

E Full Table for Zero-shot Low-resource Language TEE with Additional Source Languages

Table 6 shows the precision and recall of zero-shot low-resource language TEE with additional source languages (w/o type).

F Zero-shot Low-resource Language TEE with type

Tables 7 and 8 show the results for zero-shot low-resource language TEE when considering the types of the temporal expressions. Note that the superiority of our proposed XLTime over the previous automatic SOTA still holds.

G Language Models on English TEE

In our early experiments, we reexamine the language models on English TEE. This section presents the results.

G.1 Experimental Setup

We study BERT (Devlin et al., 2019) and XLMR (Conneau et al., 2020) variants, RoBERTa (Liu et al., 2019b) and T5 Encoder (Raffel et al., 2019). We compare them to rule-based methods including HeidelTime (Strötgen and Gertz, 2013), SynTime (Zhong et al., 2017), and PTime (Ding et al., 2019), which report SOTA performances on Wikiwars, TE3, and Tweets, respectively. We experiment on both settings, i.e., “with type" and “without type", and report F1, precision, and recall in strict match (UzZaman et al., 2013). We use the data splits following Ding et al. (2019) and the experimental settings introduced in Appendix D.

G.2 Evaluation Results

Table 9 shows the results. We observe: **1)** When ignoring the types, the language models are inferior to SynTime on TE3, on par with or better than the rule-based methods on Wikiwars and Tweets. **2)** When considering the types, the language models outperform the previous SOTAs by 11-22%, 18-21%, and 30-41% in F1 on TE3, Wikiwars, and Tweets datasets.

H Future Work

We observe negative transfer in some rare cases when transferring from multiple source languages (Tables 4 and 6). As suggested by Wu et al. (2020), the extent of negative transfer is affected by *task covariance*, which measures the similarities between the embedded task samples. We plan to verify this on XLTime by calculating and comparing the task covariances of the positively transferred cases to that of the negatively transferred cases.

One approach to reduce task covariance is to transform task sample embeddings by inserting an alignment layer between the lexicon encoder and the first Transformer layer. Wu et al. (2020) proposes an alignment layer design, i.e., one linear matrix for each of the tasks. However, as the training data for low-resource TEE is sparse, the parameters introduced by these matrices might cause the model to overfit. We plan to design a new alignment layer that is more suitable for XLTime. The new design aims to reduce task covariance while prompting parameter sharing and reducing overfitting.

Table 6: Zero-shot low-resource language TEE with additional source languages (precision and recall scores w/o type). The blue cells are expected to, while the underlined cells actually outperform (by $\geq 4\%$) using EN as the only source language.

Precision								
Target Language	FR				ES			
Source Language(s)	EN	EN, EU	EN, PT	EN, ES	EN	EN, EU	EN, PT	EN, FR
XLTime-mBERT	0.73	0.76	0.76	0.77	0.77	0.76	0.79	0.79
XLTime-XLMRbase	0.79	0.77	0.81	0.79	0.70	0.72	0.75	0.78
XLTime-XLMRlarge	0.79	0.81	0.84	0.82	0.79	0.70	0.79	0.74
Target Language	PT				EU			
Source Language(s)	EN	EN, FR	EN, ES	EN, EU	EN	EN, PT	EN, ES	EN, FR
XLTime-mBERT	0.64	0.77	0.67	0.77	0.81	0.78	0.79	0.82
XLTime-XLMRbase	0.67	0.72	0.60	0.54	0.76	0.82	0.79	0.86
XLTime-XLMRlarge	0.74	0.79	0.82	0.72	0.85	0.85	0.84	0.84
Recall								
Target Language	FR				ES			
Source Language(s)	EN	EN, EU	EN, PT	EN, ES	EN	EN, EU	EN, PT	EN, FR
XLTime-mBERT	0.72	0.77	0.69	0.82	0.66	0.69	0.66	0.74
XLTime-XLMRbase	0.78	0.76	0.75	0.86	0.63	0.64	0.68	0.68
XLTime-XLMRlarge	0.73	0.81	0.77	0.86	0.67	0.75	0.71	0.72
Target Language	PT				EU			
Source Language(s)	EN	EN, FR	EN, ES	EN, EU	EN	EN, PT	EN, ES	EN, FR
XLTime-mBERT	0.71	0.83	0.74	0.83	0.71	0.69	0.70	0.72
XLTime-XLMRbase	0.70	0.75	0.66	0.59	0.66	0.67	0.70	0.73
XLTime-XLMRlarge	0.81	0.87	0.87	0.77	0.71	0.74	0.74	0.71

Table 7: Zero-shot low-resource language TEE results (w/ type).

