A Novel Retrieve-Read-Group Paradigm for Open Knowledge Base Canonicalization

Binhan Yang[1], **Wei Shen**[1*], **Han Tian**[1]

[1]DISSec, College of Computer Science, Nankai University, Tianjin, China
{yangbinhan, tianhan}@mail.nankai.edu.cn, shenwei@nankai.edu.cn

Abstract

Noun phrases (NPs) in open knowledge bases (OKBs) are not canonicalized, leading to scattered knowledge that necessitates the exploration of the OKB canonicalization task (i.e., clustering synonymous noun phrases into the same group and assigning them a unique identifier). However, existing OKB canonicalization methods typically adhere to a traditional embedding-centered pipeline, which fails to exploit the direct interaction between NPs for pairwise NP similarity calculations, resulting in suboptimal performance and instead relying extensively on external resources. To address these limitations, we introduce a groundbreaking *retrieve-read-group* paradigm that enables fine-grained pairwise NP similarity calculations by effectively leveraging the direct NP interaction via the reading stage, thereby relieving the reliance on external resources. As an instantiation of this paradigm, we propose DUVK, a novel self-supervised framework that fully integrates the dual-view knowledge involved in OKBs from the relational view and the semantic view. In the retriever component of DUVK, a dual-view cross-training strategy is designed to make two view-specific encoders mutually reinforce each other by capitalizing on the complementary knowledge delivered from both views. Experimental results demonstrate that, even without the need of any external resources, DUVK outperforms all state-of-the-art competitors that rely on such resources.

1 Introduction

Curated knowledge bases (CKBs), which are constructed based on manually predefined ontologies, store comprehensive factual knowledge about real-world entities. Prominent CKBs (e.g., Wikidata (Vrandečić and Krötzsch 2014), DBpedia (Auer et al. 2007), and YAGO (Suchanek, Kasneci, and Weikum 2007)) have proven to be valuable resources for facilitating various knowledge-driven tasks including information retrieval (Xiong, Power, and Callan 2017; Dietz, Kotov, and Meij 2018), question answering (Lukovnikov et al. 2017; Huang et al. 2019) and recommendation system (Wang et al. 2019). As the world evolves, it is crucial to update obsolete knowledge and incorporate new knowledge to better support downstream applications. However, the construction and curation of CKBs heavily rely on predefined ontologies and require extensive human intervention, limiting the adaptability of CKBs to dynamic knowledge.

As an alternative, open information extraction (OIE) (Yates et al. 2007; Etzioni et al. 2011) has experienced a strong momentum for its scalability and unsupervised manner. Without any pre-specified ontology, OIE can automatically extract knowledge in the form of relational triples (i.e., $<$*noun phrase, relation phrase, noun phrase*$>$) from large-scale web text data containing up-to-date information. These extracted OIE triples collectively form large open knowledge bases (OKBs), such as ReVerb (Fader, Soderland, and Etzioni 2011), OPIEC (Gashteovski et al. 2019) and DefIE (Bovi, Telesca, and Navigli 2015). While the automated construction of OKBs enables them to be highly adaptable to new knowledge, it also introduces certain limitations. In contrast to CKBs where each entity is well defined with a unique identifier, noun phrases (NPs) in OKBs are not ***canonicalized***, which means that the factual knowledge associated with a particular real-world entity may be scattered across multiple distinct NPs. For instance, the real-world entity **"Cristiano Ronaldo"** in an OKB may be represented by different NPs like *CR7* or *C. Ronaldo*, which hinders the consolidation of internal knowledge within OKBs and further poses challenges for downstream applications.

To address the issue of scattered knowledge in OKBs, the task of OKB canonicalization (Galárraga et al. 2014; Shen, Yang, and Liu 2022; Vashishth, Jain, and Talukdar 2018; Dash et al. 2021; Liu et al. 2021; Shen, Yang, and Liu 2024) is proposed, which aims to aggregate synonymous noun phrases (e.g., *CR7* and *C. Ronaldo*) into the same group and assign them a unique identifier, beneficial for knowledge integration. Essentially, this task could be viewed as a clustering problem, where the key challenge lies in computing NP similarities (Jain, Murty, and Flynn 1999). Most existing advanced OKB canonicalization solutions (Shen, Yang, and Liu 2022; Vashishth, Jain, and Talukdar 2018; Dash et al. 2021; Shen, Yang, and Liu 2024) adopt an embedding-centered pipeline, which first encodes each NP *individually* into an embedding vector and then calculates pairwise similarities between NPs via a simple metric (e.g., cosine similarity) based on learned embeddings. Nevertheless, these approaches suffer from two major drawbacks:

(1) Indirect NP interaction: The separation between the processes of individual encoding and similarity calcu-

*Wei Shen is the corresponding author.

lation prevents the direct interaction between NPs, which inevitably leads to suboptimal pairwise NP similarities, negatively affecting the overall canonicalization performance.

(2) Resource-dependent: As each NP is encoded individually, these embedding-centered approaches are susceptible to knowledge sparsity. Unlike in CKBs, where entities are typically encoded by integrating multi-perspective internal knowledge (Zhang et al. 2019; Trisedya, Qi, and Zhang 2019; Moon, Jones, and Samatova 2017; Tang et al. 2020), such as attribute values, type information and textual descriptions, the internal knowledge for NPs within OKBs is limited to the form of OIE triples, which does not suffice for learning an accurate and comprehensive embedding for each NP. As a result, existing embedding-centered methods (Shen, Yang, and Liu 2022; Vashishth, Jain, and Talukdar 2018; Dash et al. 2021; Shen, Yang, and Liu 2024) additionally resort to various external resources (Pavlick et al. 2015; Shen et al. 2023) to generate *training seeds* of synonymous NPs as prior knowledge to further refine the embeddings. However, in practice, such external resources are not always accessible and may be very costly to fetch, limiting the adaptability and scalability of these approaches.

To overcome the above drawbacks, it is especially necessary and important to explore the direct interaction between NPs to enhance the pairwise NP similarity calculation, thereby improving the performance of OKB canonicalization and relieving the reliance on external resources. In recent years, some high-capacity models (e.g., BERT-based cross-encoder (Devlin et al. 2019)), which capture the direct interaction by modeling pairwise inputs jointly, have made remarkable success in low-resource tasks like zero-shot entity linking (Wu et al. 2020, 2023) and ad-hoc information retrieval (Dai and Callan 2019; Ma et al. 2021). Armed with these insights, we propose to revolutionize the traditional embedding-centered pipeline with a high-capacity *reader*, which enables more fine-grained similarity calculations to yield more precise similarities between NPs by effectively leveraging the direct interaction between them.

