An Effective and Efficient Time-aware Entity Alignment Framework via Two-aspect Three-view Label Propagation

Li Cai[1,2] , **Xin Mao**[1] , **Youshao Xiao**[5] , **Changxu Wu**[6] and **Man Lan**[1,3,4*]
[1]School of Computer Science and Technology, East China Normal University, Shanghai 200062, China
[2]College of Computer Science and Technology, Guizhou University, Guiyang 5220025, China
[3]Shanghai Institute of AI for Education, East China Normal University
[4]Lingang Laboratory, Shanghai 200031, China
[5]Ant Group, China
[6]Department of Industrial Engineering, Tsinghua University
lcai2020@stu.ecnu.edu.cn, xmao@stu.ecnu.edu.cn, youshao.xys@antgroup.com,
wuchangxu@tsinghua.edu.cn, mlan@cs.ecnu.edu.cn

Abstract

Entity alignment (EA) aims to find the equivalent entity pairs between different knowledge graphs (KGs), which is crucial to promote knowledge fusion. With the wide use of temporal knowledge graphs (TKGs), time-aware EA (TEA) methods appear to enhance EA. Existing TEA models are based on Graph Neural Networks (GNN) and achieve state-of-the-art (SOTA) performance, but it is difficult to transfer them to large-scale TKGs due to the scalability issue of GNN. In this paper, we propose an effective and efficient non-neural EA framework between TKGs, namely LightTEA, which consists of four essential components: (1) Two-aspect Three-view Label Propagation, (2) Sparse Similarity with Temporal Constraints, (3) Sinkhorn Operator, and (4) Temporal Iterative Learning. All of these modules work together to improve the performance of EA while reducing the time consumption of the model. Extensive experiments on public datasets indicate that our proposed model significantly outperforms the SOTA methods for EA between TKGs, and the time consumed by LightTEA is only dozens of seconds at most, no more than 10% of the most efficient TEA method.

1 Introduction

Knowledge graphs (KGs) describe the real world with structured facts. A fact consists of a head entity, a tail entity, and a relation connecting them, which can be formally represented as a triple (e_h, r, e_t), such as (*George Porter, hasWonPrize, Nobel Prize in Chemistry*). KGs have been widely used in information retrieval [Dietz *et al.*, 2018], question answering [Lan *et al.*, 2021], and recommedation systems [Guo *et al.*, 2022].

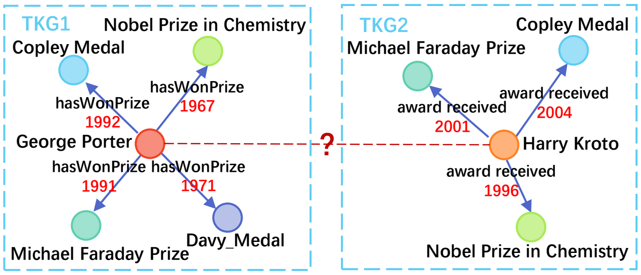

Figure 1: Example of the wrong alignment between similar entities in different temporal knowledge graphs.

Existing KGs ignore the temporal information which indicates when a fact occurred. In the real world, some facts only happened at a specific time. Therefore, Wikidata [Vrandecic and Krötzsch, 2014] and YOGO2 [Hoffart *et al.*, 2013] add temporal information to represent the KGs more accurately, and some event KGs [Lautenschlager *et al.*, 2015; Leetaru and Schrodt, 2013] also contain the timestamps indicating when the events occurred. In the temporal knowledge graphs (TKGs), a fact is expanded into a quadruple (e_h, r, e_t, τ), where τ represents the timestamps.

Entity alignment (EA) seeks to find the same entities in the real world between different KGs, which is important for knowledge fusion between multi-source and multi-lingual KGs. In recent years, the embedding-based EA methods [Sun *et al.*, 2020] have been widely investigated, which represent the entities in the low-dimensional vector space, and calculate the similarity of these vectors to obtain the equivalent entity pairs. Earlier EA methods [Chen *et al.*, 2017; Sun *et al.*, 2017] are based on the translation model. However, since the inability of such models to effectively capture graph structures, graph neural networks (GNN)-based models [Wang *et al.*, 2018; Mao *et al.*, 2020a] emerge and achieve superior performance. Due to the scalability limitation of GNN [Wu *et al.*, 2022], these models are not suitable for large-scale KGs. LightEA [Mao *et al.*, 2022b] significantly enhances the efficiency of EA by utilizing a three-view label propagation instead of GNN.

[*]Corresponding author

Despite the success of the above EA methods, they all ignore the temporal information in the KGs, thus may lead to wrong alignment between similar entities in different KGs. Take Figure 1 as an example. There are two subgraphs from YAGO and Wikidata. *George Porter* in TKG1 and *Harray Kroto* in TKG2 have similar structure and relations. Both of them have the same neighbors (*Copley Medal*, *Nobel Prize in Chemistry*, and *Michael Faraday Prize*) and relations (*hasWonPrize* in TKG1 is same as *award recieved* in TKG2). *George Porter* has another neighbor *Davy Medal* and relation *hasWonPrize*. Existing EA methods disregard temporal information of KGs and may wrongly take *George Porter* and *Harray Kroto* as an equivalent entity pair.

Recently, time-aware methods [Xu *et al.*, 2021; Xu *et al.*, 2022] emerge to improve the performance of EA between TKG. STEA [Cai *et al.*, 2022] adopts a simple GNN with a temporal information matching mechanism and achieves state-of-the-art (SOTA) performance. All of these TEA methods are based on GNN [Wu *et al.*, 2022], which has an inherent defect: the GNN is trained using the gradient descent algorithm and takes much time to converge to the optimal solution. Therefore, to promote the development of EA between TKGs, a straightforward approach is to combine the advantages of EA and TEA methods.

