Enhanced Temporal Knowledge Embeddings with Contextualized Language Representations

Zhen Han[1,2]*, **Ruotong Liao**[3]*, **Beiyan Liu**[3], **Yao Zhang**[1], **Zifeng Ding**[1,2], **Jindong Gu**,[1] **Heinz Köppl**[4], **Hinrich Schütze**[1], **Volker Tresp**[1,2]
[1]Institute of Informatics, LMU Munich [2] Corporate Technology, Siemens AG
[3]Department of Informatics, Technical University of Munich
[4]Department of Informatics, Technical University of Darmstadt
zhen.han@campus.lmu.de, volker.tresp@siemens.com

Abstract

Within the emerging research efforts to combine structured and unstructured knowledge, many approaches incorporate factual knowledge, e.g., available in form of structured knowledge graphs (KGs), into pre-trained language models (PLMs) and then apply the knowledge-enhanced PLMs to downstream NLP tasks. However, (1) they typically only consider *static* factual knowledge, whereas, e.g., knowledge graphs (KGs) also contain *temporal facts* or *events* indicating evolutionary relationships among entities at different timestamps. (2) PLMs cannot be directly applied to many KG tasks, such as temporal KG completion. In this paper, we focus on **e**nhancing temporal knowledge embeddings with **co**ntextualized **la**nguage representations (ECOLA). We align structured knowledge, contained in temporal knowledge graphs, with their textual descriptions extracted from news articles, and propose a novel knowledge-text prediction task to inject the abundant information from descriptions into temporal knowledge embeddings. ECOLA jointly optimizes the knowledge-text prediction objective and the temporal knowledge embeddings, which can simultaneously take full advantage of textual and knowledge information. The proposed fusion method is model-agnostic and can be combined with potentially any temporal KG model. For training ECOLA, we introduce three temporal KG datasets with aligned textual descriptions. Experimental results on the temporal knowledge graph completion task show that ECOLA outperforms state-of-the-art temporal KG models by a large margin. The proposed datasets can serve as new temporal KG benchmarks and facilitate future research on structured and unstructured knowledge integration.

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) and o (object) correspond to nodes in the graph connected through the edge type p (predicate). The nodes in KGs represent entities of the real world, and predicates describe relations between entity pairs. Common knowledge graphs assume that the relations between entities are static connections. However, in the real world, there are not only static facts and properties but also time-evolving relations associated with the entities. For example, the political relationship between two countries might worsen because of trade fights. To this end, temporal knowledge graphs (tKGs) were introduced that capture temporal aspects of relations by extending a triple to a *quadruple*, which adds a timestamp or time intervals to describe when the relation is valid, e.g. (*Argentina*, *deep comprehensive strategic partnership with*, *China*, *2022*). Since real-world tKGs are usually incomplete, the task of *temporal knowledge graph completion* has gained growing interest, which is to infer missing facts at specific timestamps by answering queries such as (US, president, ?, 2015). Extensive studies have been focusing on learning temporal knowledge embedding (tKE), and aim to effectively embed entities and relations into a low-dimensional vector space. tKE can not only help with the tKG completion task but also benefit various knowledge-

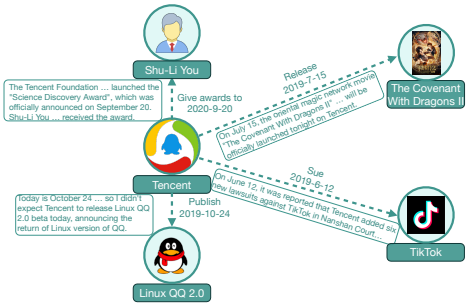

Figure 1: An example of a temporal knowledge graph with textual event descriptions.

*Equal Contribution.

related downstream applications, such as temporal question answering (Saxena et al., 2021) and time-aware recommendation systems (Zhao et al., 2021).

Typical temporal knowledge graphs, e.g., GDELT (Leetaru and Schrodt, 2013) and ICEWS (Boschee et al., 2015), solely focus on entities and relations. However, other semantic components, e.g., attributes, adjectives, adverbs, clauses, and tones, are generally ignored so that KGs cannot cover all available textual information. Thus, temporal knowledge embedding inherently suffers from the incompleteness of temporal knowledge graphs. To address this problem, additional information needs to be introduced to enrich the knowledge representations. An abundant resource is textual information, such as event descriptions in the original news article and entity descriptions in encyclopedias. Recent pre-trained language models (PLMs) such as BERT (Devlin et al., 2018) learn contextualized language representations from large-scale textual corpora with language modeling objectives. Intuitively, the informative textual knowledge captured by PLMs can benefit tKE by fusing both models.