Unfortunately, while the high-capacity reader improves similarity precision, it also introduces intensive computational overhead, making its application over all possible NP pairs infeasible due to the excessive $\mathcal{O}(n^2)$ complexity. Considering that only a few pairs of NPs are synonymous, we follow an intuitive idea to firstly retrieve some relatively similar NP pairs in a light-weight manner, and then utilize the high-capacity reader to conduct elaborate similarity calculations over these retrieved pairs. Through this retrieval mechanism, the obviously non-synonymous NP pairs are filtered out, significantly narrowing down the computational overhead of the high-capacity reader. To this end, we propose a novel *retrieve-read-group* paradigm for OKB canonicalization (see Figure 1 for an overview), which consists of three stages: the *retrieving* stage that efficiently identifies the top-k most similar candidate NPs for each NP, the *reading* stage that derives similarities for retrieved candidate NP pairs via exploiting the direct NP interaction, and the *grouping* stage that takes the pairwise NP similarities output from the *reading* stage to generate the final clustering result. Notably, our paradigm is highly flexible, allowing any suitable technique to be employed to instantiate each stage.

To demonstrate the effectiveness of the new paradigm, we instantiate it as a novel self-supervised framework **DUVK** that integrates **DU**al-**V**iew **K**nowledge in the OIE triples without requiring any external resources. To make full use of the limited internal knowledge within OKBs, we leverage the complementary knowledge by modeling OIE triples from two different views (i.e., the relational view and the semantic view). Specifically, in the *retriever*, we initially encode each NP using two independent view-specific encoders based on each view, and subsequently propose a dual-view cross-training strategy to make the two encoders mutually reinforce each other. For the *reader*, a high-capacity model, i.e., a BERT-based cross-encoder, is employed to conduct high-precision similarity calculations for each candidate NP pair. For the *grouper*, a similarity graph is constructed based on pairwise NP similarities from the reader, and a graph-based markov clustering algorithm is applied over the similarity graph to yield the final clustering result.

Our major contributions can be summarized as follows:

- We propose a novel *retrieve-read-group* paradigm for OKB canonicalization, which enables fine-grained pairwise NP similarity calculations by leveraging the direct interaction between NPs.
- Based on the new paradigm, we develop a self-supervised instantiation DUVK that exploits the internal knowledge within OKBs from both the relational and semantic views, compensating for the absence of external resources.
- In the retriever, we design an innovative dual-view cross-training strategy by harnessing the complementary knowledge delivered from both views to make the two view-specific encoders mutually reinforce each other.
- The experimental results on two public benchmark datasets demonstrate that without the need of any external resources, our instantiation DUVK surpasses all state-of-the-art competitors.

2 Notations and Problem Definition

An OKB can be represented as $\mathcal{K} = (\mathcal{N}, \mathcal{R}, \mathcal{T}^+)$, where $\mathcal{N}$ and $\mathcal{R}$ indicate the sets of noun phrases (NPs) and relation phrases (RPs) respectively, and $\mathcal{T}^+ \subseteq \mathcal{N} \times \mathcal{R} \times \mathcal{N}$ denotes the set of OIE triples within $\mathcal{K}$ in the form of (*subject NP, RP, object NP*), abbreviated as (n^s, r, n^o).

Problem 1 (**OKB Canonicalization**). *Given an OKB, the objective is to group the NPs in $\mathcal{N} = \{n_i\}_{1 \leq i \leq |\mathcal{N}|}$ into distinct subgroups, such that two NPs n_i and $n_{i'}$ are assigned to the same subgroup if and only if they are synonymous.*

3 Paradigm

As depicted in Figure 1, considering an example OKB with five distinct NPs (i.e., n_1 to n_5), our proposed paradigm would consecutively execute three fundamental stages to canonicalize these NPs:

Stage 1. Retrieve: Due to the fact that only a few pairs of NPs are synonymous, we firstly retrieve some relatively similar NP pairs in this stage. For each NP, a small subset of similar candidate NPs is found out from the entire pool of NPs within the OKB, aiming to reduce the computational

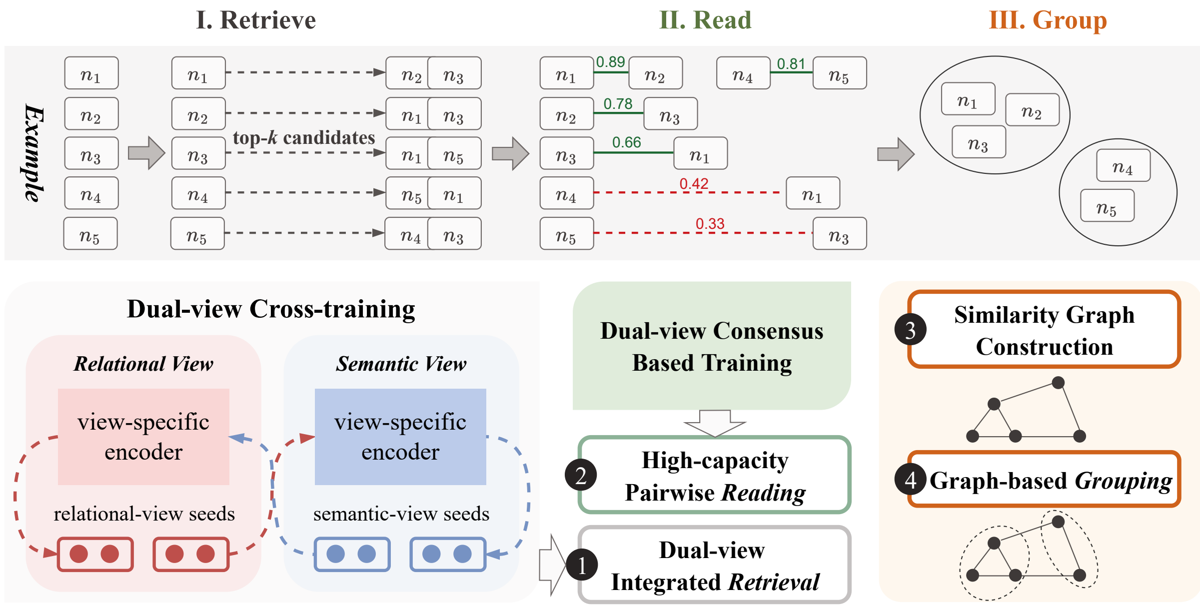

Figure 1: Overview of the retrieve-read-group paradigm and its instantiation as the DUVK framework. The upper part illustrates the input/output flow of each stage, while the lower part presents DUVK with specific components.

complexity in the subsequent reading stage. As illustrated in Figure 1, for NP n_1, NPs n_2 and n_3 exhibit top similarities, so these two NPs are selected as candidates for n_1, resulting in the identification of candidate NP pairs (n_1, n_2) and (n_1, n_3). By retaining only the top-k candidates for each NP, the number of candidate NP pairs is reduced from $|\mathcal{N}| \times (|\mathcal{N}| - 1)$ to $|\mathcal{N}| \times k$, where k is a small constant and $k \ll |\mathcal{N}|$. It should be noted that the retrieving stage needs to strike a balance between effectiveness and efficiency, given that the scale of the entire OKB could be extensive.

