

ECOLA: Enhancing Temporal Knowledge Embeddings with Contextualized Language Representations

Zhen Han^{*†7}, Ruotong Liao^{†1,6}, Jindong Gu², Yao Zhang¹, Zifeng Ding^{1,5},
Yujia Gu⁴, Heinz Koepl³, Hinrich Schütze^{1,6}, Volker Tresp^{1,6}

¹LMU Munich ²University of Oxford ³Technical University of Darmstadt

⁴Technical University of Munich ⁵Siemens AG

⁶Munich Center for Machine Learning (MCML), Munich, Germany ⁷Amazon

hanzhen02111@gmail.com, liao@dbi.lmu.de, volker.tresp@lmu.de

Abstract

Since conventional knowledge embedding models cannot take full advantage of the abundant textual information, there have been extensive research efforts in enhancing knowledge embedding using texts. However, existing enhancement approaches cannot apply to *temporal knowledge graphs* (tKGs), which contain time-dependent event knowledge with complex temporal dynamics. Specifically, existing enhancement approaches often assume knowledge embedding is time-independent. In contrast, the entity embedding in tKG models usually evolves, which poses the challenge of aligning *temporally relevant* texts with entities. To this end, we propose to study enhancing temporal knowledge embedding with textual data in this paper. As an approach to this task, we propose **Enhanced Temporal Knowledge Embeddings with Contextualized Language Representations** (ECOLA), which takes the temporal aspect into account and injects textual information into temporal knowledge embedding. To evaluate ECOLA, we introduce three new datasets for training and evaluating ECOLA. Extensive experiments show that ECOLA significantly enhances temporal KG embedding models with up to **287%** relative improvements regarding Hits@1 on the link prediction task. The code and models are publicly available[‡].

1 Introduction

Knowledge graphs (KGs) have long been considered an effective and efficient way to store structural knowledge about the world. A knowledge graph consists of a collection of *triples* (s, p, o) , where s (subject entity) and o (object entity) correspond to nodes, and p (predicate) indicates the edge type (relation) between the two entities. Common knowledge graphs (Toutanova et al., 2015;

Dettmers et al., 2018) assume that the relations between entities are static connections. However, in the real world, there are not only static facts but also temporal relations associated with the entities. To this end, temporal knowledge graphs (tKGs) (Tresp et al., 2015) were introduced that capture temporal aspects of relations by extending a triple to a *quadruple*, which adds a timestamp to describe when the relation is valid, e.g., $(R.T. \text{ Erdoğan, visit, US, 2019-11-12})$. If the temporal relationship lasts for several timestamps, most tKGs represent it by a sequence of quadruples, e.g., $\{(R.T. \text{ Erdoğan, visit, US, 2019-11-12}), (R.T. \text{ Erdoğan, visit, US, 2019-11-13})\}$.

Conventional knowledge embedding approaches learn KGs by capturing the structural information, suffering from the sparseness of KGs. To address this problem, some recent studies incorporate textual information to enrich knowledge embedding. KG-BERT (Yao et al., 2019) takes entity and relation descriptions of a triple as the input of a pre-trained language model (PLM) and turns KG link prediction into a sequence classification problem. Similarly, KEPLER (Wang et al., 2021) computes entity representations by encoding entity descriptions with a PLM and then applies KG score functions for link prediction. However, they could not be applied to tKGs. Specifically, existing approaches (e.g., KEPLER) encode an entity, no matter at which timestamp, with the same static embedding based on a shared entity description. In comparison, entity embeddings in tKG models usually evolve over time as entities often involve in different events at different timestamps. Therefore, an entity might be aligned with different textual knowledge at different time. And it should be taken into account which textual knowledge is relevant to which entity at which timestamp. We name this challenge as **temporal alignment** between texts and tKGs, which is to establish a correspondence between textual knowledge and their

^{*}Zhen Han’s work done prior to joining Amazon.

[†]Equal contribution.

[‡]<https://github.com/mayhugotong/ECOLA>

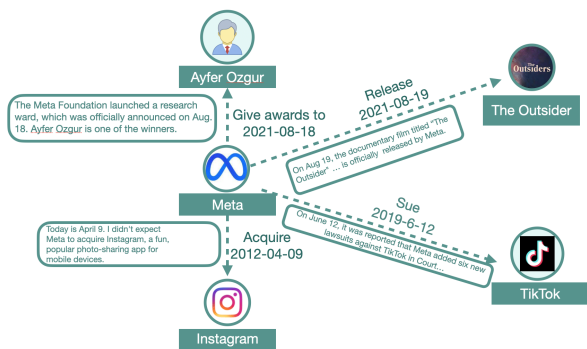


Figure 1: An example of a temporal knowledge graph with textual event descriptions.

tKG depiction. Another challenge is that many temporal knowledge embedding models (Goel et al., 2020; Han et al., 2020a) learn the entity representations as a function of time. However, the existing enhancement approaches cannot be naturally applicable to such tKG embedding. We refer to this challenge as **dynamic embedding challenge**. In this work, we propose to study *enhancing temporal knowledge embedding with textual data*. As an approach to this task, we develop Enhanced Temporal Knowledge Embeddings with Contextualized Language Representations (ECOLA), which uses temporally relevant textual knowledge to enhance the time-dependent knowledge graph embedding. Specifically, we solve the **temporal alignment challenge** using tKG quadruples as an implicit measure. We pair a quadruple with its relevant textual data, e.g., event descriptions, which corresponds to the temporal relations between entities at a specific time. Then we use the event description to enhance the representations of entities and the predicate involved in the given quadruple. Besides, ECOLA solves the **dynamic embedding challenge** using a novel knowledge-text prediction (KTP) task which injects textual knowledge into temporal knowledge embeddings. Specifically, given a quadruple-text pair, we feed both the temporal knowledge embeddings of the quadruple and token embeddings of the text into a PLM. The KTP task is an extended masked language modeling task that randomly masks words in texts and entities/predicates/timestamp in quadruples. With the help of the KTP task, ECOLA would be able to recognize mentions of the subject entity and the object entity and align semantic relationships in the text with the predicate in the quadruple.

For training ECOLA, we need datasets with tKG quadruples and aligned textual event descriptions,

which are unavailable in the existing temporal KG benchmarks. Thus, we construct three new temporal knowledge graph datasets by adapting two existing datasets, i.e., GDELTA (Leetaru and Schrodt, 2013) and Wiki (Dasgupta et al., 2018), and an event extraction dataset (Li et al., 2020).