We propose **E**nhanced Temporal Knowledge Embeddings with **Co**ntextualized **La**nguage Representations (ECOLA), which use a novel knowledge-text prediction task to align language representations with temporal knowledge embeddings and jointly optimize this prediction and the tKG completion objectives. For the knowledge-text prediction task, we pair quadruples with their textual descriptions from news articles. We use knowledge embeddings to encode entities and predicates and subword embeddings to encode text. Specifically, to capture the evolutionary dynamics of temporal KGs, we apply time-dependent entity embedding, such as diachronic embedding (Goel et al., 2020). Then we feed the quadruple-text pairs into a PLM and design the prediction task as an extended masked language modeling task by randomly masking words in texts and entity/predicates in quadruples. For the tKG completion objective, we train the temporal knowledge embeddings with benchmark tKG interaction models, i.e., proposed by Goel et al. (2020); Han et al. (2020b, 2021a).

There are also some recent works combining knowledge embedding and PLMs. ERNIE-THU (Zhang et al., 2019) and KnowBert (Peters et al., 2019) separately pre-train the entity embeddings with some knowledge embedding models, e.g., TransE (Bordes et al., 2013), and fix the embeddings during training PLMs. Thus, they are not real joint models for learning the knowledge embedding and language embedding simultaneously (Sun et al., 2020). KG-Bert (Yao et al., 2019), KEPLER (Wang et al., 2021), and ERNIE-Baidu (Sun et al., 2021) use entity description or fact description to bridge the gap between knowledge embeddings and PLMs. However, they ignored the temporal nature and the evolutionary dynamics of knowledge graphs. Thus, these approaches can only integrate PLMs with static factual knowledge but not temporal event-based knowledge. Besides, the models mentioned above enhance language of PLMs by injecting external knowledge from KGs. But for many knowledge graph tasks, e.g., tKG completion, it is not appropriate to apply PLMs. How to get enhanced temporal knowledge embedding to improve such tasks has not been well studied.

In comparison, the enhanced temporal knowledge embeddings of ECOLA can be directly applied in the tKG completion task. With the help of the knowledge-text prediction task, ECOLA would be able to recognize mentions of subject entity and object entity and align semantic relationships in the text with the predicates in the quadruple. Thus, the model can take full advantage of the abundant information from the event descriptions, which is especially helpful for embedding entities and predicates that only appear in a few quadruples. Moreover, we assume that the textual information about entities in tKG changes over time. Taking financial crises as an example, companies are more likely involved in events such as laying off employees. But when the economy recovers, companies hire staff again rather than cut jobs. Thus, the entities should also be able to drift their representations overtime to manage the changes. ECOLA is able to capture the evolutionary dynamics available in temporal multi-relation data by learning time-dependent entity embeddings. Since we pair quadruples with relevant texts that describe the temporal relation at the timestamp of interest, this correspondence ensures that the enhanced entity representations would preserve the temporal nature.

For training ECOLA, we need datasets with tKG quadruples and aligned textual event descriptions, which cannot be provided by existing tKG benchmark datasets. Thus, we construct new tKG completion datasets by adapting two existing tKG datasets, i.e., GDELT (Leetaru and Schrodt, 2013)

and YAGO (Leblay and Chekol, 2018), and an event extraction dataset (Li et al., 2020). We combine ECOLA with several benchmark tKG embedding models and show that ECOLA significantly improves their performance and achieves state-of-the-art performance. To make a fair comparison with other tKG models, we only take the enhanced temporal knowledge embeddings to perform the tKG completion task on the test set but do not use any textual event descriptions of test quadruples.

To summarize, our contributions are as follows: (i) We propose ECOLA: it enhances temporal knowledge graph representation models with contextualized language representations. ECOLA shows its superiority on the tKG completion task and can be potentially applied on a wide range of NLP tasks. (ii) We are the first to address the challenge of integrating temporal knowledge embedding and language representations while capturing the temporal dynamics available on tKGs. The proposed fusion method preserves the temporal nature and can be potentially combined with any tKG embedding model. (iii) To train the integration models, we construct three datasets, which align each quadruple with a relevant textual descriptions, by adapting three existing tKG completion datasets. Extensive experiments show that ECOLA is model-agnostic and enhances tKG embedding models with up to **287%** relative improvements in the Hits@1 metric.