Stage 2. Read: After obtaining candidate NP pairs, the high-capacity reader is utilized to conduct a fine-grained similarity calculation for each candidate NP pair. As shown in Figure 1, the pairwise similarity between n_1 and n_2 is high (i.e., 0.89), indicating that they are likely to be synonymous. Conversely, n_1 and n_4 demonstrate relatively low pairwise similarity (i.e., 0.42), suggesting that they are likely to represent different entities. Unlike the retrieving stage, the reading stage primarily prioritizes precision, even though the computational cost may be a little expensive.

Stage 3. Group: Finally, NPs within candidate pairs are grouped into distinct clusters based on their pairwise similarities output from the reading stage. In Figure 1, NPs n_1, n_2 and n_3 show high similarities with each other, leading to their inclusion in the same cluster, while NPs n_4 and n_5 exhibit a high similarity with each other but low similarities with the other NPs, thus forming another distinct cluster. It can be seen that the performance of this grouping stage heavily depends on the accuracy of the pairwise NP similarities acquired from the reading stage.

In summary, our proposed retrieve-read-group paradigm departs from the traditional embedding-centered pipeline by facilitating more fine-grained pairwise NP similarity calculations via the reading stage, which could effectively leverage the direct interaction between NPs and consequently improve the overall canonicalization performance. Notably, since our paradigm is abstract, the specific implementation of each stage can be adaptively tailored to meet its unique requirements in terms of both effectiveness and efficiency.

4 Instantiated Framework: DUVK

In this section, we introduce our proposed self-supervised framework DUVK, which is instantiated based on the retrieve-read-group paradigm discussed in Section 3, integrating the complementary knowledge within the OKB from both the relational and semantic views. Figure 1 illustrates the overall architecture of DUVK, consisting of the *retriever*, *reader* and *grouper* components, which are the specific implementations of the three stages respectively. Concretely, in the *retriever*, each NP is embedded using two independent view-specific encoders, which are trained collectively via a novel dual-view cross-training strategy to facilitate mutual reinforcement between them. With the refined embeddings, candidate NP pairs are identified through a dual-view integrated retrieval process (❶ in Figure 1). In the *reader*, we design a dual-view consensus based training approach to fine-tune the high-capacity reader, and then leverage it to perform high-capacity pairwise similarity calculations for each candidate NP pair (❷ in Figure 1). In the *grouper*, a similarity graph is constructed based on pairwise NP similarities derived from the reader (❸ in Figure 1). Subsequently, a graph-based markov clustering algorithm is applied over the generated similarity graph to produce the final clustering result (❹ in Figure 1). We elaborate these three components in the following.

Retriever

Dual-view Cross-training Given that the retrieval process is conducted over considerable NPs, the retriever is expected to be both precise and scalable, so that potential synonymous NPs can be comprehensively detected within tolerable computational costs. To address the scalability concern, we adopt the prevalent dense retrieval solution (Karpukhin et al. 2020) to encode NPs into dense embedding vectors, enabling efficient comparison and retrieval via light-weight vector computations (e.g., cosine similarity). However, as outlined in Section 1, the lack of *training seeds* (i.e., pre-canonicalized NP pairs from external resources) could severely impair the quality of NP embeddings, degrading retrieval accuracy.

To compensate for the lack of supervision from pre-canonicalized training seeds in our framework, we propose to creatively perform **cross-training** of the two independent view-specific encoders, in which one encoder would iteratively derive pseudo training seeds for the other, facilitating the mutual supervision of NP embedding learning between two complementary views: **1) the relational view** and **2) the semantic view**. Next, we will first introduce the relational-view encoder and the semantic-view encoder respectively, followed by an explanation of our cross-view seed generation and iterative training process to make these two view-specific encoders reinforce each other reciprocally.

Relational-view encoder. In OKBs, the relational information about NPs is preserved in the form of OIE triples, where NPs are interconnected via RPs. To capture such relational knowledge, we choose the simple TransE (Bordes et al. 2013) as the ***relational-view encoder*** for projecting NPs into relational-view embedding vectors $Emb_r(\cdot) \in \mathbb{R}^d$ (where d denotes the dimension of NP embeddings), giving prominence to the effect of our overall framework in comparison to other complicated KB embedding models (Wang et al. 2017; Ji et al. 2021). In TransE, for an OIE triple (n^s, r, n^o), the RP r is interpreted as a translation from its subject NP n^s to its object NP n^o with the constraint $Emb_r(n^s) + Emb_r(r) \approx Emb_r(n^o)$. Specifically, TransE utilizes the score function $f(n^s, r, n^o) = \|Emb_r(n^s) + Emb_r(r) - Emb_r(n^o)\|_2$ to measure the plausibility of triple (n^s, r, n^o), and the margin-based loss function is:

$$\mathcal{L}_{rel} = \sum_{\substack{(n^s, r, n^o) \in \mathcal{T}^+ \\ \wedge (n^{s\prime}, r, n^{o\prime}) \in \mathcal{T}^-}} [\gamma + f(n^s, r, n^o) - f(n^{s\prime}, r, n^{o\prime})]_+$$

where $[\cdot]_+ = \max(0, \cdot)$, and $\gamma > 0$ is a margin hyperparameter. $\mathcal{T}^-$ denotes the set of negative relational triples, which are generated by replacing the subject NP or object NP of a positive relational triple in $\mathcal{T}^+$ with a random NP in $\mathcal{N}$.

Semantic-view encoder. In essence, an OIE triple (e.g., $<$ *C. Ronaldo, won, Ballon d'Or* $>$) can be viewed as a condensed representation of its original sentence (e.g., *"C. Ronaldo won the prestigious Ballon d'Or in 2014."*), containing inherent semantic information that is beneficial for the canonicalization task, as synonymous NPs typically exhibit similar semantics. In light of this, we employ the basic BERT (Devlin et al. 2019) as the ***semantic-view encoder*** to map NPs into semantic-view embedding vectors $Emb_s(\cdot) \in \mathbb{R}^d$, by capturing their semantic knowledge. Concretely, given an NP n^s which is the subject NP of m distinct OIE triples $\{(n^s, r_i, n^o_i)\}_{1 \le i \le m} \subset \mathcal{T}^+$, we first construct its input sequence for BERT as:

$$Seq(n^s) = n^s \;\texttt{[TRI]}\; n^s \, r_1 \, n^o_1 \;\ldots\; \texttt{[TRI]}\; n^s \, r_m \, n^o_m \quad (1)$$

in which `[TRI]` is a special token for specifying OIE triples. Then, the semantic-view embedding of NP n^s can be derived as:

$$Emb_s(n^s) = \mathrm{red}(BERT(\texttt{[CLS]}\; Seq(n^s) \;\texttt{[SEP]})) \quad (2)$$

Here, $\mathrm{red}(\cdot)$ denotes a function for extracting the last layer representation of `[CLS]` output by BERT.