To summarize, our contributions are as follows: (i) We are the first to address the challenge of enhancing temporal knowledge embedding with temporally relevant textual information while preserving the time-evolving properties of entity embedding. (ii) We construct three datasets to train the text-enhanced tKG models. Specifically, we adapt three existing temporal KG completion datasets by augmenting each quadruple with a relevant textual description. (iii) Extensive experiments show that ECOLA is model-agnostic and can be potentially combined with any temporal KG embedding model. ECOLA also has a superior performance on the temporal KG completion task and enhances temporal KG models with up to 287% relative improvements in the Hits@1 metric. (iv) As a joint model, ECOLA also empowers PLMs by integrating *temporal structured knowledge* into them. We select temporal question answering as a downstream NLP task, demonstrating that ECOLA can considerably enhance PLMs.

2 Preliminaries and Related Work

Temporal Knowledge Graphs Temporal knowledge graphs are multi-relational, directed graphs with labeled timestamped edges between entities (nodes). Let \mathcal{E} and \mathcal{P} represent a finite set of entities and predicates, respectively. A quadruple $q = (e_s, p, e_o, t)$ represents a timestamped and labeled edge between a subject entity $e_s \in \mathcal{E}$ and an object entity $e_o \in \mathcal{E}$ at a timestamp $t \in \mathcal{T}$. Let \mathcal{F} represent the set of all true quadruples, the temporal knowledge graph completion (tKGC) is the task of inferring \mathcal{F} based on a set of observed facts \mathcal{O} . Specifically, tKGC is to predict either a missing subject entity $(?, p, e_o, t)$ given the other three components or a missing object entity $(e_s, p, ?, t)$. We provide related works on temporal knowledge representations in Appendix A.

Joint Language and Knowledge Models Recent studies have achieved great success in jointly learning language and knowledge representations. Zhang et al. (2019) and Peters et al. (2019) focus on enhancing language models using external knowledge. They separately pre-train the entity em-

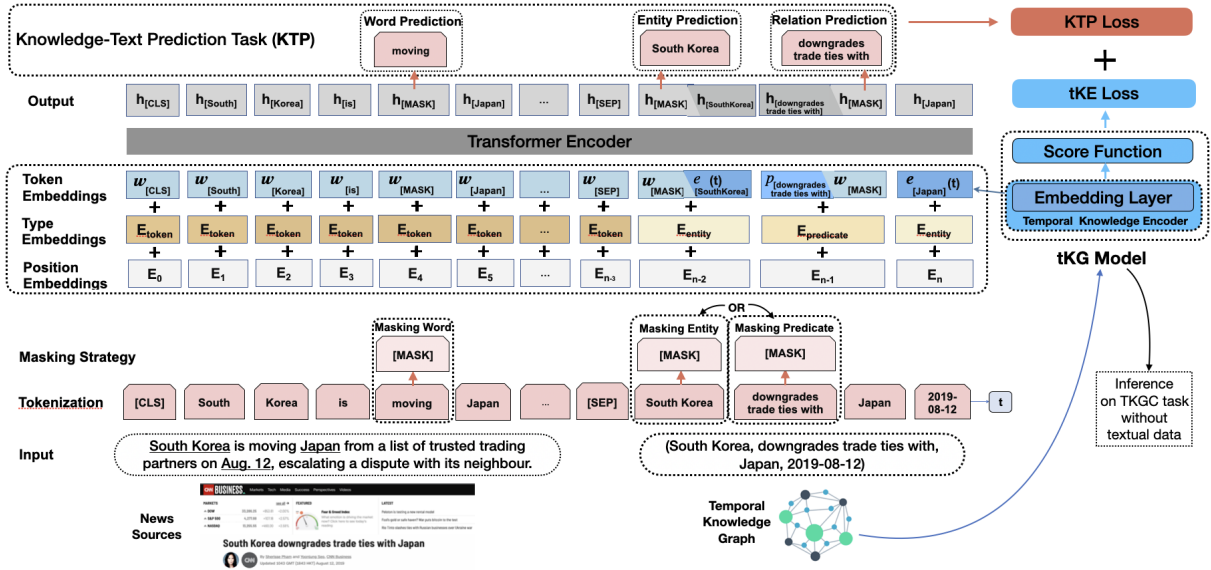


Figure 2: Model architecture. ECOLA jointly optimizes the knowledge-text prediction (KTP) objective and the temporal knowledge embedding (tKE) objective.

bedding with knowledge embedding models, e.g., TransE (Bordes et al., 2013), and inject the pre-trained entity embedding into PLMs, while fixing the entity embedding during training PLMs. Thus, they are not real joint models for learning knowledge embedding and language embedding simultaneously. Yao et al. (2019), Kim et al. (2020), and Wang et al. (2021) learn to generate entity embeddings with PLMs from entity descriptions. Moreover, He et al. (2019), Sun et al. (2020), and Liu et al. (2020) exploit the potential of contextualized knowledge representation by constructing subgraphs of structured knowledge and textual data instead of treating single triples as training units. Nevertheless, none of these works consider the temporal aspect of knowledge graphs, which makes them different from our proposed ECOLA.

3 ECOLA

In this section, we present the overall framework of ECOLA, including the model architecture in Section 3.1 - 3.3, a novel task designed for aligning knowledge embedding and language representation in Section 3.4, and the training procedure in Section 3.5. As shown in Figure 2, ECOLA implicitly incorporates textual knowledge into temporal knowledge embeddings by jointly optimizing the *knowledge-text prediction loss* and the *temporal knowledge embedding loss*. Note that, at inference time, we only take the enhanced temporal knowledge embeddings to perform the temporal

KG completion task without using PLM and any textual data for preventing information leakage and keeping a fast inference speed.