Model	FR			ES			PT			EU		
	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.
Automatic Baseline Models												
HeidelTime-auto	0.53	0.63	0.46	0.41	0.56	0.32	0.49	0.66	0.39	0.15	0.60	0.09
BiLSTM+CRF	0.58	0.64	0.51	0.56	0.61	0.51	0.58	0.59	0.58	0.44	0.54	0.37
mBERT	0.56	0.61	0.51	0.56	0.62	0.51	0.60	0.56	0.64	0.59	0.64	0.55
XLMR-base	0.64	0.69	0.59	0.51	0.58	0.46	0.59	0.59	0.59	0.43	0.60	0.34
XLMR-large	0.69	0.70	0.68	*0.68*	*0.71*	**0.66**	0.71	0.69	0.73	0.66	0.70	0.63
Cross-Lingual Transfer from EN (Ours)												
XLTime-mBERT	0.62	0.62	0.62	0.65	0.70	0.61	0.61	0.58	0.66	0.68	0.72	0.65
XLTime-XLMRbase	0.67	0.67	0.68	0.60	0.63	0.58	0.64	0.62	0.66	0.64	0.68	0.60
XLTime-XLMRlarge	*0.71*	**0.74**	0.68	**0.70**	**0.76**	*0.65*	*0.74*	*0.71*	*0.78*	*0.72*	**0.79**	*0.66*
Cross-Lingual Transfer from EN and Additional Source Languages (Ours)												
XLTime-mBERT	*0.71*	0.69	0.73	*0.68*	0.69	**0.66**	0.73	0.70	0.76	0.68	0.72	0.65
XLTime-XLMRbase	0.70	0.67	*0.74*	0.65	0.69	0.62	0.66	0.64	0.68	0.70	*0.76*	0.65
XLTime-XLMRlarge	**0.75**	*0.72*	**0.78**	**0.70**	**0.76**	*0.65*	**0.81**	**0.79**	**0.84**	**0.74**	**0.79**	**0.69**
Handcrafted Method												
HeidelTime	0.80	0.81	0.79	0.85	0.90	0.80	0.57	0.60	0.53	/	/	/

Table 8: Zero-shot low-resource language TEE with additional source languages (F1, precision, and recall scores w/ type). The blue cells are expected to, while the underlined cells actually outperform (by $\geq 3\%$) using EN as the only source language.