Cross-view seed generation & Iterative training. After encoding NPs independently from the relational view and semantic view, we propose to generate cross-view pseudo training seeds and then facilitate mutual supervision of embedding learning between the two view-specific encoders by harnessing the complementary knowledge delivered from the two views. Specifically, we first obtain view-specific similarity matrices by calculating embedding similarities of NP pairs, and apply a top-k constraint for selecting pseudo training seed pairs. Subsequently, these pseudo training seeds generated from one view are leveraged to optimize the other view-specific encoder, enabling them to learn complementary knowledge from each other alternately.

Given two distinct NPs n_i and n_j ($n_i, n_j \in \mathcal{N}$), their relational-view pairwise similarity $s_r(n_i, n_j)$ and semantic-view pairwise similarity $s_s(n_i, n_j)$ are computed via cosine similarity between their corresponding view-specific embeddings. Then, the relational-view similarity matrix $\mathbf{M}^r$ and the semantic-view similarity matrix $\mathbf{M}^s$ are:

$$\mathbf{M}^{r(s)}_{ij} = \begin{cases} s_{r(s)}(n_i, n_j), & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

In order to facilitate cross-training between the two views, for each NP $n_i \in \mathcal{N}$, we select its top-k most similar NPs based on the *relational-view* similarity matrix $\mathbf{M}^r$ to construct the set of *semantic-view* training seeds $\mathcal{S}_s$, which can be formulated as:

$$\mathcal{S}_s = \bigcup_{n_i \in \mathcal{N}} \left\{ (n_i, n_j) \mid n_j \in \mathcal{N} \wedge \mathbf{M}^r_{ij} \in \text{top-}k(\mathbf{M}^r_{i:}) \right\} \quad (4)$$

where top-$k(\cdot)$ is a row-wise function that returns the top-k values for each row, filtering out low-quality matches and preventing noise accumulation. Similarly, based on the *semantic-view* similarity matrix $\mathbf{M}^s$, we could yield the set of *relational-view* training seeds $\mathcal{S}_r$. Then, the generated pseudo training seeds $\mathcal{S}_r$ and $\mathcal{S}_s$ are used to optimize their corresponding view-specific encoders respectively.

For the *relational-view encoder*, we follow (Shen, Yang, and Liu 2022) to augment the training data by swapping the seed NP with its paired counterpart in their involved OIE triples, thereby generating additional positive relational triples for training according to Eq. 1. Specifically, given a relational-view training seed $(n_i, n_j) \in \mathcal{S}_r$, new positive relational triples $\mathcal{T}^+_{(n_i, n_j)}$ can be derived as:

$$\{(n_j, r, n^o) \mid (n_i, r, n^o) \in \mathcal{T}^+\} \cup \{(n^s, r, n_j) \mid (n^s, r, n_i) \in \mathcal{T}^+\} \cup$$
$$\{(n_i, r, n^o) \mid (n_j, r, n^o) \in \mathcal{T}^+\} \cup \{(n^s, r, n_i) \mid (n^s, r, n_j) \in \mathcal{T}^+\}$$

For the *semantic-view encoder*, we optimize it via the cross-entropy loss, maximizing the embedding similarities of semantic-view training seeds $(n_i, n_j) \in \mathcal{S}_s$, with respect to randomly sampled pairs:

$$\mathcal{L}_{sem} = \sum_{(n_i, n_j) \in \mathcal{S}_s} -\log \frac{\exp(s_s(n_i, n_j))}{\exp(s_s(n_i, n_j)) + \sum_{(n_{i'}, n_{j'}) \in \mathcal{S}_s^-} \exp(s_s(n_{i'}, n_{j'}))}$$

Here, $\mathcal{S}_s^-$ denotes the set of negative semantic-view seeds, which are generated by replacing either n_i or n_j in a positive semantic-view seed $(n_i, n_j) \in \mathcal{S}_s$ with a random NP in $\mathcal{N}$.

This generate-and-train process is carried out iteratively to optimize both view-specific encoders until reaching the predefined number T. The well-trained view-specific encoders are then employed in the retrieval process, improving the retrieval precision without the need of external resources.

Dual-view Integrated Retrieval After the cross-training process introduced above, we leverage the well-trained view-specific encoders to obtain view-specific similarity matrices $\mathbf{M}^r$ and $\mathbf{M}^s$ according to Eq. 3, based on which we can retrieve the top-k candidates for each NP from the relational view and semantic view respectively as:

$$\mathbf{R}^r = g(\mathbf{M^r})\,;\, \mathbf{R}^s = g(\mathbf{M^s}) \tag{5}$$

$$g(\mathbf{M}) = [\mathbf{R}_{ij}] = \begin{cases} 1, & \text{if } \mathbf{M}_{ij} \in \text{top-}k(\mathbf{M}_{i:}) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Here, $\mathbf{R}^r$ and $\mathbf{R}^s$ denote retrieval results of the relational view and semantic view respectively, with each element $\mathbf{R}_{ij} \in \{0, 1\}$ indicating whether n_j is retrieved as a top-k candidate for n_i. Then, to facilitate more comprehensive retrieval by integrating the complementary knowledge from both views, we propose to combine $\mathbf{R}^r$ and $\mathbf{R}^s$ to derive the final candidate NP pairs $\mathcal{P}$ as follows:

$$\mathcal{P} = \bigcup_{n_i, n_j \in \mathcal{N}} \{(n_i, n_j) \mid \mathbf{R}^r_{ij} = 1 \vee \mathbf{R}^s_{ij} = 1\} \tag{7}$$

Reader

Dual-view Consensus Based Training As introduced in Section 1, the high-capacity reader is an integral component that severely affects the overall canonicalization performance. In order to capture the direct interaction between NPs, we instantiate the reader as a BERT-based cross-encoder (Devlin et al. 2019), which performs deep cross-interaction between pairwise inputs and has been widely employed for fine-grained similarity calculations between sentences (Wu et al. 2020, 2023). However, despite the effectiveness of cross-encoders in assessing sentence similarities, directly applying them to the OKB canonicalization task may lead to suboptimal results due to the divergence between unstructured sentences and structured OIE triples. To bridge this gap, we propose to tailor the cross-encoder for OKB canonicalization, capitalizing on the dual-view knowledge preserved in the retriever. Concretely, the cross-encoder is expected to mimic pairwise NP similarities embedded in the view-specific similarity matrices $\mathbf{M}^r$ and $\mathbf{M}^s$ yielded by the retriever.