3.1 Embedding Layer

In tKG embedding models, entity representations evolve over time. Thus, the key point of enhancing a time-dependent entity representation $e_i(t)$ is to find texts that are relevant to the entity at the time of interest t . To this end, we use tKG quadruples (e.g., (e_i, p, e_j, t)) as an implicit measure for the alignment. We pair a quadruple with its relevant textual data and use such textual data to enhance the entity representation $e_i(t)$. Therefore, a training sample is a pair of quadruple from temporal KGs and the corresponding textual description, which are packed together into a sequence. As shown in Figure 2, the input embedding is the sum of token embedding, type embedding, and position embedding. For token embedding, we maintain **three lookup tables** for subwords, entities, and predicates, respectively. For subword embedding, we first tokenize the textual description into a sequence of subwords following (Devlin et al., 2018) and use the WordPiece algorithm (Wu et al., 2016). As the light blue tokens shown in Figure 2, we denote an embedding sequence of subword tokens as $\{w_1, \dots, w_n\}$. In contrast to subword embedding, the embeddings for entities and predicates are directly learned from scratch, similar to common knowledge embedding methods. We denote the entity embedding and predicate embedding as e

and \mathbf{p} , respectively, as the dark blue tokens shown in Figure 2. We separate the knowledge tokens, i.e., entities and predicates, and subword tokens with a special token [SEP]. To handle different token types, we add type embedding to indicate the type of each token, i.e., subword, entity, and predicate. For position embedding, we assign each token an index according to its position in the input sequence and follow Devlin et al. (2018) to apply fully-learnable absolute position embeddings.

3.2 Temporal Knowledge Encoder

As shown in Figure 2, the input embedding for *entities* and *predicates* consists of knowledge token embedding, type embedding, and position embedding. In this section, we provide details of the temporal knowledge embedding (tKE) objective.

A temporal embedding function defines entity embedding as a function that takes an *entity* and a *timestamp* t as input and generates a *time-dependent representation in a vector space*. There is a line of work exploring temporal embedding functions. Since we aim to propose a model-agnostic approach, we combine ECOLA with three temporal embedding functions, i.e., DyERNIE-Euclid (Han et al., 2020a), UTEE (Han et al., 2021c), and DE-Simple (Goel et al., 2020). In the following, we refer to DyERNIE-Euclid as DyERNIE and take it as an example to introduce our framework. Specifically, the entity representation is derived from an initial embedding and a velocity vector $\mathbf{e}_i^{DyER}(t) = \bar{\mathbf{e}}_i^{DyER} + \mathbf{v}_{e_i}t$, where $\bar{\mathbf{e}}_i^{DyER}$ represents the initial embedding that does not change over time, and \mathbf{v}_{e_i} is an entity-specific velocity vector. The combination with other temporal embedding functions is discussed in Section 4. The score function measuring the plausibility of a quadruple is defined as follows,

$$\begin{aligned} \phi^{DyER}(e_i, p, e_j, t) = \\ -d(\mathbf{P} \odot \mathbf{e}_i^{DyER}(t), \mathbf{e}_j^{DyER}(t) + \mathbf{p}) + b_i + b_j, \end{aligned} \quad (1)$$

where \mathbf{P} and \mathbf{p} represent the predicate matrix and the translation vector of predicate p , respectively; d denotes the Euclidean distance, and b_i, b_j are scalar biases. By learning tKE, we generate M negative samples for each positive quadruple in a batch. We choose the binary cross entropy as the temporal

knowledge embedding objective

$$\mathcal{L}_{tKE} = \frac{-1}{N} \sum_{k=1}^N (y_k \log(p_k) + (1-y_k) \log(1-p_k)), \quad (2)$$

where N is the sum of positive and negative training samples, y_k represents the binary label indicating whether a training sample is positive or not, p_k denotes the predicted probability $\sigma(\phi_k^{DyER})$, and $\sigma(\cdot)$ represents the sigmoid function.

3.3 Masked Transformer Encoder

To encode the input sequence, we use the pre-trained language representation model BERT (Devlin et al., 2018). Specifically, the encoder feeds a sequence of N tokens including *entities*, *predicates*, and *subwords* into the embedding layer introduced in Section 3.1 to get the input embeddings and then computes L layers of d -dimensional contextualized representations. Eventually, we get a contextualized representation for each token, which could be further used to predict masked tokens.

3.4 Knowledge-Text Prediction Task

To incorporate textual knowledge into temporal knowledge embedding, we use the pre-trained language model BERT to encode the textual description and propose a knowledge-text prediction task to align the language representations and the knowledge embedding. The knowledge-text prediction task is an extension of the masked language modeling (MLM) task. As illustrated in Figure 2, given a pair of a quadruple and the corresponding event description, the knowledge-text prediction task is to randomly mask some of the input tokens and train the model to predict the original index of the masked tokens based on their contexts. As different types of tokens are masked, we encourage ECOLA to learn different capabilities:

- **Masking entities.** To predict an entity token in the quadruple, ECOLA has the following ways to gather information. First, the model can detect the textual mention of this entity token and determine the entity; second, if the other entity token and the predicate token are not masked, the model can utilize the available knowledge token to make a prediction, which is similar to the traditional semantic matching-based temporal KG models. Masking entity nodes helps ECOLA align the representation spaces of language and structured knowledge,

and inject contextualized representations into entity embeddings.

- **Masking predicates.** To predict the predicate token in the quadruple, the model needs to detect mentions of the subject entity and object entity and classify the semantic relationship between the two entity mentions. Thus, masking predicate tokens helps the model integrate language representation into the predicate embedding and map words and entities into a common representation space.
- **Masking subwords.** When subwords are masked, the objective is similar to traditional MLM. The difference is that ECOLA considers not only the dependency information in the text but also the entities and the logical relationship in the quadruple. Additionally, we initialize the encoder with the pre-trained BERT_{base}. Thus, masking subwords helps ECOLA keep linguistic knowledge and avoid catastrophic forgetting while integrating contextualized representations into temporal knowledge embeddings.

In each quadruple, the predicate and each entity have a probability of 15% to be masked. Similarly, we mask 15% of the subwords of the textual description at random. We ensure that entities and the predicate cannot be masked at the same time in a single training sample, where we conduct an ablation study in Section 6 to show the improvement of making this constraint. When a token is masked, we replace it with (1) the [MASK] token 80% of the time, (2) a randomly sampled token with the same type as the original token 10% of the time, (3) the unchanged token 10% of the time. For each masked token, the contextualized representation in the last layer of the encoder is used for three classification heads, which are responsible for predicting entities, predicates, and subword tokens, respectively. At last, a cross-entropy loss \mathcal{L}_{KTP} is calculated over these masked tokens.