2 Preliminaries and Related Work

Temporal Knowledge Graphs Temporal knowledge graphs are multirelational, directed graphs with labeled timestamped edges between entities (nodes). Let $\mathcal{E}$ and $\mathcal{P}$ represent a finite set of entities and predicates, respectively. tKG contains a collection of timestamped facts written as quadruples. 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 $\mathcal{F}$ represents the set of all true quadruples, i.e., real events in the world, the temporal knowledge graph completion (tKGC) is the task of inferring $\mathcal{F}$ based on a set of observed facts $\mathcal{O}$, which is a subset of $\mathcal{F}$. Specifically, tKGC is to predict either a missing subject entity $(?, p, e_o, t)$ given the other three components or a missing object entity $(e_s, p, ?, t)$. Temporal Knowledge Embedding (tKE) is also termed as Temporal Knowledge Representation Learning (TKRL), which is to embed entities and predicates of temporal knowledge graphs into low-dimensional vector spaces. We provided related works on temporal knowledge representations in Appendix A in the supplementary material.

Joint Language and Knowledge Models Recent studies have achieved great success in jointly learning language and knowledge representations. Yamada et al. (2016) and Ganea and Hofmann (2017) use entity linking to map entities and words into the same representation space. Inspired by the success of contextualized language representation, Zhang et al. (2019) and Peters et al. (2019) focus on enhancing language models with external knowledge by injecting pre-trained entity embedding of KGs. However, the static and inflexible pre-trained entity embeddings limit the knowledge gains. In comparison, Yao et al. (2019), Kim et al. (2020), and Wang et al. (2021) does not separately learn embeddings for each entity using KG models but learns to generate entity embeddings with PLMs from entity descriptions. Moreover, Sun et al. (2020), Liu et al. (2020) and He et al. (2019) exploits the potential of contextualized knowledge representation. Instead of treating single triples as training units, they construct subgraphs and integrate them with pre-trained language models. Nevertheless, none of these works consider the temporal aspect of knowledge graphs, which makes them different from our proposed ECOLA.

3 ECOLA

In the training phase, a training sample is a pair of a quadruple from tKGs and its corresponding textual event description, which are packed together into a sequence. As shown in Figure 2, ECOLA implicitly incorporates contextualized language representations into temporal knowledge embeddings by jointly optimizing the *knowledge-text prediction loss* and the *tKE loss*. Note that, at inference time, we only take the enhanced temporal knowledge embeddings to perform the tKG completion task without using any textual data for preventing information leak. In this section, we introduce the tKG representation model, the knowledge-text prediction task, and the training objectives.

3.1 Embedding Layer

As shown in Figure 3, the input embeddings are the sum of token embedding, type embedding, and position embedding. For token embedding, we