To this end, we propose an effective and efficient non-neural EA framework between TKGs, namely LightTEA, which consists of four key components: *(1) Two-aspect Three-view Label Propagation.* We combine relational-aspect and temporal-aspect three-view label propagation (LP) to improve the performance of EA. In this module, GNN is replaced by LP, which does not require gradient propagation to train the neural networks, greatly reducing the time consumption of the model. *(2) Sparse Similarity with Temporal Constraints.* Instead of calculating the similarity of all entities, we only retrieve the *top-k* nearest neighbors of each entity to find the equivalent entity pairs, which saves the time complexity and space complexity of calculation. By utilizing the temporal information of each entity as the constraints of entity similarity, the EA performance is also improved. *(3) Sinkhorn Operator.* To further promote the model's effectiveness, we regard the EA problem as a one-to-one assignment problem and use the Sinkhorn operator to solve it. The Sinkhorn operator is a fast and completely parallelizable algorithm and only takes seconds to converge to the optimal solution. and *(4) Temporal Iterative Learning.* Several studies have demonstrated that iterative learning effectively enhances EA and helps address the lack of alignment seeds in the real scenario. We adopt a temporal iterative learning strategy, which utilizes the additional temporal information of entities to get more credible alignment pairs for augmenting the training set to obtain better alignment results. In general, this paper presents the following contributions:

- We propose an effective and efficient TEA framework that consists of four essential components: (1) Two-aspect Three-view Label Propagation, (2) Sparse Similarity with Temporal Constraints, (3) Sinkhorn Operator, and (4) Temporal Iterative Learning. All of these modules work together to improve the performance of EA while reducing time consumption.
- The proposed TEA framework combines the strengths of the latest SOTA EA and TEA models and addresses the limitations of current EA models that do not effectively utilize time information, as well as the scalability constraints of TEA models due to GNN usage.
- Extensive experiments on public datasets indicate that our proposed model significantly outperforms the SOTA methods for EA between TKGs, and the time consumed by LightTEA is only dozens of seconds at most, no more than 10% of the most efficient TEA method.

2 Related Work

2.1 Entity Alignment

The purpose of EA is to find the equivalent entity pairs from different KGs. EA usually adopts embedding-based approaches, which are divided into two sub-categories: translation-based and GNN-based models.

Translation-based models regard the relations as the translation from the head entities to the tail entities, such as $(\boldsymbol{h}_{e_h} + \boldsymbol{h}_r \approx \boldsymbol{h}_{e_t})$. MTransE [Chen *et al.*, 2017] is the early entity alignment model based on TransE [Bordes *et al.*, 2013], which maps two KGs into different vector spaces and considers the entities with similar positions in the two vector spaces as equivalent pairs. In addition to learning the structure embeddings of entities based on TransE with the relation triples, JAPE [Sun *et al.*, 2017] joins the attribute embedding and structure embedding of entities to align entities. BootEA [Sun *et al.*, 2018] obtains the alignment-oriented KG embeddings and proposes a bootstrapping process by adding likely alignment entities into training data iteratively to improve the performance of EA.

GNN-based models promote EA by utilizing the graph structure of KGs. GCN-Align [Wang *et al.*, 2018] encodes the entities into a unified vector space via GCN [Kipf and Welling, 2017] and aligns the entities with their structure embeddings and attribute embeddings. However, GCN-Align does not effectively utilize the relation of KGs. MRAEA [Mao *et al.*, 2020a] learns the entity embeddings through a relation-aware self-attention GNN to obtain the alignment entities. RREA [Mao *et al.*, 2020b] proposes a GNN with relational reflection to get the relation-specific embeddings and uses an iterative strategy to enhance EA.

The above methods focus on encoding the entity embeddings and aligning the entities by calculating the similarity of their embeddings, DATTI [Mao *et al.*, 2022a] applies a decoding process using the adjacency and inner correlation isomorphisms of KGs to the advanced methods and gains significant performance improvements. LightEA [Mao *et al.*, 2022b] adopts a three-view label propagation approach for entity alignment instead of using GNN and achieves comparable performance with SOTA EA methods while taking only one-tenth of the time consumption of those methods.

Although these methods have significantly advanced the development of EA, they all have the limitation of neglecting the temporal information in KGs.

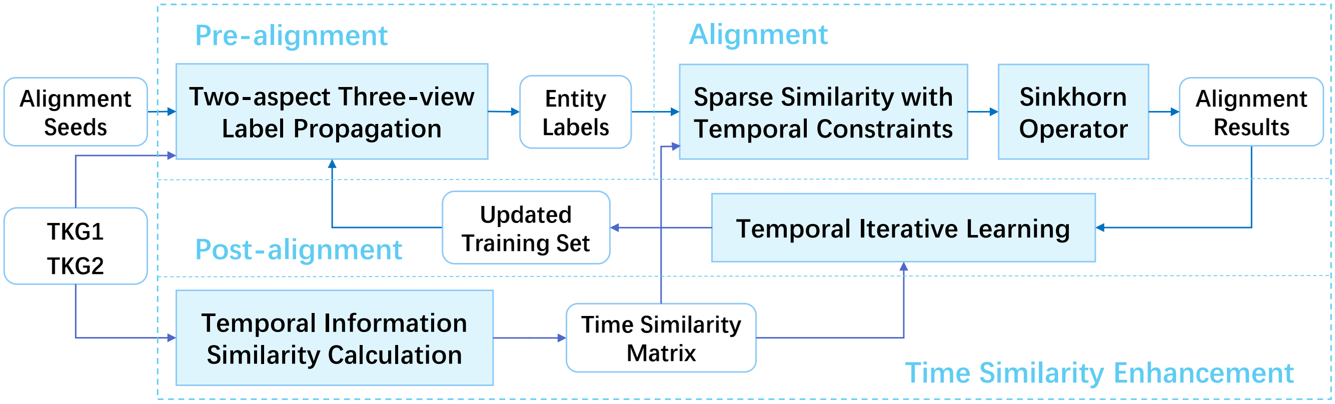
Pre-alignment
Alignment
Alignment Seeds
Two-aspect Three-view Label Propagation
Entity Labels
Sparse Similarity with Temporal Constraints
Sinkhorn Operator
Alignment Results
TKG1
TKG2
Updated Training Set
Temporal Iterative Learning
Post-alignment
Temporal Information Similarity Calculation
Time Similarity Matrix
Time Similarity Enhancement

Figure 2: The framework of LightTEA.