3.5 Training Procedure and Inference

We initialize the transformer encoder with BERT_{base}[§] and the knowledge encoder with random vectors. Then we use the temporal knowledge embedding (tKE) objective \mathcal{L}_{tKE} to train the knowledge encoder and use the knowledge-text

[§]<https://huggingface.co/bert-base-uncased>

prediction (KTP) objective \mathcal{L}_{KTP} to incorporate temporal factual knowledge and textual knowledge in the form of a multi-task loss:

$$\mathcal{L} = \mathcal{L}_{tKE} + \lambda \mathcal{L}_{KTP},$$

where λ is a hyperparameter to balance tKE loss and KTP loss. Note that those two tasks share the same embedding layer of entities and predicates. At inference time, we aim to answer link prediction queries, e.g., $(e_s, p, ?, t)$. Since there is no textual description at inference time, we take the entity and predicate embedding as input and use the score function of the knowledge encoder, e.g., Equation 1, to predict the missing links. Specifically, the score function assigns a plausibility score to each quadruple, and the proper object can be inferred by ranking the scores of all quadruples $\{(e_s, p, e_j, t), e_j \in \mathcal{E}\}$ that are accompanied with candidate entities.

4 The Model-Agnostic Property of ECOLA

ECOLA is model-agnostic and can enhance different temporal knowledge embedding models. Besides ECOLA-DyERNIE, we introduce here two additional variants of ECOLA.

ECOLA-DE enhances DE-SimpleE, which applies the diachronic embedding (DE) function (Goel et al., 2020). DE-function defines the temporal embeddings of entity e_i at timestamp t as

$$\mathbf{e}_i^{DE}(t)[n] = \begin{cases} \mathbf{a}_{e_i}[n] & \text{if } 1 \leq n \leq \gamma d, \\ \mathbf{a}_{e_i}[n] \sin(\boldsymbol{\omega}_{e_i}[n]t + \mathbf{b}_{e_i}[n]) & \text{else.} \end{cases} \quad (3)$$

Here, $\mathbf{e}_i^{DE}(t)[n]$ denotes the n^{th} element of the embeddings of entity e_i at time t . $\mathbf{a}_{e_i}, \boldsymbol{\omega}_{e_i}, \mathbf{b}_{e_i} \in \mathbb{R}^d$ are entity-specific vectors with learnable parameters, d is the dimensionality, and $\gamma \in [0, 1]$ represents the portions of the time-independent part.

ECOLA-UTEE enhances UTEE (Han et al., 2021c) that learns a *shared* temporal encoding for all entities to address the overfitting problem of DE-SimpleE on sparse datasets. Compared to ECOLA-DE, ECOLA-UTEE replaces Equation 3 with $\mathbf{e}_i^{UTEE}(t) = [\bar{\mathbf{e}}_i \parallel \mathbf{a} \sin(\boldsymbol{\omega}t + \mathbf{b})]$, $\bar{\mathbf{e}}_i \in \mathbb{R}^{\gamma d}$; $\mathbf{a}, \boldsymbol{\omega}, \mathbf{b} \in \mathbb{R}^{(1-\gamma)d}$, where $\bar{\mathbf{e}}_i$ denotes entity-specific time-invariant part, \parallel denotes concatenation, $\mathbf{a}, \boldsymbol{\omega}$, and \mathbf{b} are shared among all entities.

Table 1: Datasets Statistics

Dataset	# Entities	# Predicates	# Timestamps	# training set	# validation set	# test set
GDELТ	5849	237	2403	755166	94395	94395
DUEE	219	41	629	1879	247	247
WIKI	10844	23	82	233525	19374	19374

5 Datasets

Training ECOLA requires both temporal KGs and textual descriptions. Given a quadruple (e_s, p, e_o, t) , the key point is to find texts that are temporally relevant to e_s and e_o at t . Existing tKG datasets do not provide such information. To facilitate the research on integrating textual knowledge into temporal knowledge embedding, we reformat GDELТ[¶], DuEE[¶], and Wiki^{**}. We show the dataset statistics in Table 1.

GDELТ is an initiative knowledge base storing events across the globe connecting people and organizations, e.g., (*Google, consult, the United States, 2018/01/06*). For each quadruple, GDELТ provides the link to the news report which the quadruple is extracted from. We assume each sentence that contains both mentions of the subject and object is relevant to the given quadruple, and, thus, temporally aligned with the subject and object at the given timestamp. We pair each of these sentences with the given quadruple to form a training sample. This process is similar to the distant supervision algorithm (Mintz et al., 2009) in the relation extraction task. The proposed dataset contains 5849 entities, 237 predicates, 2403 timestamps, and 943956 quadruples with accompanying sentences.

DuEE is originally a human-annotated dataset for event extraction containing 65 event types and 121 argument roles. Each sample contains a sentence and several extracted event tuples. We select 41 event types that could be represented by quadruples and reformat DuEE by manually converting event tuples into quadruples and then pairing quadruples with their corresponding sentence.

Wiki is a temporal KG dataset proposed by Leblay and Chekol (2018). Following the post-processing by Dasgupta et al. (2018), we discretize the time span into 82 different timestamps. We align each entity to its Wikipedia page and extract

the first section as its description. To construct the relevant textual data of each quadruple, we combine the subject description, relation, and object description into a sequence. In this case, the knowledge-text prediction task lets the subject entity learn the descriptions of its *neighbors at different timestamps*, thus, preserving the temporal alignment between time-dependent entity representation and textual data.

6 Experiments

We evaluate the enhanced temporal knowledge embedding on the temporal KG completion task. Specifically, we take the entity and predicate embedding of ECOLA-DyERNIE and use Equation 1 to predict missing links. The textual description of test quadruples could introduce essential information and make the completion task much easier. Thus, to make a **fair comparison** with other temporal KG embedding models, we take the enhanced *lookup table embedding* of temporal KGs to perform the link prediction task at test time but use neither textual descriptions of test quadruples nor the language model. We report such results in Table 2. As additional results, we also show the prediction outcome that takes the text description of test quadruples as input in Figure 4a.