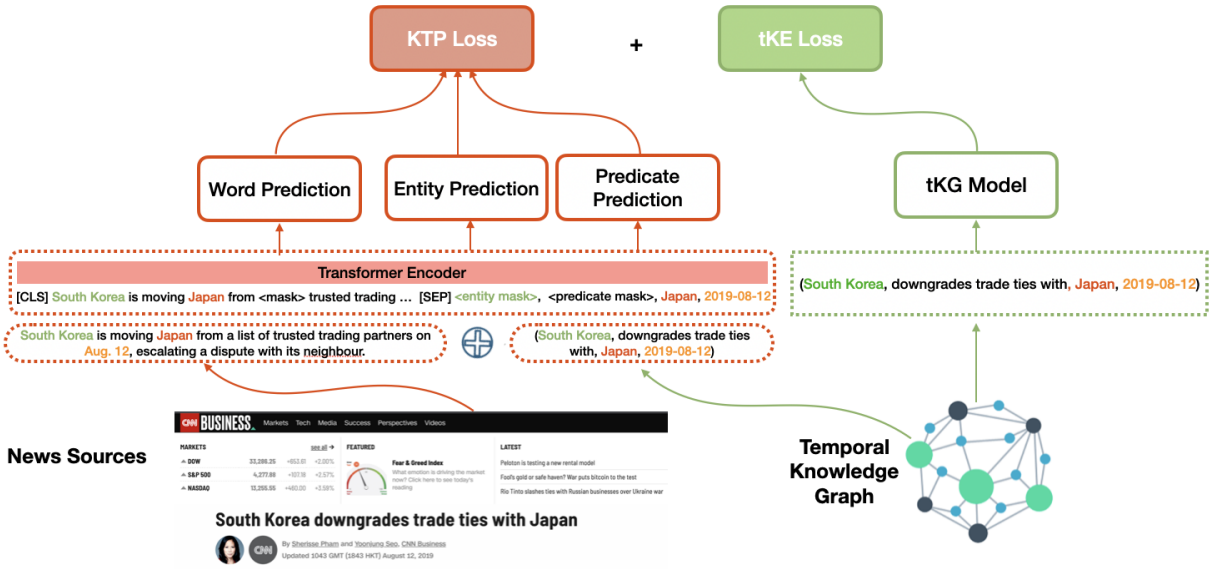

Figure 2: Model architecture. ECOLA jointly optimizes the knowledge-text prediction (KTP) objective and the temporal knowledge embedding (tKE) objective. For quadruples in the KTP loss and tKE loss, we apply knowledge embeddings. For tokens in sentences, we apply pre-trained token embeddings.

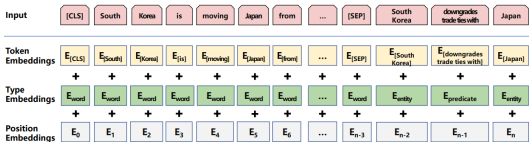

Figure 3: ECOLA input representation. The input embeddings are the sum of token embedding, type embedding, and position embedding.

maintain three lookup tables for subwords, entities, and predicates, respectively. For subword embedding, we first tokenize the textual description into a sequence of subword units to handle the large vocabulary by following Bert (Devlin et al., 2018) and use WordPiece embeddings (Wu et al., 2016) with a 30,000 token vocabulary. In contrast to subword embedding, the embeddings for entities and predicates are directly learned from scratch, similar to common knowledge embedding methods. 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 Embedding

As introduced in Section 3.1, 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 knowledge embedding and the tKE objective.

To capture the evolutionary dynamics on tKGs and the temporal information associated with the textual data, we apply temporal embedding functions to entities. Specifically, a temporal embedding function defines entity embedding as a function that takes an entity and a timestamp as input and generates a time-dependent representation for the entity at that time. There is a line of work exploring temporal embedding functions (Goel et al., 2020; Xu et al., 2019; Han et al., 2021b). Since our proposed training framework is model-agnostic, it can be principally combined with any temporal embeddings functions. In the following, we take the diachronic entity embedding (DE) function (Goel et al., 2020) as an example to introduce our framework. DE-function defines the temporal embeddings of entity e_i at timestamp t as

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

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 of the embedding,

and $\gamma \in [0, 1]$ represents the portions of the time-independent part. The combination with other temporal embedding functions are discussed in Section 3.6. We use the interaction model from Kazemi and Poole (2018) as the decoder, which considers two embeddings $\mathbf{e}_{i,s}^{DE}(t), \mathbf{e}_{i,o}^{DE}(t) \in \mathbb{R}^d$ for each entity e_i, i.e., as subject and as object separately, and two vectors $\mathbf{v}_p, \mathbf{v}_{p^{-1}} \in \mathbb{R}^d$ for each predicate p. The score function measuring the plausibility of a quadruple is defined as

$$\phi^{DE}(e_i, p, e_j, t) = \frac{1}{2}(\langle \mathbf{e}_{i,s}^{DE}(t), \mathbf{v}_p, \mathbf{e}_{j,o}^{DE}(t) \rangle + \langle \mathbf{e}_{j,s}^{DE}(t), \mathbf{v}_{p^{-1}}, \mathbf{e}_{i,o}^{DE}(t) \rangle) \quad (2)$$

where $\langle \mathbf{w}, \mathbf{v}, \mathbf{x} \rangle = \sum_{j=1}^{d} \mathbf{w}[j] * \mathbf{v}[j] * \mathbf{x}[j]$ represents the sum of the element-wise product of the vectors. By learning tKE, we generate M negative samples for each positive quadruple in a batch following Bordes et al. (2013)'s procedure. Then we choose the binary cross entropy as the tKE objective:

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

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 $\sigma(\phi_k^{DE})$, and $\sigma(\cdot)$ represents the sigmoid function.