2.2 Time-aware Entity Alignment

Recently, the research on TKGs has developed rapidly, and TEA methods have also sprung up. TEA-GNN [Xu *et al.*, 2021] is the first TEA method using temporal information in KGs. It introduces a time-aware attention mechanism to learn entity embedding based on GNN and constructs five datasets extracted from ICEWS, YAGO3, and Wikidata to evaluate the model. TREA [Xu *et al.*, 2022] utilizes a temporal relational attention mechanism to integrate relational and temporal features of entities from the neighborhood and adopts the margin-based multi-class log-loss (MML) with sequential time regularizer to train the model. TREA is the most efficient TEA model since the MML can achieve fast convergence. STEA [Cai *et al.*, 2022] presents a simple GNN to learn the entity embeddings and uses a temporal information matching mechanism to calculate the time similarity of entities. Then it balances the time similarity and embedding similarity of entities to obtain the equivalent pairs.

By using the temporal information, the TEA methods achieve better performance. However, since the experimental datasets are much smaller than real-world KGs, these methods focus on improving performance and ignore efficiency. Although TREA utilizes MML to accelerate convergence, all TEA methods are based on GNN [Wu *et al.*, 2022] and suffer from scalability issues. So we propose an effective and efficient TEA framework to address the limitations.

3 Problem Formulation

A TKG can be formalized as $G = (E, R, T, Q)$, where E, R and T are the sets of entities, relations, and timestamps respectively, $Q \subset E \times R \times E \times T$ denotes the set of quadruples. A quadruple stores the real-world fact and can be presented as (e_h, r, e_t, τ), where $e_h, e_t \in E$. Given two TKGs $G_1 = (E_1, R_1, T_1, Q_1)$, $G_2 = (E_2, R_2, T_2, Q_2)$, and alignment seeds set $S = \{(e_{1_i}, e_{2_j})|e_{1_i} \in E_1, e_{2_j} \in E_2, e_{1_i} \equiv e_{2_j}\}$ where $\equiv$ denotes equivalence. EA task aims to find new equivalent entity pairs between G_1 and G_2 based on S. C is the set of reference entity pairs used for evaluation. Specifically, a uniform time set $T^* = T_1 \cup T_2$ is constructed by merging the timestamps in the two time sets. Therefore, the two TKGs can be renewed as $G_1 = (E_1, R_1, T^*, Q_1)$ and $G_2 = (E_2, R_2, T^*, Q_2)$ sharing the same set of timestamps.

4 The Proposed Approach

The LightTEA framework can be described as three phases with time similarity enhancement. (1) For *pre-alignment* phase, we use two-aspect three-view label propagation to learn the labels of entities. (2) For *alignment* phase, we first compute the sparse similarity with temporal constraints, then translate the EA problem to the assignment problem and utilize the Sinkhorn operator to resolve it. (3) For *post-alignment* phase, we adopt a temporal iterative learning strategy to enhance EA. Figure 2 shows the framework of LightTEA.

4.1 Two-aspect Three-view Label Propagation

Inspired by LightEA [Mao *et al.*, 2022b], we extend the three-view label propagation to two-aspect (relational-aspect and temporal-aspect) three-view label propagation, which can enhance EA with temporal information while not increase the time and space complexity.

Specifically, the TKG requires a four-order tensor $\mathscr{A} \in \mathbb{R}^{|E|\times|E|\times|R|\times|T|}$ to fully describe the adjacency relations. As shown in Figure 3(a), we regard TKG as two three-order tensor $\boldsymbol{\mathcal{A}}^R \in \mathbb{R}^{|E|\times|E|\times|R|}$ and $\boldsymbol{\mathcal{A}}^T \in \mathbb{R}^{|E|\times|E|\times|T|}$, there are five-view in TKGs: $\boldsymbol{A}^{ER} \in \mathbb{R}^{|E|\times|R|}$, $\boldsymbol{A}^{RE} \in \mathbb{R}^{|R|\times|E|}$, $\boldsymbol{A}^{EE} \in \mathbb{R}^{|E|\times|E|}$, $\boldsymbol{A}^{TE} \in \mathbb{R}^{|T|\times|E|}$, and $\boldsymbol{A}^{ET} \in \mathbb{R}^{|E|\times|T|}$, which represent the adjacency relations from head entity to relation, relation to tail entity, head entity to tail entity, timestamps to tail entity, head entity to timestamps, respectively. Then, we use the relational-aspect and temporal-aspect three-views label propagation to update the labels of entities, relations, and timestamps (as shown in Figure 3(b)).

The relational-aspect three-views label propagation can be presented as follows:

$$\boldsymbol{L}_e^{(n+1)} = \boldsymbol{A}^{EE} \cdot \boldsymbol{L}_e^{(n)} + \boldsymbol{A}^{ER} \cdot \boldsymbol{L}_r^{(n)} \tag{1}$$

$$\boldsymbol{L}_r^{(n+1)} = \boldsymbol{A}^{RE} \cdot \boldsymbol{L}_e^{(n)} \tag{2}$$

where $\boldsymbol{L}_e \in \mathbb{R}^{|E|\times|d|}$, $\boldsymbol{L}_r \in \mathbb{R}^{|R|\times|d|}$ are label matrixs of entities and relations, $\cdot$ means the dot product. Following LightEA, we regard each pair of alignment entities as an independent class, and independently sample random vectors on the d-dimensional hyper-sphere to approximate the one-hot label vectors for representing the alignment seeds, so we