Baselines We include both static and temporal KG embedding models. From the static KG embedding models, we use TransE (Bordes et al., 2013), DistMult (Yang et al., 2014), and Simple (Kazemi and Poole, 2018). These methods ignore the time information. From the temporal KG embedding models, we compare our model with several state-of-the-art methods, including ATiSE (Xu et al., 2019), TNTComple (Lacroix et al., 2020), DyERNIE^{††} (Han et al., 2020a), TeRO (Xu et al., 2020), and DE-Simple (Goel et al., 2020). We provide implementation details in Appendix B and attach the source code in the supplementary material.

^{††}For a fair comparison with other baselines, we choose DyERNIE-Euclid.

[¶]<https://www.gdelтproject.org/data.html#googlebigquery>

[¶]<https://ai.baidu.com/broad/download>

^{**}https://www.wikidata.org/wiki/Wikidata:Main_Page

Table 2: Temporal link prediction results: Mean Reciprocal Rank (MRR, %) and Hits@1/3(%). The results of the proposed fusion models (with prefix ECOLA-) and their counterpart KG models are listed together.

Datasets	GDELT - filtered			Wiki - filtered			DuEE - filtered		
Model	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3
TransE	8.08	0.00	8.33	27.25	16.09	33.06	34.25	4.45	60.73
SimplE	10.98	4.76	10.49	20.75	16.77	23.23	51.13	40.69	58.30
DistMult	11.27	4.86	10.87	21.40	17.54	23.86	48.58	38.26	55.26
TeRO	6.59	1.75	5.86	32.92	21.74	39.12	54.29	39.27	63.16
ATiSE	7.00	2.48	6.26	35.36	24.07	41.69	53.79	42.31	59.92
TNTComplEx	8.93	3.60	8.52	34.36	22.38	40.64	57.56	43.52	65.99
DE-SimplE	12.25	5.33	12.29	42.12	34.03	45.23	58.86	44.74	68.62
ECOLA-DE	19.67 ± 00.11	16.04 ± 00.19	19.50 ± 00.04	43.53 ± 00.08	35.78 ± 00.17	46.42 ± 00.02	60.78 ± 00.16	47.43 ± 00.13	69.43 ± 00.64
UTEE	9.76	4.23	9.77	26.96	20.98	30.39	53.36	43.92	60.52
ECOLA-UTEE	19.11 ± 00.16	15.29 ± 00.38	19.46 ± 00.05	38.35 ± 00.22	30.56 ± 00.18	42.11 ± 00.14	60.36 ± 00.36	46.55 ± 00.51	69.22 ± 00.93
DyERNIE	10.72	4.24	10.81	23.51	14.53	25.21	57.58	41.49	70.24
ECOLA-DyERNIE	19.99 ± 00.05	16.40 ± 00.09	19.78 ± 00.03	41.22 ± 00.04	33.02 ± 00.06	45.00 ± 00.27	59.64 ± 00.20	46.35 ± 00.18	67.87 ± 00.53

Evaluation Protocol For each quadruple $q = (e_s, p, e_o, t)$ in the test set \mathcal{G}_{test} , we create two queries: $(e_s, p, ?, t)$ and $(?, p, e_o, t)$. For each query, the model ranks all possible entities \mathcal{E} according to their scores. Let $Rank(e_s)$ and $Rank(e_o)$ represent the rank for e_s and e_o of the two queries, respectively, we evaluate our models using standard metrics across the link prediction literature: *mean reciprocal rank (MRR)*: $\frac{1}{2 \cdot |\mathcal{G}_{test}|} \sum_{q \in \mathcal{G}_{test}} (\frac{1}{Rank(e_s)} + \frac{1}{Rank(e_o)})$ and *Hits@k* ($k \in \{1, 3, 10\}$): the percentage of times that the true entity candidate appears in the top k of ranked candidates.

Quantitative Study Table 2 reports the tKG completion results on the test sets, which are averaged over three trials. Firstly, we can see that ECOLA-UTEE improves its baseline temporal KG embedding model, UTEE, by a large margin, demonstrating the effectiveness of our fusing strategy. Specifically, ECOLA-UTEE enhances UTEE on GDELT with a *relative improvement* of **95%** and **99%** in terms of mean reciprocal rank (MRR) and Hits@3, even nearly **four times** better in terms of Hits@1. Thus, its superiority is clear on GDELT, which is the most challenging dataset among benchmark tKG datasets, containing nearly one million quadruples. Secondly, ECOLA-UTEE and ECOLA-DE generally outperform UTEE and DE-SimplE on the three datasets, demonstrating that ECOLA is model-agnostic and can enhance different tKG embedding models. Besides, in the DuEE dataset, ECOLA-DyERNIE achieves a better performance than DyERNIE in Hits@1 and MRR, but the gap

reverses in Hits@3. The reason could be that ECOLA-DyERNIE is good at classifying hard negatives using textual knowledge, and thus has a high Hits@1; however, since DuEE is much smaller than the other two datasets, ECOLA-DyERNIE may overfit in some cases, where the ground truth is pushed away from the top 3 ranks.

Ablation Study We compare DE-SimplE, ECOLA-DE, and ECOLA-SF on GDELT in Figure 3a. ECOLA-SF is the **static counterpart** of ECOLA-DE, where we do not consider the temporal alignment while incorporating textual knowledge. Specifically, ECOLA-SF integrates all textual knowledge into the **time-invariant part** of entity representations. We randomly initialize an embedding vector $\bar{e}_i \in \mathbb{R}^d$ for each entity $e_i \in \mathcal{E}$, where \bar{e}_i has the same dimension as the token embedding in the pre-trained language model. Then we learn the **time-invariant part** \bar{e}_i via the knowledge-text prediction task. For the temporal KG completion task, we combine \bar{e}_i with temporal knowledge embeddings,

$$\mathbf{e}_i^{SF}(t)[n] = \begin{cases} \mathbf{W}_{sf} \bar{e}_i[n] & \text{if } 1 \leq n \leq \gamma d, \\ \mathbf{a}_{e_i}[n] \sin(\omega_{e_i}[n]t + \mathbf{b}_{e_i}[n]) & \text{else,} \end{cases}$$

where $\mathbf{e}_i^{SF}(t) \in \mathbb{R}^d$ is an entity embedding containing static and temporal embedding part. $\mathbf{a}_{e_i}, \omega_{e_i}, \mathbf{b}_{e_i} \in \mathbb{R}^{d-\gamma d}$ are entity-specific vectors with learnable parameters. $\mathbf{W}_{sf} \in \mathbb{R}^{d \times \gamma d}$ is matrix with learnable weights. As shown in Figure 3a, the performance gap between ECOLA-DE and ECOLA-SF is significant, demonstrating the *tempo-*