Consensus pair selection. Since both views inevitably contain noise that might misguide the subsequent training process of the cross-encoder, it is crucial to mitigate such noise and prevent error propagation. Following the insight that predictions agreed on by different views are more likely to be correct and less affected by view-specific noise (Christoudias, Urtasun, and Darrell 2008), we employ a strategy to opt for *reliable* training data based on the ***consensus*** of both views. To be specific, we construct the set of consensus pairs $\mathcal{C}$ by simultaneously considering the predictions of the relational view (i.e., the matrix $\mathbf{R}^r$) and the semantic view (i.e., the matrix $\mathbf{R}^s$):

$$\mathcal{C} = \bigcup_{n_i, n_j \in \mathcal{N}} \{(n_i, n_j) \mid \mathbf{R}^{con}_{ij} = 1\}\,;\, \mathbf{R}^{con} = \mathbf{R}^r \odot \mathbf{R}^s$$

where operator $\odot$ denotes the element-wise multiplication, ensuring that an NP pair (n_i, n_j) is included in $\mathcal{C}$ only if both the relational and semantic views identify it as a positive prediction, thus effectively alleviating potential noise.

Training with dual-view aggregated similarity. With the dual-view aggregated similarity matrix $\overline{\mathbf{M}} = \frac{1}{2}(\mathbf{M}^r + \mathbf{M}^s)$, and the set of consensus pairs $\mathcal{C} = \{(n_i, n_j)\}$, we optimize the cross-encoder by minimizing the discrepancy between its output similarities and the corresponding dual-view aggregated similarities using the binary cross-entropy loss function as follows:

$$\mathcal{L}_{cross} = -\sum_{(n_i, n_j) \in \mathcal{C}} \big(\overline{\mathbf{M}}_{ij} \cdot \log(s_{read}(n_i, n_j)) + (1 - \overline{\mathbf{M}}_{ij}) \cdot \log(1 - s_{read}(n_i, n_j))\big)$$

Here, $s_{read}(n_i, n_j)$ represents the fine-grained similarity output by the cross-encoder for the NP pair (n_i, n_j) via high-capacity pairwise reading, with the calculation process elucidated in Section 4. After learning from dual-view aggregated similarities of consensus pairs, the cross-encoder becomes more adept at modeling NP pairs and is anticipated to deliver more accurate similarities for candidate NP pairs owing to its higher model capacity.

High-capacity Pairwise Reading For a retrieved candidate NP pair $(n_i, n_j) \in \mathcal{P}$, we first construct their respective sequences $Seq(n_i)$ and $Seq(n_j)$ as per Eq. 1, and then concatenate them with an additional special token `[SEP]` in between to form the input: $d_{n_i,n_j} =$ `[CLS]` $Seq(n_i)$ `[SEP]` $Seq(n_j)$ `[SEP]`. Subsequently, the input d_{n_i,n_j} is fed into the high-capacity cross-encoder to compute the fine-grained pairwise similarity as:

$$s_{read}(n_i, n_j) = \sigma(\mathbf{CE}(d_{n_i,n_j})) \tag{8}$$

Here, $\mathbf{CE}(\cdot)$ denotes the cross-encoder, which consists of a BERT model followed by a linear layer, and $\sigma(\cdot)$ is the sigmoid activation function used to scale the similarity score within the range of $[0, 1]$.

Grouper

Since the actual number of clusters is unknown, the grouper module requires a clustering algorithm that does not necessitate a predetermined cluster number. What's more, traditional clustering methods (e.g., hierarchical agglomerative clustering (HAC), k-means, etc.) relying on *complete* pairwise similarities between instances (often calculated over instance embeddings) may not be well-suited in our scenario, considering that only *partial* pairwise NP similarities from the reader are available as input. In light of these constraints, the ***graph-based*** clustering algorithm (Ma, Strube, and Zhao 2024; Lin, Li, and Jia 2023) stands out as a fitting choice for the grouper, as it can effectively operate solely based on partial pairwise similarities, aligning seamlessly with the specific characteristics of our framework.

Similarity Graph Construction To enable the operation of graph-based clustering algorithms, we start with the construction of an undirected, weighted similarity graph $G = (V, E, W)$, where V represents the set of nodes and E signifies the set of undirected edges weighted by W. In this graph, each node $v_i \in V$ corresponds to the NP $n_i \in \mathcal{N}$, and an edge $e_{ij} = (v_i, v_j) \in E$ is created if the corresponding NP pair (n_i, n_j) is identified as a candidate NP pair by the retrieval process. The weight $w_{ij} \in W$ assigned to the edge e_{ij} is initialized with the fine-grained pairwise similarity for the NP pair (n_i, n_j) obtained from the reading process (i.e., $w_{ij} = s_{read}(n_i, n_j)$).

Graph-based Grouping As a classical graph-based clustering technique, markov clustering (MCL) (Van Dongen 2008) is employed over the constructed similarity graph G to group NPs into canonicalization clusters, each representing a distinct real-world entity. Built upon the intuition that random walks on the graph are more likely to get stuck within the same cluster rather than traversing across different clusters, MCL computes the probability of random walks between nodes and prunes paths with low probabilities to discover target clusters in the graph. Initially, the transition matrix $\mathbf{P}^0 \in \mathbb{R}^{|V|\times|V|}$ is computed by normalizing the weights of edges around each node in a *column-wise* manner as: $\mathbf{P}^0_{ij} = \frac{w_{ij}}{\sum_{l=1}^{|V|} w_{lj}}$, with $\mathbf{P}^0_{ij} \in [0, 1]$ denoting the transition probability from node v_j to node v_i. Afterwards, MCL simulates random walks by alternating two key operations: *expansion* and *inflation*. Specifically, during iteration t, the *expansion* operation corresponds with squaring the transition matrix $\mathbf{P}^{t-1}$ from the previous iteration to produce the expansion matrix $\mathbf{E}^t$, which facilitates potential connections between distant nodes:

$$\mathbf{E}^t = expansion(\mathbf{P}^{t-1}) = (\mathbf{P}^{t-1})^2 = \mathbf{P}^{t-1}\,\mathbf{P}^{t-1} \quad (9)$$

Subsequently, the *inflation* operation deliberately increases the probabilities of intra-cluster walks and demotes those of inter-cluster walks by raising each element in $\mathbf{E}^t$ to the power of r and then re-normalizing each column, which can be formulated as:

$$\mathbf{P}^t_{ij} = [inflation(\mathbf{E}^t)]_{ij} = \frac{(\mathbf{E}^t_{ij})^r}{\sum_{l=1}^{|V|}(\mathbf{E}^t_{lj})^r} \quad (10)$$

Here, $r \geq 1$ is the inflation parameter that influences the granularity of clusters. The *expansion* and *inflation* operations are repeated iteratively until convergence, and the resulting connected components in the graph constitute the final canonicalization clusters of synonymous NPs.