3.3 Masked Transformer Encoder

To encode the input sequence, we use the pre-trained contextual language representation model Bert (Devlin et al., 2018) built on a multilayer bidirectional Transformer encoder (Vaswani et al., 2017). 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. At each layer, the encoder uses masked multi-head bidirectional self-attention to control the information flow and aggregate features non-locally. In addition to attention sub-layers, each of the layers in the encoder contains a multi-layer perceptron (MLP). Besides, a residual connection (He et al., 2016) is applied after each of the attention networks and MLP, followed by a layer normalization (Ba et al., 2016). 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 language representation into temporal knowledge embeddings, we introduce the knowledge-text prediction task, which is an extension of the masked language modeling (MLM) task. The knowledge-text prediction task requires both temporal knowledge graphs and unstructured textual descriptions. As illustrated in Figure 2, given a pair of quadruple from tKG and the corresponding event description from news article, 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 tKG 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 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 not only considers 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 $\text{BERT}_{\text{base}}$[1]. 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 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 Appendix C 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.

Although we focus on generating informative knowledge embeddings in this work, joint models often benefit both the language model and the tKG model because of the mutual information existing in language and tKGs. Unlike previous joint models (Zhang et al., 2019; Peters et al., 2019), we do not modify the Transformer encoder architecture, e.g., adding entity linkers or fusion layers. Thus, the encoder enhanced by external temporal knowledge can be adapted to a wide range of downstream tasks as easily as Bert.

3.5 Training Objectives and Inference

To enhance temporal knowledge embedding with the contextualized language representations, we design a multitask loss as

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

where $\mathcal{L}_{tKE}$ and $\mathcal{L}_{KTP}$ are the losses for tKE and the knowledge-text prediction, respectively. λ is a hyperparameter to balance tKE loss and KTP loss. Note that those two tasks share the same embedding layer for 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 in Equation 2 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.

3.6 Variants

As mentioned in Section 3.2, the proposed training framework is model-agnostic, which means it can potentially be combined with any temporal entity embedding models. Thus, we introduce here multiple versions of ECOLA combined with different temporal knowledge embedding models. Besides, we compare the effectiveness of enhancing temporal knowledge embedding and enhancing static knowledge embedding. In particular, we only feed the static part of entity embeddings into PLM to perform the knowledge-text prediction task. We refer it as ECOLA-SF (**S**tatic**F**usion).

ECOLA-DE is the principal model in our experiments that applies the diachronic function (Equation 1) to entity embedding and use SimplE (Equation 2) as the score function of temporal knowledge embedding.

ECOLA-SF is the static counterpart of ECOLA-DE, where we only apply temporal knowledge embedding to the tKG loss $\mathcal{L}_{tKE}$ but not to the knowledge-text prediction objective $\mathcal{L}_{KTP}$. Specifically, we randomly initialize an embedding vector $\bar{\mathbf{e}}_i \in \mathbb{R}^d$ for each entity $e_i \in \mathcal{E}$, where $\bar{\mathbf{e}}_i$ has the same dimension as the PLM token embedding. Then we learn $\bar{\mathbf{e}}_i$ via the knowledge-text prediction task. For the tKE objective, we have the following temporal knowledge embedding,

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

where $\mathbf{e}_i^{SF}(t) \in \mathbb{R}^d$ is an entity embedding containing static and temporal embedding parts. $\mathbf{a}_{e_i}, \boldsymbol{\omega}_{e_i}, \mathbf{b}_{e_i} \in \mathbb{R}^{d-\gamma d}$ are entity-specific vectors with learnable parameters. $\mathbf{W}_{sf} \in \mathbb{R}^{d \times \gamma d}$ is matrix with learnable weights. Note that $\mathbf{e}_i^{SF}(t)$ only plays a role in $\mathcal{L}_{tKE}$, and we use static embedding $\bar{\mathbf{e}}_i$ instead of $\mathbf{e}_i^{SF}(t)$ in $\mathcal{L}_{KTP}$.