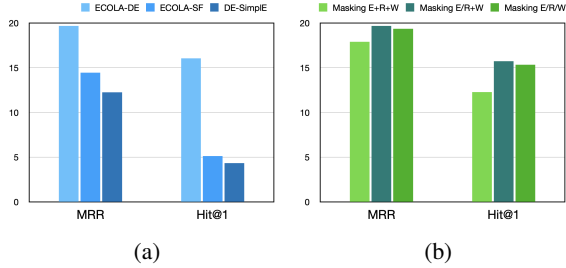


Figure 3: Ablation Study. (a) Temporal alignment analysis. We compare DE-Simple, ECOLA-DE, and ECOLA-SF in terms of MRR(%) and Hits@1(%) on GDELT. (b) Masking strategy analysis. We compare ECOLA-DE with different masking strategies and show the results of MRR(%) and Hits@1(%) on GDELT.

ral alignment between time-dependent entity representation and textual knowledge is more powerful than the *static alignment*.

Moreover, Figure 3b shows the results of different masking strategies on GDELT. The first strategy, e.g., *Masking E+R+W*, allows to simultaneously mask predicate, entity, and subword tokens in the same training sample. The second strategy is *Masking E/R+W*, where we mask 15% subword tokens in the language part, and either an entity or a predicate in the knowledge tuple. In the third strategy called *Masking E/R/W*, for each training sample, we choose to mask either subword tokens, an entity, or the predicate. Figure 3b shows the advantage of the second masking strategy, indicating that remaining adequate information in the knowledge tuple helps the model to align the knowledge embedding and language representations.

Qualitative Analysis To investigate why incorporating textual knowledge can improve the tKG embedding models’ performance, we study the test samples that have been correctly predicted by the fusion model ECOLA-DE but wrongly by the tKG model DE-Simple. It is observed that language representations help overcome the incompleteness of the tKG by leveraging knowledge from augmented textual data. For example, there is a test quadruple (*US, host a visit, ?, 2019-11-14*) with ground truth *R.T. Erdoğan*. The training set contains a quite relevant quadruple, i.e., (*Turkey, intend to negotiate with, US, 2019-11-11*). However, the given tKG does not contain information indicating that the entity *R.T. Erdoğan* is a representative of *Turkey*. So it is difficult for the tKG model DE-Simple to infer the correct answer from the above-mentioned quadruple. In ECOLA-DE, the augmented textual

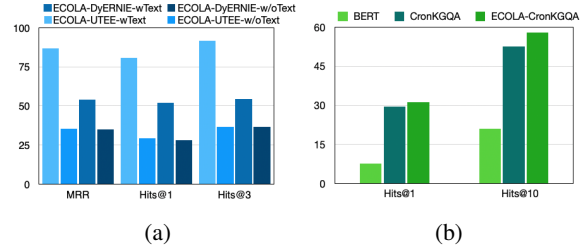


Figure 4: (a) Results of tKG completion task on GDELT *with* and *without* using a textual description of test quadruples. (b) ECOLA benefits language representations on the temporal question-answering task. The BERT model means that we directly apply BERT on the CronQuestions dataset, the CronKGQA is a model proposed by Saxena et al. (2021), and ECOLA-CronKGQA represents the model where we enhance CronKGQA using ECOLA.

data do contain such information, e.g. *"The president of Turkey, R.T. Erdoğan, inaugurated in Aug. 2014."*, which narrows the gap between *R.T. Erdoğan* and *Turkey*. Thus, by integrating textual information into temporal knowledge embedding, the enhanced model can gain additional information which the knowledge base does not include.

7 Discussion

Inference with Textual Data In Section 6, we compared different tKG embedding models, where textual data of test quadruples is absent during inference time. However, if the textual descriptions of the test quadruples are given during inference, will the contextualized language model incorporate this information into tKG embeddings? We use the entity predictor of the knowledge-text prediction task to perform the tKG completion task on GDELT. As shown in Figure 4a, the results show significant improvement across all metrics, specifically, 145% relatively higher regarding MRR of ECOLA-UTEE when given textual data during inference than not given. Thus, the results confirm that KTP task is a good choice for successful alignment between knowledge and language space and ECOLA utilizes the pre-trained language model to inject language representations into temporal knowledge embeddings.

Masking Temporal Information in KTP As temporal alignment is crucial for enhancing temporal knowledge embeddings, we study the effect of masking temporal information by extending the existing KTP task with an additional *time prediction task*, where the timestamp in the input is masked,

Table 3: Performance of masking temporal information on the knowledge-text prediction task.

Datasets	GDELТ - filtered			Wiki - filtered		
Model	MRR	Hits@1	Hits@3	MRR	Hits@1	Hits@3
ECOLA-UTEE	19.11	15.29	19.46	38.35	30.56	42.11
tECOLA-UTEE	20.39 (6.7% ↑)	16.83 (10.1% ↑)	20.08 (3.2% ↑)	42.53 (10.9% ↑)	34.06 (11.5% ↑)	46.32 (10.0% ↑)

and the model learns to predict the original timestamp. The extended model is named tECOLA-UTEE and has significant performance gain on both GDELТ and Wiki datasets across all metrics as shown in Table 3. We conjecture that the additional time prediction task forces the model to capture the temporal dynamics in temporal knowledge embeddings and utilize the temporal information in given textual descriptions. Since each temporal knowledge embedding models the temporal information in different ways, masking and predicting temporal information will be specific to each temporal knowledge embedding model. We leave this finding to future work for further inspections.