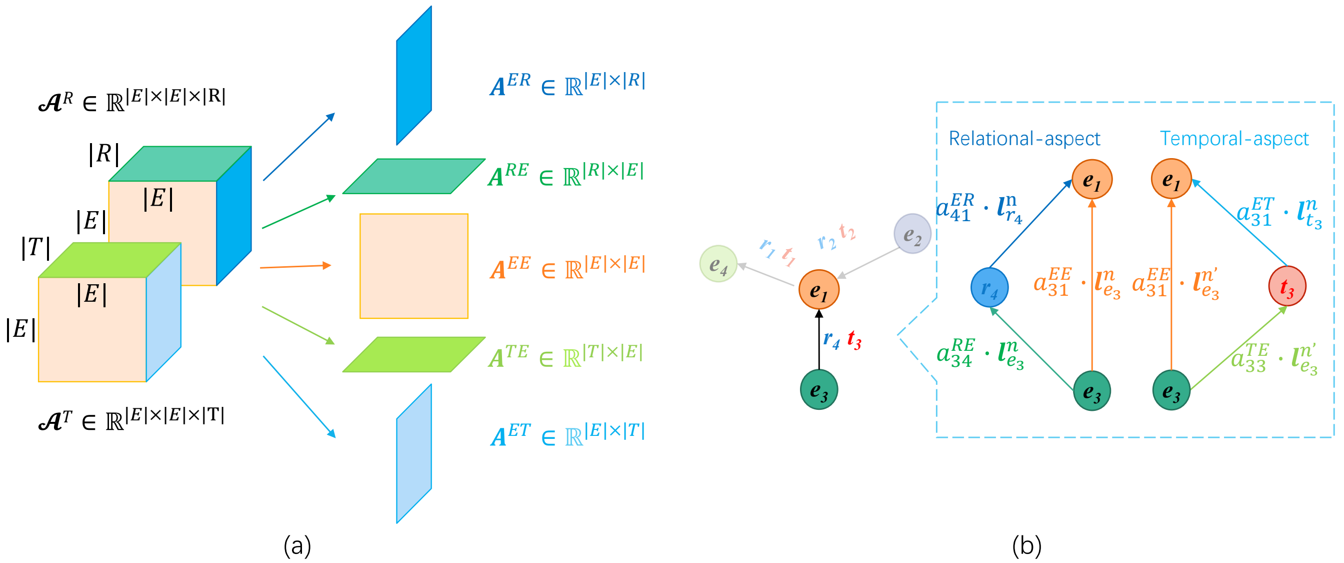
$\mathcal{A}^R \in \mathbb{R}^{|E|\times|E|\times|R|}$
$|R|$
$|E|$
$|E|$
$|T|$
$|E|$
$|E|$
$\mathcal{A}^T \in \mathbb{R}^{|E|\times|E|\times|T|}$
$\boldsymbol{A}^{ER} \in \mathbb{R}^{|E|\times|R|}$
$\boldsymbol{A}^{RE} \in \mathbb{R}^{|R|\times|E|}$
$\boldsymbol{A}^{EE} \in \mathbb{R}^{|E|\times|E|}$
$\boldsymbol{A}^{TE} \in \mathbb{R}^{|T|\times|E|}$
$\boldsymbol{A}^{ET} \in \mathbb{R}^{|E|\times|T|}$
(a)
Relational-aspect
Temporal-aspect
e_4
$r_1\ t_1$
$r_2\ t_2$
e_2
e_1
$r_4\ t_3$
e_3
$a_{41}^{ER} \cdot \boldsymbol{l}_{r_4}^{n}$
e_1
r_4
$a_{31}^{EE} \cdot \boldsymbol{l}_{e_3}^{n}$
e_3
$a_{34}^{RE} \cdot \boldsymbol{l}_{e_3}^{n}$
e_1
$a_{31}^{EE} \cdot \boldsymbol{l}_{e_3}^{n'}$
$a_{31}^{ET} \cdot \boldsymbol{l}_{t_3}^{n}$
t_3
e_3
$a_{33}^{TE} \cdot \boldsymbol{l}_{e_3}^{n'}$
(b)

Figure 3: Illustrations of Two-aspect Three-view Label propagation. a) Regarding TKG as two three-order tensors in terms of relational and temporal aspects, which include five types of adjacency relationships. b) Label propagation from e_3 to e_1 using the two-aspect three-view approach.

set $\boldsymbol{l}_{e_i}^{(0)} = \boldsymbol{l}_{e_j}^{(0)} = random(d)\ \forall(e_i, e_j) \in S$. For the entities in the candidate set C, we initialize the label of them as $\boldsymbol{l}_{e_k}^{(0)} = \boldsymbol{l}_{e_l}^{(0)} = 0\ \forall(e_k, e_l) \in C$, and the initial label matrix of relations is also set to all-zero $\boldsymbol{L}_r^{(0)} = 0$.

The final label of entity e_i in the relational aspect is the concatenation of label vectors in all steps:

$$\boldsymbol{l}_{e_i}^{r} = [\boldsymbol{l}_{e_i}^{(0)} || \boldsymbol{l}_{e_i}^{(1)} || ... || \boldsymbol{l}_{e_i}^{(n-1)} || \boldsymbol{l}_{e_i}^{(n)}] \tag{3}$$

The temporal-aspect three-views label propagation can be expressed as follows:

$$\boldsymbol{L}_e^{(n+1)'} = \boldsymbol{A}^{EE} \cdot \boldsymbol{L}_e^{(n)'} + \boldsymbol{A}^{ET} \cdot \boldsymbol{L}_t^{(n)} \tag{4}$$

$$\boldsymbol{L}_t^{(n+1)} = \boldsymbol{A}^{TE} \cdot \boldsymbol{L}_e^{(n)'} \tag{5}$$

where $\boldsymbol{L}_e^{'} \in \mathbb{R}^{|E|\times|d|}$, and the initializtion label matrix of $\boldsymbol{L}_e^{(0)'}$ is the same as $\boldsymbol{L}_e^{(0)}$. $\boldsymbol{L}_t \in \mathbb{R}^{|T|\times|d|}$ is label matrix of timestamps and is initialized to $\boldsymbol{L}_t^{(0)} = 0$.

We concatenate the label vectors of all steps as the final label of entity e_i in the temporal aspect:

$$\boldsymbol{l}_{e_i}^{t} = [\boldsymbol{l}_{e_i}^{(0)'} || \boldsymbol{l}_{e_i}^{(1)'} || ... || \boldsymbol{l}_{e_i}^{(n-1)'} || \boldsymbol{l}_{e_i}^{(n)'}] \tag{6}$$

Finally, the label of entity e_i is the balanced result of the two aspect labels:

$$\boldsymbol{l}_{e_i} = (1 - \alpha) \times \boldsymbol{l}_{e_i}^{r} + \alpha \times \boldsymbol{l}_{e_i}^{t} \tag{7}$$

where α is a hyper-parameter to balance the label of relational aspect and temporal aspect.