5 Experiments

Experimental Settings

Datasets We perform experiments on two commonly utilized benchmark datasets of OKB canonicalization, i.e., ReVerb45K (Vashishth, Jain, and Talukdar 2018) and OPIEC59K (Shen, Yang, and Liu 2022), to evaluate the performance of our instantiated framework DUVK.

Evaluation Metrics Following previous OKB canonicalization studies (Galárraga et al. 2014; Shen, Yang, and Liu 2022; Vashishth, Jain, and Talukdar 2018; Dash et al. 2021; Liu et al. 2021; Shen, Yang, and Liu 2024), we employ **average F1** (i.e., averaging macro F1, micro F1 and pairwise F1) as the standard comprehensive evaluation metric.

Implementation Details The dimension d of NP embeddings is set to 300 for both the relational view and semantic view, and the number of cross-training iterations T is 3. The margin hyperparameter γ and inflation parameter r are set to 12 and 3 respectively. We use Adam as the optimizer for training, with a learning rate of 1e-4. During the retrieving stage, k is set to 1 and 3 over ReVerb45K and OPIEC59K, respectively. All experiments are implemented using the PyTorch Framework with a single NVIDIA 3090 (24G) GPU. The datasets and source code are publicly available at https://github.com/TianMonki/DUVK.

Effectiveness Study

To demonstrate the effectiveness of our proposed framework DUVK, we benchmark it against several state-of-the-art baselines, which can be categorized into two types:
(i) *Resource-dependent methods* exploit various external resources to generate seed pairs of synonymous NPs as prior knowledge, including CESI (Vashishth, Jain, and Talukdar 2018), CUVA (Dash et al. 2021), JOCL$_{cano}$ (Liu et al. 2021), CMVC (Shen, Yang, and Liu 2022), CLUE (Shen, Yang, and Liu 2024).
(ii) *Resource-independent methods* rely exclusively on internal knowledge instead of external resources, offering higher adaptability. These methods include Morph Norm (Fader, Soderland, and Etzioni 2011), Text Similarity, IDF Token Overlap and Attribute Overlap (Galárraga et al. 2014).

The experimental results of these baseline methods are shown in Table 1, sourced directly from the paper of CMVC (Shen, Yang, and Liu 2022). Notably, since CLUE (Shen, Yang, and Liu 2024) additionally leverages manually annotated linking labels that are not available in the task of OKB canonicalization, we re-ran its open-source implementation excluding this annotated data to ensure a fair comparison. Overall, the experimental results in Table 1 show a significant performance gap between resource-dependent and resource-independent baselines, indicating that tackling OKB canonicalization without external resources is indeed

Methods	ReVerb45K				OPIEC59K			
	Macro F1	*Micro F1*	*Pairwise F1*	***Average F1***	*Macro F1*	*Micro F1*	*Pairwise F1*	***Average F1***
Resource-dependent Methods								
CESI	0.640	0.855	0.842	0.779	0.328	0.807	0.667	0.600
CUVA	0.682	0.872	0.878	0.810	0.128	0.789	0.686	0.534
$JOCL_{cano}$	0.537	0.854	0.823	0.738	0.465	0.790	0.776	0.677
CMVC	0.662	0.881	0.893	0.812	0.521	0.909	0.878	0.769
CLUE	0.732	0.876	0.814	0.807	0.548	0.912	0.877	0.779
Resource-independent Methods								
Morph Norm	0.627	0.558	0.334	0.506	0.476	0.222	0.186	0.294
Text Similarity	0.625	0.566	0.394	0.528	0.480	0.228	0.192	0.300
IDF Token Overlap	0.603	0.551	0.338	0.497	0.457	0.225	0.190	0.290
Attribute Overlap	0.621	0.558	0.342	0.507	0.474	0.226	0.187	0.295
DUVK (ours)	0.690	0.879	0.870	**0.813**	0.650	0.904	0.847	**0.800**

Table 1: Performance on OKB canonicalization task.

Variants	*ReVerb45K*	*OPIEC59K*
DUVK (full)	0.813	0.800
- w/o relational-view encoder	$0.791_{(-2.2\%)}$	$0.746_{(-5.4\%)}$
- w/o semantic-view encoder	$0.803_{(-1.0\%)}$	$0.754_{(-4.6\%)}$
- w/o dual-view cross-training	$0.808_{(-0.5\%)}$	$0.773_{(-2.7\%)}$
- w/o reader	$0.769_{(-4.4\%)}$	$0.748_{(-5.2\%)}$
- w/o reader & grouper	$0.762_{(-5.1\%)}$	$0.731_{(-6.9\%)}$

Table 2: Average F1 of different variants for DUVK.

more complex and challenging. Despite this, without relying on any external resources, DUVK still achieves competitive or even superior performance compared to state-of-the-art baselines (most of which are resource-dependent methods), demonstrating its strong efficacy. Specifically, in comparison to the latest resource-dependent baseline CLUE, which leverages the correlation between OKB canonicalization and OKB linking, our resource-independent DUVK achieves 0.6% and 2.1% performance gains in average F1 on ReVerb45K and OPIEC59K, respectively. This improvement can be attributed to two factors. Firstly, CLUE solely learns from relational information, while DUVK utilizes both relational and semantic knowledge. Secondly, CLUE follows a traditional embedding-centered pipeline that separates the processes of NP embedding learning and similarity calculations, failing to leverage the direct interaction between NPs, as discussed in Section 1. In contrast, built upon our newly proposed *retrieve-read-group* paradigm, DUVK is capable of exploiting such direct NP interaction effectively to facilitate more fine-grained pairwise NP similarity calculations.

Ablation Study

To validate the effectiveness of each key component in our proposed DUVK, we conduct an ablation study considering five variants, with the results summarized in Table 2.

Effectiveness of Dual-view Knowledge We remove one of the view-specific encoders, retaining the other to function solely as the retriever. Both variants demonstrate prominently decreased performance compared to the full DUVK framework. To gain deeper insights, we discard the dual-view cross-training strategy, making these two view-specific encoders unable to learn from each other. This variant also exhibits a performance drop, confirming that each view could contribute complementary knowledge and the dual-view cross-training strategy can facilitate mutual reinforcement between the two view-specific encoders. Moreover, the influence of dual-view knowledge is less obvious on ReVerb45K, which may be due to the higher homogeneity between relational and semantic knowledge in this dataset, mitigating the incremental benefit of combining them.