F1								
Target Language	FR				ES			
Source Language(s)	EN	EN, EU	EN, PT	EN, ES	EN	EN, EU	EN, PT	EN, FR
XLTime-mBERT	0.62	0.61	0.61	0.71	0.65	0.66	0.65	0.68
XLTime-XLMRbase	0.67	0.67	0.66	0.70	0.60	0.61	0.64	0.65
XLTime-XLMRlarge	0.71	0.73	0.73	0.75	0.70	0.68	0.69	0.68
Target Language	PT				EU			
Source Language(s)	EN	EN, FR	EN, ES	EN, EU	EN	EN, PT	EN, ES	EN, FR
XLTime-mBERT	0.61	0.72	0.59	0.73	0.68	0.66	0.66	0.68
XLTime-XLMRbase	0.64	0.66	0.55	0.52	0.64	0.66	0.66	0.70
XLTime-XLMRlarge	0.74	0.79	0.81	0.71	0.72	0.71	0.74	0.72
Precision								
Target Language	FR				ES			
Source Language(s)	EN	EN, EU	EN, PT	EN, ES	EN	EN, EU	EN, PT	EN, FR
XLTime-mBERT	0.62	0.59	0.62	0.69	0.70	0.69	0.71	0.69
XLTime-XLMRbase	0.67	0.66	0.67	0.67	0.63	0.64	0.67	0.69
XLTime-XLMRlarge	0.74	0.72	0.76	0.72	0.76	0.65	0.73	0.68
Target Language	PT				EU			
Source Language(s)	EN	EN, FR	EN, ES	EN, EU	EN	EN, PT	EN, ES	EN, FR
XLTime-mBERT	0.58	0.68	0.56	0.70	0.72	0.70	0.69	0.72
XLTime-XLMRbase	0.62	0.64	0.51	0.49	0.68	0.73	0.69	0.76
XLTime-XLMRlarge	0.71	0.75	0.79	0.68	0.79	0.75	0.79	0.79
Recall								
Target Language	FR				ES			
Source Language(s)	EN	EN, EU	EN, PT	EN, ES	EN	EN, EU	EN, PT	EN, FR
XLTime-mBERT	0.62	0.62	0.59	0.73	0.61	0.64	0.60	0.66
XLTime-XLMRbase	0.68	0.67	0.64	0.74	0.58	0.59	0.61	0.62
XLTime-XLMRlarge	0.68	0.73	0.71	0.78	0.65	0.71	0.65	0.67
Target Language	PT				EU			
Source Language(s)	EN	EN, FR	EN, ES	EN, EU	EN	EN, PT	EN, ES	EN, FR
XLTime-mBERT	0.66	0.75	0.62	0.76	0.65	0.63	0.64	0.64
XLTime-XLMRbase	0.66	0.68	0.60	0.55	0.60	0.60	0.63	0.65
XLTime-XLMRlarge	0.78	0.83	0.84	0.74	0.66	0.67	0.69	0.67

Table 9: Supervised English TEE results (w/\| w/o type).

Model	Datasets								
	TE3			Wikiwars			Tweets		
	F1	Pr.	Re.	F1	Pr.	Re.	F1	Pr.	Re.
`Rule-based Models`									
HeidelTime	0.77| 0.81	*0.80*| 0.84	0.75| 0.79	0.80| 0.85	0.86| 0.92	0.75| 0.80	0.80| 0.80	*0.90*| 0.90	0.72| 0.72
SynTime	0.65| **0.92**	0.65| **0.91**	0.66| **0.93**	0.79| 0.79	0.79| 0.79	0.79| 0.79	0.63| 0.92	0.62| 0.91	0.65| 0.95
PTime	0.67| *0.85*	0.68| *0.88*	0.65| 0.83	0.86| 0.86	0.87| 0.87	0.86| 0.86	0.66| **0.95**	0.65| **0.94**	0.67| *0.96*
`Language Models`									
BERT-base	0.76| 0.82	0.78| 0.85	0.74| 0.80	0.94| 0.94	*0.95*| *0.95*	0.94| 0.94	*0.92*| *0.94*	*0.90*| *0.93*	0.93| 0.95
BERT-large	**0.79**| 0.83	0.77| 0.82	**0.80**| *0.84*	0.95| 0.95	0.94| 0.94	0.96| 0.96	0.86| 0.92	0.84| 0.92	0.88| 0.92
mBERT	**0.79**| 0.84	*0.80*| 0.86	0.77| 0.82	**0.97**| **0.97**	**0.96**| **0.96**	*0.97*| *0.97*	0.87| 0.91	0.85| 0.88	0.90| 0.94
RoBERTa	*0.78*| 0.84	0.79| 0.86	0.77| 0.82	0.95| 0.95	0.94| 0.94	*0.97*| *0.97*	0.91| **0.95**	0.89| *0.93*	*0.94*| **0.97**
XLMR-base	**0.79**| 0.81	*0.80*| 0.82	0.77| 0.81	**0.97**| **0.97**	*0.95*| *0.95*	**0.98**| **0.98**	0.90| *0.94*	0.87| 0.92	0.93| **0.97**
XLMR-large	*0.78*| 0.81	0.78| 0.82	*0.78*| 0.81	*0.96*| *0.96*	0.94| 0.94	*0.97*| *0.97*	**0.93**| **0.95**	**0.91**| *0.93*	**0.95**| *0.96*
T5Encoder	**0.79**| 0.82	**0.82**| 0.85	*0.78*| 0.80	*0.96*| *0.96*	*0.95*| *0.95*	*0.97*| *0.97*	0.87| 0.93	0.84| 0.91	0.91| 0.95