Temporal Question Answering Although we focus on generating informative temporal knowledge embeddings in this work, joint models often benefit both the language model and the temporal KG model mutually. Unlike previous joint models (Zhang et al., 2019; Peters et al., 2019), we do not modify the Transformer architecture, e.g., adding entity linkers or fusion layers. Thus, the language encoder enhanced by external knowledge can be adapted to a wide range of downstream tasks as easily as BERT. Besides the tKG completion task, we evaluate the enhanced language model in ECOLA on the temporal question-answering task to study its enhancement. Natural questions often include temporal constraints, e.g., who was the US president before Jimmy Carter? To deal with such challenging temporal constraints, temporal question answering over temporal knowledge base, formulated as TKGQA task, has become trendy since tKGs help to find entity or timestamp answers with support of temporal facts. Saxena et al. (2021) introduced the dataset CRONQUESTIONS containing natural temporal questions with different types of temporal constraints. They proposed a baseline CRONKGQA that uses BERT to understand the temporal constraints, followed by a scoring function for answer prediction. We apply ECOLA to enhance the BERT in CRONKGQA then plug it back into CRONKGQA and finetune it on the question

answering dataset. We name the enhanced model as ECOLA-CRONKGQA. The models are evaluated with standard metrics $Hits@k$ ($k \in \{1, 3\}$): the percentage of times that the true entity or time candidate appears in the top k of ranked candidates. Figure 4b shows that our proposed ECOLA considerably enhances CronKGQA, demonstrating the benefits of ECOLA to the language model.

8 Conclusion

We introduced ECOLA to enhance time-evolving entity representations with temporally relevant textual data using a novel knowledge-text prediction task. Besides, we constructed three datasets that contain paired structured temporal knowledge and unstructured textual descriptions, which can benefit future research on fusing *temporal* structured and unstructured knowledge. Extensive experiments show ECOLA can improve various temporal knowledge graph models by a large margin.

Limitations

To train ECOLA, we need to provide structured knowledge with aligned unstructured textual data to the model. Thus, we should either manually pair quadruples with event descriptions or use some matching algorithm to automatically build the pairs. The former requires human labeling effort and is hard to apply on large-scale datasets, while the latter would introduce noise into the dataset. Thus, ECOLA is currently tailored for domain adaptation and enhances pre-trained models with domain knowledge. There is still work to be done to let models be jointly trained on *large-scale* structured and unstructured data.

Ethics Statement

ECOLA is tailored to integrate temporal knowledge embedding and textual knowledge and can be applied to a wide variety of downstream tasks, such as temporal knowledge graph link prediction and temporal question answering. It can also power search and, thus, serve as a key intermediary of

information in users' lives. Since most temporal knowledge graphs are automatically extracted from web data, it's important to ensure it does not contain offensive content. ECOLA can be used to classify the quadruples in temporal knowledge graphs using the pre-trained language model and contribute to the knowledge graph protection's perspective.

Acknowledgement

The authors acknowledge support by the German Federal Ministry for Education and Research (BMBF), funding project "Software Campus 2.0 (LMU München) (grant 01IS17048)".

References

- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.
- Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2001–2011.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Zifeng Ding, Jingpei Wu, Bailan He, Yunpu Ma, Zhen Han, and Volker Tresp. 2022. Few-shot inductive learning on temporal knowledge graphs using concept-aware information. *arXiv preprint arXiv:2211.08169*.
- Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3988–3995.
- Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020a. DyERNIE: Dynamic Evolution of Riemannian Manifold Embeddings for Temporal Knowledge Graph Completion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7301–7316, Online. Association for Computational Linguistics.
- Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2021a. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *Proceedings of the 2021 International Conference on Learning Representations*.
- Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 2021b. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8352–8364, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zhen Han, Yunpu Ma, Yuyi Wang, Stephan Günnemann, and Volker Tresp. 2020b. Graph hawkes neural network for forecasting on temporal knowledge graphs. *arXiv preprint arXiv:2003.13432*.
- Zhen Han, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. 2021c. Time-dependent entity embedding is not all you need: A re-evaluation of temporal knowledge graph completion models under a unified framework. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8104–8118, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Bin He, Di Zhou, Jinghui Xiao, Qun Liu, Nicholas Jing Yuan, Tong Xu, et al. 2019. Integrating graph contextualized knowledge into pre-trained language models. *arXiv preprint arXiv:1912.00147*.
- Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2019. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. *arXiv preprint arXiv:1904.05530*.
- Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. *Advances in neural information processing systems*, 31.
- Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. 2020. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1737–1743.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. *arXiv preprint arXiv:2004.04926*.
- Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion Proceedings of the The Web Conference 2018*, pages 1771–1776.

- Kalev Leetaru and Philip A Schrod. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.
- Xinyu Li, Fayuan Li, Lu Pan, Yuguang Chen, Weihua Peng, Quan Wang, Yajuan Lyu, and Yong Zhu. 2020. Duee: a large-scale dataset for chinese event extraction in real-world scenarios. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 534–545. Springer.
- Wei jie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.
- Yunpu Ma, Volker Tresp, and Erik A Daxberger. 2019. Embedding models for episodic knowledge graphs. *Journal of Web Semantics*, 59:100490.
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.
- Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.
- Apoorv Saxena, Soumen Chakrabarti, and Partha Talukdar. 2021. Question answering over temporal knowledge graphs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.
- Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. *arXiv preprint arXiv:2109.04101*.
- Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. *arXiv preprint arXiv:2010.00309*.
- Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoi-fung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1499–1509.
- Volker Tresp, Cristóbal Esteban, Yinchong Yang, Stephan Baier, and Denis Krompaß. 2015. Learning with memory embeddings. *arXiv preprint arXiv:1511.07972*.
- Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020. TeRo: A time-aware knowledge graph embedding via temporal rotation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1583–1593, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2019. Temporal knowledge graph embedding model based on additive time series decomposition. *arXiv preprint arXiv:1911.07893*.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.
- Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kgbert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Appendix

A Related Work of Temporal Knowledge Embedding

Temporal Knowledge Embedding (tKE) is also termed Temporal Knowledge Representation Learning (TKRL), which is to embed entities and predicates of temporal knowledge graphs into low-dimensional vector spaces. TKRL is an expressive and popular paradigm underlying many KG models (Jin et al., 2019; Han et al., 2020b, 2021b; Sun et al., 2021; Lacroix et al., 2020; Ding et al., 2022). To capture temporal aspects, each model either embeds discrete timestamps into a vector space or learns time-dependent representations for each entity. Ma et al. (2019) developed extensions of static knowledge graph models by adding timestamp embeddings to their score functions. Besides, HyTE (Dasgupta et al., 2018) embeds time information in the entity-relation space by learning a temporal hyperplane to each timestamp and projects the embeddings of entities and relations onto timestamp-specific hyperplanes. Later, Goel et al. (2020) equipped static models with a diachronic entity embedding function which provides the characteristics of entities at any point in time and achieves strong results. Moreover, Han et al. (2020a) introduced a non-Euclidean embedding approach that learns evolving entity representations in a product of Riemannian manifolds. It is the first work to contribute to geometric embedding for tKG and achieves state-of-the-art performances on the benchmark datasets. Besides, Han et al. (2021a) proposed a subgraph reasoning algorithm for temporal knowledge graph forecasting, which can provide human-understandable evidence to its prediction.