Effectiveness of Our Paradigm In the "w/o reader" variant, the reader component is eliminated, leaving only the "retrieve-group" stages. As shown in Table 2, this results in a notable performance decrease of 4.4% on ReVerb45K and 5.2% on OPIEC59K, which indicates the crucial role of the reader in leveraging direct interactions between NPs to provide fine-grained similarities and improve overall canonicalization performance. For the "w/o reader & grouper" variant, we exclude both the reader and grouper, directly applying HAC over the learned embeddings from the retriever to yield the final result, which actually aligns with the traditional embedding-centered pipeline. This variant results in a more pronounced performance degradation (i.e., 5.1% on ReVerb45K and 6.9% on OPIEC59K), highlighting the efficacy of our three-stage *retrieve-read-group* paradigm.

Analysis of Paradigm Flexibility

To demonstrate the flexibility of our proposed *retrieve-read-group* paradigm, we conduct experiments by instantiating each stage with alternative components, as presented in Figure 2. Specifically, for the *retriever*, the lexical-based BM25 (Robertson and Walker 1994), while efficient, achieves lower performance than more advanced semantic-based GloVe (Pennington, Socher, and Manning 2014). Additionally, when comparing the results in ablation study and Figure 2, it can be observed that both DUVK (BM25) and DUVK (GloVe) perform worse than two single-encoder variants, underscoring the superiority of our relational-view and semantic-view encoders. For the *reader*, replacing the original BERT with either the more lightweight ALBERT

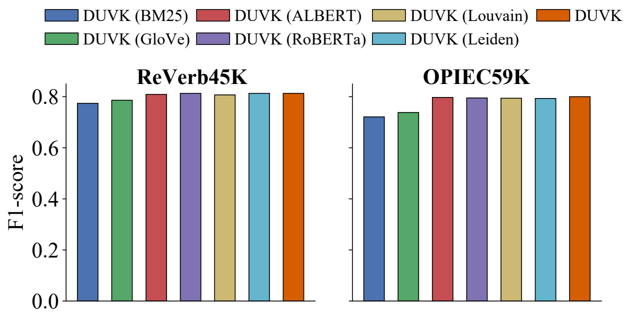

Figure 2: Performance across different instantiations.

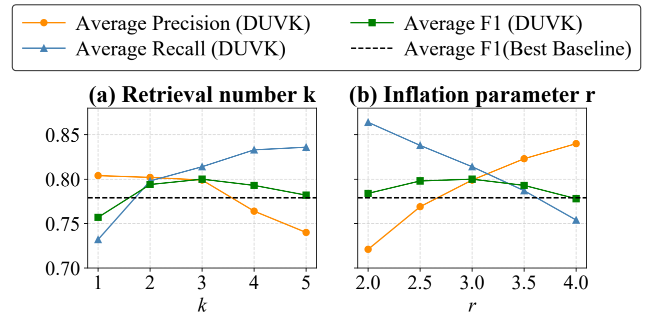

Figure 3: Impact of varying hyperparameters k and r.

(Lan 2019) or the more complex RoBERTa (Liu 2019) yields competitive performance. For the *grouper*, substituting MCL with two common graph-based clustering techniques (i.e., Louvain (De Meo et al. 2011) and Leiden (Traag, Waltman, and Van Eck 2019)) results in minimal performance differences, with Louvain delivering slightly worse outcomes on ReVerb45K. Overall, these results verify the flexibility of our paradigm, as it supports diverse instantiations while maintaining competitive performance.

Sensitivity Study

To examine the sensitivity of DUVK, we investigate its performance with respect to two key hyperparameters: the number of retrieved candidates k and the inflation parameter r.

The Number of Retrieved Candidates k We conduct experiments by varying k from 1 to 5. From Figure 3(a), it is observed that as k initially increases (i.e., when $k < 3$), the overall performance in terms of average F1 gradually improves. This can be explained by the fact that a larger k allows for a more comprehensive retrieval of synonymous NP pairs, resulting in higher recall scores. However, as k continues to increase (i.e., when $k \geq 3$), the overall performance in terms of average F1 peaks and then begins to decline. This could be attributed to the incorporation of more non-synonymous NP pairs as incorrect training seeds, which introduce additional noise to the cross-encoder, thus degrading precision scores. Overall, compared to the best baseline displayed in Figure 3(a), DUVK consistently achieves superior performance over the range of k from 2 to 5, demonstrating its robustness and insensitivity to k.

The Inflation Parameter r The inflation parameter r controls the granularity of the clustering output. To be specific, a lower value of r tends to produce fewer and larger clusters, resulting in higher recall scores, as displayed in Figure 3(b). Conversely, a higher value of r is likely to generate more and smaller clusters, leading to higher precision scores, which is consistent with the performance presented in Figure 3(b). Therefore, the choice of r involves a trade-off between precision and recall. As depicted in Figure 3(b), DUVK maintains competitive performance with the best baseline method across the range of $\{2.0, 2.5, 3.0, 3.5, 4.0\}$, which verifies its robustness and insensitivity to the parameter r.

6 Related Work

For the task of OKB canonicalization, previous methods can be categorized into two types: (i) resource-dependent methods that resort to external resources, and (ii) resource-independent methods that rely solely on internal knowledge within OKBs. Specifically, the first work on OKB canonicalization (Galárraga et al. 2014) belongs to the latter type, utilizing hand-crafted pairwise features like text similarity and IDF token overlap based on internal textual knowledge. CESI (Vashishth, Jain, and Talukdar 2018) inventively introduces the embedding-centered pipeline to tackle OKB canonicalization, which exploits external resources and a KB embedding method (Nickel, Rosasco, and Poggio 2016) to learn the embeddings of NPs, and then calculates pairwise cosine similarities for HAC clustering. Subsequent approaches such as CUVA (Dash et al. 2021), CMVC (Shen, Yang, and Liu 2022) and CLUE (Shen, Yang, and Liu 2024) further extend this pipeline by incorporating various external resources to learn better NP embeddings. However, in these embedding-centered methods, the direct interaction between NPs is not leveraged due to the separation between the processes of individual encoding and similarity calculation, limiting the overall canonicalization performance. In contrast, without the need of any external resources, our proposed paradigm and instantiated framework DUVK achieve leading canonicalization performance by effectively leveraging the direct interaction between NPs for fine-grained pairwise NP similarity calculations.

7 Conclusion

In this paper, we delve into the task of OKB canonicalization. Unlike the traditional embedding-centered pipeline, we introduce an innovative *retrieve-read-group* paradigm that exploits the direct NP interaction to facilitate fine-grained pairwise NP similarity calculations, thereby reducing reliance on external resources. To instantiate this paradigm, we propose DUVK, a novel self-supervised framework that integrates dual-view knowledge from both relational and semantic views. Comprehensive experiments demonstrate that without the use of any external resources, DUVK still outperforms all state-of-the-art baselines.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62272247 and CCF-Huawei Populus Grove Fund.