ECOLA-UTEE takes the idea of UTEE (Han et al., 2021a) that learns a shared temporal embedding function for all entities to deal with the overfitting problem of the DE approach (Goel et al., 2020)

[1]https://huggingface.co/bert-base-uncased

on sparse datasets. Compared to ECOLA-DE, the only difference here is ECOLA-UTEE replaces Equation 1 with the follows:

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

where the amplitude vector $\mathbf{a}$, frequency vector $\boldsymbol{\omega}$, and bias $\mathbf{b}$ are shared for all entities.

ECOLA-DyERNIE adopts DyERNIE-Euclid (Han et al., 2020b) as the tKE model that associates each entity with a time-dependent representation. Specifically, the entity representation is derived from an initial embedding and a velocity vector to encode both the stationary properties and the time-varying behavior, i.e., $\mathbf{e}_i^{DyER}(t) = \bar{\mathbf{e}}_i^{DyER} + \mathbf{v}_{e_i} t$, 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. Besides, DyERNIE-Euclid takes the following score function, which is $\phi^{DyER}(e_i, p, e_j, t) = -d(\mathbf{P} \odot e_i^{DyER}(t), e_j^{DyER}(t) + \mathbf{p}) + b_i + b_j$, where $\mathbf{P}$ and $\mathbf{p}$ represent the diagonal predicate matrix and the translation vector of predicate p, respectively, and d is the Euclidean distance.

4 Datasets

The training procedure of ECOLA requires datasets of tKG quadruples with relevant textual descriptions, which existing tKG datasets do not provide. To facilitate the research on integrating temporal knowledge embeddings and language representations, we reformat three existing datasets, i.e., GDELT[2], DuEE[3], and Wiki[4], for evaluating the proposed integration method. We show the statistics of the datasets in Table 1 in the appendix of supplementary material.

GDELT is an initiative knowledge base storing events across the globe connecting people and organizations, e.g., (Google, consult, the United States, 2018/01/06 01:15). The quadruples were extracted from news reports using automated information extraction methods. For each quadruple, GDELT also provides links to news resource where the quadruple are extracted from. Since a news report is usually relevant to multiple quadruples, we use the following procedure to obtain pairs of a quadruple and its relevant sentence. Given a quadruple, each sentence that contains both mentions of subject entity and object entity is paired with this quadruple to form a training sample. This process is similar to the distant supervision algorithm (Mintz et al., 2009) in the relation extraction task, which assumes that, given a relationship between two entities, any sentence containing these two entities would express this relation. In total, the 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 training sample contains a sentence and the extracted event tuples with their occurrence timestamp. We construct a subset of DuEE by selecting event types that can be converted into quadruples and then pair the quadruples with their corresponding sentence.

Wiki is a tKG dataset proposed by Dasgupta et al. (2018). Different from GDELT and DuEE, time annotations in Wiki are represented as time intervals, e.g., (Savonranta, instance of, municipality of Finland, 1882 - 2009). Following the setting used in HyTE Dasgupta et al. (2018), we only deal with the year level granularity by dropping the month and date information and treat timestamps as 82 different time steps in the consideration of balancing the triple amount in different timestamps. To obtain text descriptions, we align each entity to its Wikipedia page and extract the relevant sentences as its description. Given a quadruple, we combine the subject entity description, relation, and object description to form a text description. The final KG contains 10844 entities, 23 predicates, and 272,273 quadruples.

5 Experiments

We evaluate the enhanced temporal knowledge embedding on the tKG completion task. Specifically, we take the entity and predicate embedding of ECOLA and use Equation 2 to predict missing links. To make a fair comparison with other tKG baseline models, we do not use any textual event descriptions of test set quadruples in the evaluation. Although this work focuses on tKG completion, the integrated tKE and PLM could be further used in many other downstream tasks, i.e., tKG forecasting and temporal question answering. We leave

[2]https://www.gdeltproject.org/data.html#googlebigquery
[3]https://ai.baidu.com/broad/download
[4]https://www.wikidata.org/wiki/Wikidata:Main_Page

Table 1: Link prediction results: MRR (%) and Hits@1/3/10 (%). The results of the proposed fusion models (in bold) and their counterpart KG models are listed together. The standard errors of the fusion models are also provided.