4.2 Temporal Information Similarity Calculation

Recent research([Cai *et al.*, 2022]) suggests that the temporal information similarity can enhance the EA between TKGs, so we calculate the time similarity matrix and use it in the alignment and post-alignment phase.

First, we collect all timestamps of entities. Then we calculate the time similarity $s_{e_i e_j}^{t}$ of e_i and e_j by the following:

$$s_{e_i e_j}^{t} = \frac{2 \times v}{k + q} \tag{8}$$

where v denote the number of same items of e_i and e_j, k and q are the numbers of timestamps of e_i and e_j, respectively.

4.3 Sparse Similarity with Temporal Constraints

After obtaining the labels of entities, we calculate the similarity of entities by their labels in a sparse way with temporal constraints.

Early studies [Chen *et al.*, 2017; Wang *et al.*, 2018; Xu *et al.*, 2021] calculate the embedding similarity (Cosine, Euclidean, or Manhattan distance) of all entities to find the equivalent entity pairs. The calculation complexity is $O(|E|^2 d)$, and the space complexity of the similarity matrix is $O(|E|^2)$. LightEA [Mao *et al.*, 2022b] notices that the similarity of many entities is very small and infinitely close to zero. Even if these smaller values are removed initially, it does not significantly affect the alignment results. Therefore, instead of calculating the similarities between all entities, we only retrieve the *top-k* nearest neighbors for each entity by approximate nearest neighbor (ANN) algorithms [1]. It only takes several seconds to find the *top-k* nearest neighbors, and the space complexity of the sparse similarity matrix is $O(|E|k), k \ll |E|$.

Different from LightEA, we use the time similarity matrix as a constraint to obtain the sparse similarity of entities. Specifically, we get the sparse similarity matrix $S^l \in \mathbb{R}^{|E|\times k}$ by the ANN algorithms with the labels of entities and find the related time similarity of these entities, the final sparse

[1] In LightTEA, we use the FAISS framework [Johnson *et al.*, 2021] for approximate vector retrieval

Datasets	$\|E_1\|$	$\|E_2\|$	$\|R_1\|$	$\|R_2\|$	$\|T^*\|$	$\|Q_1\|$	$\|Q_2\|$	$\|S\|$	$\|C\|$
DICEWS-1K/200	9,517	9,537	247	246	4,017	307,552	307,553	1,000/200	7,566/8,366
YAGO-WIKI50K-5K/1K	49,629	49,222	11	30	245	221,050	317,814	5,000/1,000	44,172/48,172

Table 1: Statistics of DICEWS-1K/200 and YAGO-WIKI50K-5K/1K. $|\cdot|$ denotes the numbers.

similarity of entities is as follows:

$$\boldsymbol{S}' = (1-\beta) \times \boldsymbol{S}^l + \beta \times \boldsymbol{S}^t \tag{9}$$

where β is the hyper-parameter for balancing the label similarity and time similarity of entities.

4.4 Sinkhorn Operator

Existing TEA methods [Xu *et al.*, 2021; Xu *et al.*, 2022; Cai *et al.*, 2022] simply calculate the similarity of the entities to obtain the equivalent entity pairs in the alignment phase. To further improve the effectiveness of the model, we adopt the Sinkhorn operator in the alignment phase to enhance EA.

LightEA [Mao *et al.*, 2022b] regards the EA problem as a one-to-one assignment problem to improve the performance of EA. The goal of the assignment problem is to find the optimal strategy to obtain the maximum profit and can be formulated as follows:

$$\underset{P \in \mathcal{P}_{|E|}}{\arg\max} \, \langle \boldsymbol{P}, \, \boldsymbol{S} \rangle_F \tag{10}$$

where $\boldsymbol{P}$ is a permutation matrix that has exactly one entry of 1 in each row and each column and 0s elsewhere. $\mathcal{P}_{|E|}$ is the set of $|E| \times |E|$ permutation matrices, $\boldsymbol{S} \in \mathbb{R}^{|E| \times |E|}$ is the similarity matrix of entities, and $\langle \cdot \rangle_F$ represents the Frobenius inner product.

The Sinkhorn operator proposes a fast and completely parallelizable algorithm for the assignment problem. It iteratively normalizes rows and columns of the similarity matrix:

$$\begin{aligned} Sinkhorn^0(\boldsymbol{S}) &= exp(\boldsymbol{S}), \\ Sinkhorn^m(\boldsymbol{S}) &= \mathcal{N}_c(\mathcal{N}_r(Sinkhorn^{m-1}(\boldsymbol{S}))), \\ Sinkhorn(\boldsymbol{S}) &= \lim_{m \to \infty} Sinkhorn^m(\boldsymbol{S}) \end{aligned} \tag{11}$$

where $\mathcal{N}_r(\boldsymbol{S}) = \boldsymbol{S} \oslash (\boldsymbol{S} 1_N 1_N^\top)$, $\mathcal{N}_c(\boldsymbol{S}) = \boldsymbol{S} \oslash (1_N 1_N^\top \boldsymbol{S})$ are the row and column-wise normalization operators, $\oslash$ denotes the element-wise division, 1_N represents a column vector of ones, and m is the iterations. The time complexity of Sinkhorn is $O(m|E|^2)$.

We also regard the EA problem as an assignment problem and employ the Sinkhorn operator to obtain the approximate solution:

$$\begin{aligned} &\underset{\boldsymbol{P} \in \mathcal{P}_{|E|}}{\arg\max} \, \langle \boldsymbol{P}, \, \boldsymbol{S}' \rangle_F \\ &= \underset{t \to 0^+}{lim} \, Sinkhorn(\boldsymbol{S}'/t) \end{aligned} \tag{12}$$

where $\boldsymbol{S}' \in \mathbb{R}^{|E| \times k}$ is the sparse similarity matrix with temporal constraints, which is calculated by equation (9), t is the temperature. In this way, the performance of EA improves significantly, and the computational complexity drops to $O(m|E|k)$.