B Implementation

We use the datasets augmented with reciprocal relations to train all baseline models. We tune the hyperparameters of our models using the random search and report the best configuration. Specifically, we set the loss weight λ to be 0.3, except for ECOLA-DE model trained on Wiki dataset where λ is set to be 0.001. We use the Adam optimizer (Kingma and Ba, 2014). We use the implementation of DE-Simple^{‡‡}, ATiSE/TeRO^{§§}. We use the code for TNTCompIEx from the tKG framework

^{‡‡}<https://github.com/BorealisAI/de-simple>

^{§§}<https://github.com/soledad921/ATiSE>

(Han et al., 2021c). We implement TTransE based on the implementation of TransE in PyKEEN^{¶¶}. We tune the model across a range of hyperparameters as shown in Table 6. We provide the detailed settings of hyperparameters of each baseline model and ECOLA in Table 4.

C The amount of Compute and the Type of Resources Used

We run our experiments on an NVIDIA A40 with a memory size of 48G. We provide the **training time** of our models and some baselines in Table 5. Note that there are no textual descriptions at inference time, and we take the entity and predicate embedding as input and use the score function of KG models to predict the missing links. Thus, the inference time of ECOLA (e.g., ECOLA-DE) and its counterpart KG model (e.g., DE-Simple) is the **same**. The numbers of parameters are in Table 7.

D The license of the Assets

We adapt three existing datasets, i.e., GDELТ, DuEE, and Wiki. We would first state the original license.

- **GDELТ**: as stated in the term of use of GDELТ^{***}, the GDELТ Project is an open platform for research and analysis of global society and thus all datasets released by the GDELТ Project are available for unlimited and unrestricted use for any academic, commercial, or governmental use of any kind without fee. One may redistribute, rehost, republish, and mirror any of the GDELТ datasets in any form. However, any use or redistribution of the data must include a citation to the GDELТ Project and a link to this website (<https://www.gdelтproject.org/>).
- **Wiki** is proposed by Dasgupta et al. (2018) and has Apache License 2.0.
- **DuEE** is released by Baidu Research. As stated on its website^{†††}, they have committed to provide these datasets at no cost for research and personal uses.

For the derived datasets, we only release a short version due to the size limit of uploads. Thus, we will release the full version and give the license,

^{¶¶}<https://github.com/pykeen/pykeen>

^{***}<https://www.gdelтproject.org/about.html#termsofuse>

^{†††}<https://ai.baidu.com/broad/introduction?dataset=duee>

Table 4: Hyperparameter settings of ECOLA and baselines.

Parameters	Embedding dimension			Negative Sampling			Learning rate			Batch Size		
Datasets	GDELt	DuEE	Wiki	GDELt	DuEE	Wiki	GDELt	DuEE	Wiki	GDELt	DuEE	Wiki
TransE	768	768	768	200	100	100	5e-4	5e-4	5e-4	256	128	256
Simple	768	768	768	200	100	100	5e-4	5e-4	5e-4	256	128	256
TTransE	768	768	768	200	100	100	5.2e-4	5.2e-4	5.2e-4	256	256	256
TNTComplEx	768	768	768	200	100	100	1.5e-4	1.5e-4	1.5e-4	256	256	256
DE-Simple	768	768	768	200	100	100	5e-4	5e-4	5e-4	256	128	256
ECOLA-SF	768	768	768	200	100	100	1e-4	2e-5	1e-4	64	16	64
ECOLA-DE	768	768	768	200	200	200	2e-5	2e-5	2e-5	4	8	4
ECOLA-UTEE	768	768	768	200	200	200	2e-5	2e-5	2e-5	4	8	4
ECOLA-dyERNIE	768	768	768	200	200	200	2e-5	e-4	2e-5	4	8	4

copyright information, and terms of use once the paper gets accepted.

Table 5: The runtime of the training procedure (in hours).

Dataset	GDELt	DuEE	Wiki
DE-Simple	17	0.5	5.0
ECOLA-DE	24.0	16.7	43.2
UTEE	67.3	0.5	11.3
ECOLA-UTEE	36.0	12.8	45.6
DyERNIE	25	0.1	5.9
ECOLA-DyERNIE	23.8	10.8	67.2

E Documentation of the artifacts

This paper uses three datasets, GDELt, Wiki, and DuEE. GDELt mainly covers social and political events written in English. Wiki in this paper mainly contains evolving knowledge, i.e., affiliation and residence place information, which is also written in English. DuEE is a dataset in Chinese and mainly talks about social news, such as the launch of new electronic products.

Table 6: Search space of hyperparameters.

Hyperparameter	Search space
learning rate	{e-5, 5e-5, e-4, 5e-4, e-3}
warm up	{0.05, 0.2, 0.3}
weight decay	{0.01, 0.05, 0.2}
batch size	{16, 128, 256, 512}

Table 7: The number of parameters (M).

Dataset	GDELt	DuEE	Wiki
DyERNIE	159	140	174
UTEE	150	139	158
DE	175	140	173

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
There is a Limitation section in the paper before the references.
- A2. Did you discuss any potential risks of your work?
The proposed method is a general method of jointly learning structural knowledge and textual knowledge. We do not see any obvious potentially risks.
- A3. Do the abstract and introduction summarize the paper’s main claims?
Abstract and Section 1.
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Section 1, 3, 4, 5.

- B1. Did you cite the creators of artifacts you used?
Section 5 and Section 6.
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix E.
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4 and Section 5.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The data we used is built upon the GDELT project (<https://www.gdeltproject.org/data.html>), which was collected from news webpages, e.g., BBC.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Appendix F.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 5 and Table 4 in the appendix.

C Did you run computational experiments?

Section 6.

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix D.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

Appendix C.

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

Section 6 and Table 1.

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

Appendix C.

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

Not applicable. Left blank.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

Not applicable. Left blank.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

Not applicable. Left blank.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

Not applicable. Left blank.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

Not applicable. Left blank.