Datasets	GDELT - filtered				Wiki - filtered				DuEE - filtered			
Model	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	8.08	0.00	8.33	25.33	27.25	16.09	33.06	48.24	34.25	4.45	60.73	80.97
SimplE	10.98	4.76	10.49	23.67	20.75	16.77	23.23	27.62	51.13	40.69	58.30	68.62
DistMult	11.27	4.86	10.87	24.47	21.40	17.54	23.86	28.15	48.58	38.26	55.26	65.58
TeRO	6.59	1.75	5.86	15.58	32.92	21.74	39.12	53.45	54.29	39.27	63.16	85.02
ATiSE	7.00	2.48	6.26	14.61	35.36	24.07	41.69	54.74	53.79	42.31	59.92	75.91
TNTComplEx	8.93	3.60	8.52	19.01	34.36	22.38	40.64	56.03	57.56	43.52	65.99	83.60
TTransE	11.48	4.72	11.18	25.25	30.88	20.16	35.27	53.08	61.63	48.58	69.64	85.63
DE-SimplE	12.25	5.33	12.29	26.64	42.12	34.03	45.23	58.86	58.86	44.74	68.62	86.84
ECOLA-SF	14.44	5.11	20.32	26.40	42.28	35.22	44.88	56.27	60.64	46.96	69.64	87.45
ECOLA-DE	19.67 ± 00.11	16.04 ± 00.19	19.50 ± 00.04	25.58 ± 00.03	43.53 ± 00.08	35.78 ± 00.17	46.42 ± 00.02	60.26 ± 00.04	60.78 ± 00.16	47.43 ± 00.13	69.43 ± 00.64	86.70 ± 00.17
UTEE	9.76	4.23	9.77	21.29	26.96	20.98	30.39	37.57	53.36	43.92	60.52	68.62
ECOLA-UTEE	19.11 ± 00.16	15.29 ± 00.38	19.46 ± 00.05	25.59 ± 00.09	38.35 ± 00.22	30.56 ± 00.18	42.11 ± 00.14	53.02 ± 00.41	60.36 ± 00.36	46.55 ± 00.51	69.22 ± 00.93	87.11 ± 00.07
DyERNIE	10.72	4.24	10.81	24.00	23.51	14.53	25.21	41.67	57.58	41.49	70.24	86.23
ECOLA-DyERNIE	19.99 ± 00.05	16.40 ± 00.09	19.78 ± 00.03	25.67 ± 00.04	41.22 ± 00.06	33.02 ± 00.27	45.00 ± 00.20	57.17 ± 00.32	59.64 ± 00.18	46.35 ± 00.53	67.87 ± 00.29	85.48 ± 00.35

exploring it to future work.

5.1 Baselines

In the experiments, 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) where these methods ignore the available time information. From the temporal KG embedding models, we compare the performance of our model with several state-of-the-art methods, including TTransE (Leblay and Chekol, 2018), AiTSEE (Xu et al., 2019), DE-SimplE (Goel et al., 2020), TNTComplE(Lacroix et al., 2020), DyERNIE[5] (Han et al., 2020b), and TeRO (Xu et al., 2020). We provide implementation details in Appendix B and attach the source code in the supplementary material.

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. Following the filtered setting in Han et al. (2020a), we remove all entity candidates that correspond to true triples from the candidate list apart from the current test entity. 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.

[5]For a fair comparison with other baselines, we choose DyERNIE-Euclid.

5.2 Comparative Study

Link prediction. Table 1 reports the tKG completion results on the test sets, which are averaged over three trials. We also report the error bars of the ECOLA models. We see that ECOLA-DyERNIE improves its baseline tKG model, DyERNIE, by a large margin, demonstrating the effectiveness of our fusing strategy. Specifically, ECOLA enhances DyERNIE with a *relative improvement* of up to **86%** on GDELT 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 with million quadruples and several hundreds relations. Similarly, ECOLA-UTEE and ECOLA-DE generally outperform UTEE and DE-SimplE on all three datasets, respectively, demonstrating that ECOLA is model-agnostic and can potentially enhance many tKG embedding models.

Static Fusing vs. Temporal Fusing. Comparing ECOLA-DE with ECOLA-SF, we observe that ECOLA-DE generally performs better than ECOLA-SF, indicating that the textual knowledge also changes temporally. Thus, we need temporal embeddings to characterize such evolving knowl-

edge from contextualized language representations, instead of only aligning the static part of graph embeddings with the language representations.