4.5 Temporal Iterative Learning

Iterative Learning is proposed by BootEA [Sun *et al.*, 2018] to address the problem of fewer alignment seeds and enhance the performance of EA. It is also called a semi-supervised alignment strategy which continuously selects possible entity pairs to augment the training data through an iterative method in the post-alignment phase.

To promote EA, we adopt temporal iterative learning. Different from STEA [Cai *et al.*, 2022], which adopts a bi-directional iterative strategy, we simply choose the entity and its nearest neighbor whose similarity value is greater than the threshold [2] as the alignment pairs and add them to the training set for the next iteration.

5 Experiments

We conduct the experiments on a workstation with a GeForce RTX 3090 GPU and an AMD EPYC 7502 32-Core Processor CPU, 128GB memory. The codes and datasets will be available on GitHub [3].

5.1 Datasets

To comprehensively evaluate the effectiveness and efficiency of the proposed model, we experiment on two widely used public datasets. The statistics of these datasets are listed in Table 1.

(1) **DICEWS-1K/200** [Xu *et al.*, 2021] is constructed from the event knowledge graph ICEWS05-15 which contains events during 2005 to 2015. It consists of two subsets with different alignment seeds 1K/200.

(2) **YAGO-WIKI50K-5K/1K** [Xu *et al.*, 2021] extracts the equivalent entities with temporal information from YAGO and Wikidata. There are two subsets, one with 5K alignment seeds and the other with 1K.

5.2 Baselines

In the experiments, we compared our proposed model with two categories of advanced entity alignment methods:

(1) **Supervised methods**: JAPE [Sun *et al.*, 2017], AlignE [Sun *et al.*, 2018], GCN-Align [Wang *et al.*, 2018], MRAEA [Mao *et al.*, 2020a], LightEA* [Mao *et al.*, 2022b], TEA-GNN [Xu *et al.*, 2021], TREA [Xu *et al.*, 2022], and STEA* [Cai *et al.*, 2022]. JAPE, AlignE, GCN-Align, MRAEA, and LightEA* are EA methods, TEA-GNN, TREA, and STEA* are TEA methods.

(2) **Semi-supervised methods**: BootEA [Sun *et al.*, 2018], RREA [Mao *et al.*, 2020b], LightEA [Mao *et al.*, 2022b], STEA [Cai *et al.*, 2022]. BootEA, RREA, and LightEA are EA methods, STEA is a TEA method.

[2] The threshold is 0.8 in LightTEA.

[3] https://github.com/lcai2/LightTEA

		DICEWS-1K			DICEWS-200			YAGO-WIKI50K-5K			YAGO-WIKI50K-1K		
Categories	Methods	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
Supervised	MTransE	.150	.101	.241	.104	.067	.175	.332	.242	.477	.033	.012	.067
	JAPE	.198	.144	.298	.138	.098	.210	.345	.271	.488	.157	.101	.262
	AlignE	.593	.508	.751	.303	.222	.457	.800	.756	.883	.618	.565	.714
	GCN-Align	.291	.204	.466	.231	.165	.363	.581	.512	.711	.279	.217	.398
	MRAEA	.745	.675	.870	.564	.476	.733	.848	.806	.913	.685	.623	.801
	LightEA*	.833	.785	.918	.779	.721	.878	.960	.948	.979	.902	.878	.945
	TEA-GNN	.911	.887	.947	.902	.876	.941	.909	.879	.961	.775	.723	.871
	TREA	.933	.914	.966	.927	.910	.960	.958	.940	.989	.885	.840	.937
	STEA*	.941	.928	.960	.941	.927	.961	.954	.935	.986	.916	.887	.966
	LightTEA*	.959	.952	.970	.955	.949	.966	.990	.986	.997	.977	.969	.989
	p-value	2e-7	3e-7	2e-6	5e-7	1e-7	5e-5	6e-9	1e-9	7e-7	2e-9	5e-10	4e-8
Semi-supervised	BootEA	.670	.598	.796	.614	.546	.737	-	-	-	-	-	-
	RREA	.840	.795	.917	.823	.773	.911	.913	.887	.955	.870	.836	.929
	LightEA	.875	.838	.936	.878	.842	.937	.963	.951	.980	.951	.938	.970
	STEA	.954	.945	.967	.954	.943	.968	.974	.961	.992	.962	.943	.989
	LightTEA	**.960**	**.954**	**.970**	**.963**	**.958**	**.972**	**.992**	**.988**	**.997**	**.992**	**.988**	**.997**
	p-value	2e-4	2e-5	4e-4	4e-6	2e-6	5e-5	2e-7	2e-8	5e-5	1e-8	2e-9	2e-6

Table 2: Experimental results on all datasets. - means the results are not obtained. The best results are written in bold. Underline indicate the sub-optimal results. The p-value is the result of one sample t-test between LightTEA*/LightTEA and their corresponding strong baselines.

The main experiment results of these methods reported in the paper are from STEA (SOTA TEA method), except the latest SOTA EA method LightEA. The experiments of LigthEA are implemented on its open-source code. For a fair comparison, LightTEA has two corresponding versions: (1) LightTEA* is the supervised version. (2) LightTEA is the semi-supervised version with iterative learning.

5.3 Settings

Evaluation Metrics

Following the conventions, we adopt mean reciprocal rank (MRR) and Hits@k ($k = 1, 10$) as the evaluation metrics. MRR reports the average reciprocal of the ranks, and Hits@k calculates the proportion of correct alignment pairs whose rank is not greater than k. In particular, Hits@1 represents the accuracy of the results, which is the most important indicator. The higher the MRR and Hits@k, the better the performance.

Implementation Details

We use the fixed training set and validation set provided by TEA-GNN. The hyper-parameters are set as follows: the dimension of the hyper-sphere is $d = 512$, the factor α is set to 0.6 for DICEWS and 0.5 for YAGO-WIKI50K to balance the relational-aspect and temporal-aspect label propagation, the factor β is set to 0.4 for balancing the label similarity and time similarity. Following LightEA, we use rounds $n = 2$ for two-aspect three-view label propagation, retrieve the top-$k = 500$ nearest neighbors for sparse similarity, set the number of iterations of the Sinkhorn operator to $m = 15$, and set the temperature to $t = 0.05$. The reported performances are the averages of five independent runs.