References

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, 722–735.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*, 2787–2795.

Bovi, C. D.; Telesca, L.; and Navigli, R. 2015. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *TACL*, 3: 529–543.

Christoudias, C.; Urtasun, R.; and Darrell, T. 2008. Multi-view learning in the presence of view disagreement. In *UAI*, 88–96.

Dai, Z.; and Callan, J. 2019. Deeper text understanding for IR with contextual neural language modeling. In *SIGIR*, 985–988.

Dash, S.; Rossiello, G.; Mihindukulasooriya, N.; Bagchi, S.; and Gliozzo, A. 2021. Open Knowledge Graphs Canonicalization using Variational Autoencoders. In *EMNLP*, 10379–10394.

De Meo, P.; Ferrara, E.; Fiumara, G.; and Provetti, A. 2011. Generalized louvain method for community detection in large networks. In *ISDA*, 88–93.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 4171–4186.

Dietz, L.; Kotov, A.; and Meij, E. 2018. Utilizing knowledge graphs for text-centric information retrieval. In *SIGIR*, 1387–1390.

Etzioni, O.; Fader, A.; Christensen, J.; Soderland, S.; et al. 2011. Open information extraction: The second generation. In *IJCAI*, 3–10.

Fader, A.; Soderland, S.; and Etzioni, O. 2011. Identifying relations for open information extraction. In *EMNLP*, 1535–1545.

Galárraga, L.; Heitz, G.; Murphy, K.; and Suchanek, F. M. 2014. Canonicalizing open knowledge bases. In *CIKM*, 1679–1688.

Gashteovski, K.; Wanner, S.; Hertling, S.; Broscheit, S.; and Gemulla, R. 2019. OPIEC: an open information extraction corpus. In *AKBC*.

Huang, X.; Zhang, J.; Li, D.; and Li, P. 2019. Knowledge graph embedding based question answering. In *WSDM*, 105–113.

Jain, A. K.; Murty, M. N.; and Flynn, P. J. 1999. Data clustering: a review. *ACM CSUR*, 31(3): 264–323.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Philip, S. Y. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE TNNLS*, 33(2): 494–514.

Karpukhin, V.; Oğuz, B.; Min, S.; Lewis, P.; Wu, L.; Edunov, S.; Chen, D.; and Yih, W.-t. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*, 6769–6781.

Lan, Z. 2019. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*.

Lin, L.; Li, R.; and Jia, T. 2023. Scalable and effective conductance-based graph clustering. In *AAAI*, 4471–4478.

Liu, Y. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, Y.; Shen, W.; Wang, Y.; Wang, J.; Yang, Z.; and Yuan, X. 2021. Joint open knowledge base canonicalization and linking. In *SIGMOD*, 2253–2261.

Lukovnikov, D.; Fischer, A.; Lehmann, J.; and Auer, S. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *WWW*, 1211–1220.

Ma, X.; Guo, J.; Zhang, R.; Fan, Y.; Ji, X.; and Cheng, X. 2021. Prop: Pre-training with representative words prediction for ad-hoc retrieval. In *WSDM*, 283–291.

Ma, X.; Strube, M.; and Zhao, W. 2024. Graph-based clustering for detecting semantic change across time and languages. In *EACL*, 1542–1561.

Moon, C.; Jones, P.; and Samatova, N. F. 2017. Learning entity type embeddings for knowledge graph completion. In *CIKM*, 2215–2218.

Nickel, M.; Rosasco, L.; and Poggio, T. 2016. Holographic embeddings of knowledge graphs. In *AAAI*, 1955–1961.

Pavlick, E.; Rastogi, P.; Ganitkevitch, J.; Van Durme, B.; and Callison-Burch, C. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *ACL*, 425–430.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.

Robertson, S. E.; and Walker, S. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR*, 232–241.

Shen, W.; Li, Y.; Liu, Y.; Han, J.; Wang, J.; and Yuan, X. 2023. Entity linking meets deep learning: Techniques and solutions. *IEEE TKDE*, 35(3): 2556–2578.

Shen, W.; Yang, B.; and Liu, Y. 2024. Jointly Canonicalizing and Linking Open Knowledge Base via Unified Embedding Learning. In *WWW*, 2304–2314.

Shen, W.; Yang, Y.; and Liu, Y. 2022. Multi-View Clustering for Open Knowledge Base Canonicalization. In *SIGKDD*, 1578–1588.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: a core of semantic knowledge. In *WWW*, 697–706.

Tang, X.; Zhang, J.; Chen, B.; Yang, Y.; Chen, H.; and Li, C. 2020. BERT-INT: a BERT-based interaction model for knowledge graph alignment. In *IJCAI*, 3174–3180.

Traag, V. A.; Waltman, L.; and Van Eck, N. J. 2019. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1): 1–12.

Trisedya, B. D.; Qi, J.; and Zhang, R. 2019. Entity alignment between knowledge graphs using attribute embeddings. In *AAAI*, volume 33, 297–304.

Van Dongen, S. 2008. Graph clustering via a discrete uncoupling process. *SIMAX*, 30(1): 121–141.

Vashishth, S.; Jain, P.; and Talukdar, P. 2018. Cesi: Canonicalizing open knowledge bases using embeddings and side information. In *WWW*, 1317–1327.

Vrandečić, D.; and Krötzsch, M. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10): 78–85.

Wang, Q.; Mao, Z.; Wang, B.; and Guo, L. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE TKDE*, 29(12): 2724–2743.

Wang, X.; Wang, D.; Xu, C.; He, X.; Cao, Y.; and Chua, T.-S. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*, 5329–5336.

Wu, L.; Petroni, F.; Josifoski, M.; Riedel, S.; and Zettlemoyer, L. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *EMNLP*, 6397–6407.

Wu, T.; Bai, X.; Guo, W.; Liu, W.; Li, S.; and Yang, Y. 2023. Modeling fine-grained information via knowledge-aware hierarchical graph for zero-shot entity retrieval. In *WSDM*, 1021–1029.

Xiong, C.; Power, R.; and Callan, J. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *WWW*, 1271–1279.

Yates, A.; Banko, M.; Broadhead, M.; Cafarella, M. J.; Etzioni, O.; and Soderland, S. 2007. Textrunner: open information extraction on the web. In *NAACL*, 25–26.

Zhang, Q.; Sun, Z.; Hu, W.; Chen, M.; Guo, L.; and Qu, Y. 2019. Multi-view knowledge graph embedding for entity alignment. In *IJCAI*, 5429–5435.