6 Conclusion

We propose ECOLA to enhance temporal knowledge embedding using contextualized language representations. A novel knowledge-text prediction task is introduced to align the temporal knowledge and language representation into the same semantic space. We train ECOLA with both the temporal knowledge embedding objective and the knowledge-text prediction objective. Besides, we construct three datasets that contain paired structured temporal knowledge and unstructured textual descriptions, which can benefit future research on fusing temporal structured and unstructured knowledge. We evaluate a number of baseline methods on the proposed datasets. Extensive experiments show that ECOLA is model-agnostic and can improve several temporal knowledge graph models by a large margin. Besides, joint models often benefit both language encoders and temporal knowledge embedding. In future, we will evaluate the ECOLA on other NLP tasks to investigate whether the enhanced PLM can better understand event-related texts and support the extraction of temporal knowledge from texts.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

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.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. Icews coded event data. *Harvard Dataverse*, 12.

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.

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*.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*.

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. xerte: Explainable reasoning on temporal knowledge graphs for forecasting future links. *arXiv preprint arXiv:2012.15537*.

Zhen Han, Yunpu Ma, Peng Chen, and Volker Tresp. 2020b. Dyernie: Dynamic evolution of riemannian manifold embeddings for temporal knowledge graph completion. *arXiv preprint arXiv:2011.03984*.

Zhen Han, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. 2021a. 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.

Zhen Han, Gengyuan Zhang, Yunpu Ma, and Volker Tresp. 2021b. 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.

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*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

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 Schrodt. 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.

Weijie 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. *arXiv preprint arXiv:2106.01515*.

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*.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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*.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*.

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. Kg-bert: 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*.

Yuyue Zhao, Xiang Wang, Jiawei Chen, Wei Tang, Yashen Wang, Xiangnan He, and Haiyong Xie. 2021. Time-aware path reasoning on knowledge graph for recommendation. *arXiv preprint arXiv:2108.02634*.

A Related Work of Temporal Knowledge Embedding

Temporal Knowledge Embedding (tKE) is also termed as 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. 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. (2020b) 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. In particular, ECOLA is model-agnostic, which any temporal KG embedding model can be potentially enhanced by training with the knowledge-text task.

B Implementation

We use the datasets augmented with reciprocal relations to train all baseline models. We tune 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 Baidu 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[6], ATiSE/TeRO[7]. We use the code for TNT-CopmlEx from the tKG framework (Han et al., 2021a). We implement TTransE based on the implementation of TransE[8]. We provide the detailed settings of hyperparameters of each baseline model and ECOLA in Table 3 in the appendix.

[6]https://github.com/BorealisAI/de-simple
[7]https://github.com/soledad921/ATISE
[8]https://github.com/pykeen

C Ablation study

Masking Strategy Table 4 shows the results of different masking strategies on GDELT. The first strategy (Masking E+R+W) allows to simultaneously mask predicate, entity, and subword tokens in the same training sample. In the second strategy (Masking E/R+W), we mask 15% subword tokens in the language part, and either an entity or a predicate in the knowledge tuple. In other words, simultaneously masking an entity and a predicate in a training sample is not allowed. In the third strategy, for each training sample, we choose to mask either subword tokens, an entity, or the predicate. The experimental results show 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.

Type Embedding Table 5 demonstrates the effectiveness of the proposed type embedding. The type embedding differentiates among subword tokens, entity, and predicate of the input. To investigate its contribution, a uniform embedding is implemented, where all three token types have shared type embeddings. Empirical results on GDELT indicate that providing distinguishment between subword tokens, entity tokens, and predicate tokens helps the model to better understand different input components and different prediction tasks.

Table 2: Datasets Statistics

Dataset	# Entities	# Predicates	# Timestamps	Time Granularity	# training set	# validation set	# test set
GDELT	5849	237	2403	15mins	755166	94395	94395
DUEE	219	41	629	day	1879	247	247
WIKI	10844	23	82	year	233525	19374	19374

Table 3: 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

Masking Strat.	**MRR**	**Hits@1**	**Hits@10**
Masking E+R+W	17.89	12.27	25.77
Masking E/R+W	19.66	15.73	25.84
Masking E/R/W	19.35	15.33	25.65

Table 4: ECOLA-DE with different masking strategies applied to the knowledge-text prediction task on GDELT.

Type Embedd.	**MRR**	**Hits@1**	**Hits@10**
Type Embedding	19.85	16.38	25.50
Uniform Embedding	19.53	15.02	24.73

Table 5: ECOLA-DE with/without type embeddings in knowledge-text-prediction task on GDELT.