5.4 Main Experiments

Table 2 lists the main experimental results of our proposed model and all baselines on the four datasets. Among the supervised methods, LightTEA* has significant improvements in all metrics. Compared to the SOTA supervised TEA method STEA*, LightTEA* improves the Hits@1 by 2.61%, 2.33%, 5.45%, and 9.22%, respectively. In the semi-supervised methods, LightTEA outperforms all the baselines on all datasets across all metrics. The improvements of Hits@1 compared with STEA are 0.93%, 1.57%, 2.79%, and 4.77%, respectively. TEA methods perform better in these two categories of methods than EA methods by using temporal information. The semi-supervised methods achieve better performance than supervised methods, which indicates the effectiveness of temporal iterative learning. Without iteration, LightTEA* outperforms all baselines except for STEA's Hits@10 on DICEWS-200 with tiny gap 0.002. The high performance of LightTEA* demonstrates that the two-aspect three-view label propagation, the sparse similarity with temporal constraints, and the Sinkhorn operator are effective in promoting EA. We conduct one sample t-test between LightTEA(/LightTEA*) and their corresponding strong baselines. All the p-value $<< 0.01$ indicates that our proposed model significantly outperforms all baselines.

5.5 Ablation Study

We conduct an ablation experiment on DICEWS-200 and YAGO-WIKI50K-1K to investigate the contributions of three critical components of LightTEA: (1) Temporal-aspect Label Propagation (TLP), (2) Temporal Constraints (TC), and (3) Sinkhorn Operator (SO).

	DICEWS-200			YAGO-WIKI50K-1K		
Methods	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
LightTEA	**.963**	**.958**	**.972**	**.992**	**.988**	**.997**
w/o TLP	.956	.950	.965	.989	.985	.994
w/o TC	.954	.947	.965	.987	.982	.995
w/o SO	.942	.933	.954	.962	.947	.986

Table 3: Ablation study of LightTEA on DICEWS-200 and YAGO-WIKI50K-1K. w/o means without.

Categories	Methods	**DICEWS-200**	**YAGO-WIKI50K-1K**
supervised	TEA-GNN	1,547	5,042
	TREA	128	2,655
	LightTEA*	5	24
semi-supervised	STEA	578	5,428
	LightTEA	12	85

Table 4: Time costs of TEA methods on DICEWS-200 and YAGO-WIKI50K-1K (seconds).

# Dimension	64	128	256	**512**	1024	2048
LightTEA*	0.943	0.947	0.947	**0.949**	0.949	0.949
LightTEA	0.948	0.954	0.956	**0.958**	0.958	0.958

Table 5: Hits@1 scores with different dimensions on DICEWS-200.

As reported in Table 3, without temporal-aspect label propagation (w/o TLP), the performance of LightTEA drops a little, which indicates the effectiveness of the temporal-aspect label propagation. When removing the temporal constraints (w/o TC), the underperformance of LightTEA implies the effectiveness of the temporal constraints. Without the Sinkhorn operator (w/o SO), the performance of LightTEA drops significantly, demonstrating the contribution of utilizing the Sinkhorn operator in the alignment phase to enhance EA.

5.6 Efficiency Studys

Table 4 shows the overall time costs on DICEWS-200 and YAGO-WIKI50K-1K datasets by TEA methods from data loading to evaluation. The results of TREA are from [Xu *et al.*, 2022] since it doesn't provide the source codes. The other results are obtained by directly running the source codes provided by the author.

It can be seen from Table 4, in the supervised time-aware methods, TREA costs less time than TEA-GNN with MML, which speed up the convergence. LightTEA* takes much less time than the above two methods. The time costs of LigthTEA* are 5 seconds and 24 seconds on the two datasets, respectively, which are only 3.91% and 0.90% of TREA. In the semi-supervised methods, the time costs of LightTEA are 12 seconds and 85 seconds, which are 2.08% and 1.70% of STEA on the two datasets, respectively. The time consumed by LightTEA is no more than 10% of the most efficient methods TREA. By using temporal iterative learning, LightTEA increases the time consumption while improving performance compared to LightTEA*. The high efficiency of LightTEA indicates that it could be applied to large-scale datasets for EA between TKGs.

5.7 Hyper-parameter Analysis

We conduct experiments on the following hyper-parameters to investigate their effect on the performance of LightTEA.

(1) The dimension of hyper-sphere d. We select the dimension in the set $\{64, 128, 256, 512, 1024, 2048\}$ and conduct the experiments. The Hits@1 scores with different dimensions on DICEWS-200 are shown in Table 5.

From the table we can see that as the dimension increases, the Hits@1 score for both LightTEA and LightTEA* gradually improves until it reaches their best performance at the dimension of 512. Even if the dimension is further increased, the performance remains unchanged. This indicates that when the dimension of the label vector is greater than 512, it only increases memory consumption and provides no benefit to performance.

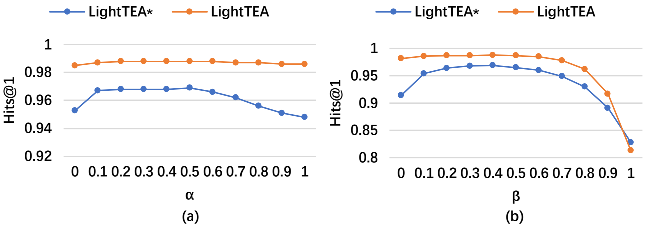

Figure 4: Experimental results with different α and β on YAGO-WIKI50K-1K.

(2) The balance factors α and β. We set the two factors in range 0$\sim$1 with interval 0.1 to investigate the impact of different values. The experimental results with different α and β on YAGO-WIKI50K-1K are shown in Figure 4.

α balances the relational-aspect label propagation (RLP) and temporal-aspect label propagation (TLP). Figure 4(a) shows the Hits@1 of LightTEA* and LightTEA with different α. Due to the use of temporal iterative learning, the performance of LightTEA does not change significantly with different α. The Hits@1 curve of LightTEA* shows that the result of combining RLP and TLP with appropriate weight ($\alpha = 0.5$) is better than using only one of them ($\alpha = 0$ or $\alpha = 1$).

β is used to balance the label similarity and time similarity. As shown in Figure 4(b), with the increase of β, the Hits@1 first increases slowly (reaching the maximum value when $\beta = 0.4$), and then decreases rapidly. When $\beta = 1$, Hits@1 of LightTEA with temporal iterative learning are lower than LightTEA*, it indicates that only using time similarity to generate the possible alignment pairs and add them to the training set for the next iteration will hurt the performance of the model.

6 Conclusion

Existing EA methods ignore the temporal information in KGs, which may wrongly align similar entities. The TEA methods lack scalability since the inherent defect of GNN. To address the limitations, we proposes an effective and efficient TEA framework that consists of four important components: (1) Two-aspect Three-view Label Propagation, (2) Sparse Similarity with Temporal Constraints, (3) Sinkhorn Operator, and (4) Temporal Iterative Learning. These modules work collaboratively to enhance the model's performance while reducing time consumption.

Extensive experiments on public datasets indicate that the proposed model significantly outperforms the SOTA methods. The time consumed by the model is only dozens of seconds at most, no more than 10% of the most efficient TEA methods, which denotes that our model has high efficiency and can be applied to large-scale TKGs.

Acknowledgements

We appreciate the support from National Natural Science Foundation of China with the Main Research Project on Machine Behavior and Human-Machine Collaborated Decision Making Methodology (72192820 & 72192824), Pudong New Area Science & Technology Development Fund (PKX2021-R05), Science and Technology Commission of Shanghai Municipality (22DZ2229004), and Shanghai Trusted Industry Internet Software Collaborative Innovation Center.

References

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NeurIPS*, pages 2787–2795, Lake Tahoe, Nevada, United States, December 2013. Curran Associates.

[Cai *et al.*, 2022] Li Cai, Xin Mao, Meirong Ma, Hao Yuan, Jianchao Zhu, and Man Lan. A simple temporal information matching mechanism for entity alignment between temporal knowledge graphs. In *COLING*, pages 2075–2086, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[Chen *et al.*, 2017] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *IJCAI*, pages 1511–1517, Melbourne, Australia, August 2017. ijcai.org.

[Dietz *et al.*, 2018] Laura Dietz, Alexander Kotov, and Edgar Meij. Utilizing knowledge graphs for text-centric information retrieval. In *SIGIR*, pages 1387–1390, Ann Arbor, MI, USA, July 2018. ACM.

[Guo *et al.*, 2022] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A survey on knowledge graph-based recommender systems. *IEEE Trans. Knowl. Data Eng.*, 34(8):3549–3568, August 2022.

[Hoffart *et al.*, 2013] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61, January 2013.

[Johnson *et al.*, 2021] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(3):535–547, July 2021.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, Toulon, France, April 2017. OpenReview.net.

[Lan *et al.*, 2021] Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. A survey on complex knowledge base question answering: Methods, challenges and solutions. In *IJCAI*, pages 4483–4491, Virtual Event / Montreal, Canada, August 2021. ijcai.org.

[Lautenschlager *et al.*, 2015] Jennifer Lautenschlager, Steve Shellman, and Michael Ward. ICEWS Event Aggregations, 2015.

[Leetaru and Schrodt, 2013] Kalev Leetaru and Philip A Schrodt. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49, San Francisco, California, USA, April 2013. Citeseer.

[Mao *et al.*, 2020a] Xin Mao, Wenting Wang, Huimin Xu, Man Lan, and Yuanbin Wu. MRAEA: an efficient and robust entity alignment approach for cross-lingual knowledge graph. In *WSDM*, pages 420–428, Houston, TX, USA, February 2020. ACM.

[Mao *et al.*, 2020b] Xin Mao, Wenting Wang, Huimin Xu, Yuanbin Wu, and Man Lan. Relational reflection entity alignment. In *CIKM*, pages 1095–1104, Virtual Event / Ireland, October 2020. ACM.

[Mao *et al.*, 2022a] Xin Mao, Meirong Ma, Hao Yuan, Jianchao Zhu, Zongyu Wang, Rui Xie, Wei Wu, and Man Lan. An effective and efficient entity alignment decoding algorithm via third-order tensor isomorphism. In *ACL*, pages 5888–5898, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[Mao *et al.*, 2022b] Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. Lightea: A scalable, robust, and interpretable entity alignment framework via three-view label propagation. In *EMNLP*, page 825–838, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[Sun *et al.*, 2017] Zequn Sun, Wei Hu, and Chengkai Li. Cross-lingual entity alignment via joint attribute-preserving embedding. In *ISWC*, volume 10587 of *Lecture Notes in Computer Science*, pages 628–644, Vienna, Austria, October 2017. Springer.

[Sun *et al.*, 2018] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, pages 4396–4402, Stockholm, Sweden, July 2018. ijcai.org.

[Sun *et al.*, 2020] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proc. VLDB Endow.*, 13(11):2326–2340, June 2020.

[Vrandecic and Krötzsch, 2014] Denny Vrandecic and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, September 2014.

[Wang *et al.*, 2018] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. Cross-lingual knowledge graph alignment via graph convolutional networks. In *EMNLP*, pages 349–357, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[Wu *et al.*, 2022] Lingfei Wu, Peng Cui, Jian Pei, and Liang Zhao. *Graph Neural Networks: Foundations, Frontiers, and Applications*. Springer Singapore, Singapore, 2022.

[Xu *et al.*, 2021] Chengjin Xu, Fenglong Su, and Jens Lehmann. Time-aware graph neural network for entity alignment between temporal knowledge graphs. In *EMNLP*, pages 8999–9010, Virtual Event / Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[Xu *et al.*, 2022] Chengjin Xu, Fenglong Su, Bo Xiong, and Jens Lehmann. Time-aware entity alignment using temporal relational attention. In *WWW*, pages 788–797, Virtual Event / Lyon, France, April 2022. ACM.