Wiki Entity Summarization Benchmark

Saeedeh Javadi[*]
Polytechnic University of Turin
saeedeh.javadi@studenti.polito.it

Atefeh Moradan[*]
Aarhus University
atefeh.moradan@cs.au.dk

Mohammad Sorkhpar[*]
Indiana State University
msorkhpar@sycamores.indstate.edu

Klim Zaporojets
Aarhus University
klim@cs.au.dk

Davide Mottin
Aarhus University
davide@cs.au.dk

Ira Assent
Aarhus University
ira@cs.au.dk

Abstract

Entity summarization aims to compute concise summaries for entities in knowledge graphs. Existing datasets and benchmarks are often limited to a few hundred entities and discard graph structure in source knowledge graphs. This limitation is particularly pronounced when it comes to ground-truth summaries, where there exist only a few labeled summaries for evaluation and training. We propose WIKES (Wiki Entity Summarization Benchmark), a comprehensive *benchmark* comprising of entities, their summaries, and their connections. Additionally, WIKES features a dataset *generator* to test entity summarization algorithms in different areas of the knowledge graph. Importantly, our approach combines graph algorithms and NLP models, as well as different data sources such that WIKES does not require human annotation, rendering the approach cost-effective and generalizable to multiple domains. Finally, WIKES is scalable and capable of capturing the complexities of knowledge graphs in terms of topology and semantics. WIKES features existing *datasets* for comparison. Empirical studies of entity summarization methods confirm the usefulness of our benchmark. Data, code, and models are available at: `https://github.com/msorkhpar/wiki-entity-summarization`.

1 Introduction

Knowledge Graphs (KGs) are a valuable information representation: interconnected networks of entities and their relationships enable machine reasoning to empower question answering Hu et al. [2018], Lan et al. [2019], recommender systems Wang et al. [2018], information retrieval Raviv et al. [2016]. KGs may comprise millions of entities representing real-world objects, concepts, or events.

Yet, the size and complexity of these KGs progressively expand, rendering it increasingly challenging to convey the essential information about an entity in a concise and meaningful way Suchanek et al. [2007], Vrandečić and Krötzsch [2014]. This is where entity summarization becomes relevant. *Entity summarization* (ES) Liu et al. [2021] is the process of generating a concise and informative summary that captures the most salient aspects of the entity description, based on the information available in

[*]Equal contribution

Submitted to the 38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks. Do not distribute.

the KGs. In ES, the entity *description* refers to all the triples involving such an entity. For instance, Figure 1 illustrates a set of relationships surrounding the entity `Ellen Johnson Sirleaf` in a KG, along with a possible summary for this entity. Extensive descriptions can overwhelm users and exceed the capacity of typical user interfaces, making it challenging to identify the most relevant triples. Entity summarization addresses this issue by computing an optimal compact summary for an entity, selecting a size-constrained subset of triples Liu et al. [2021].

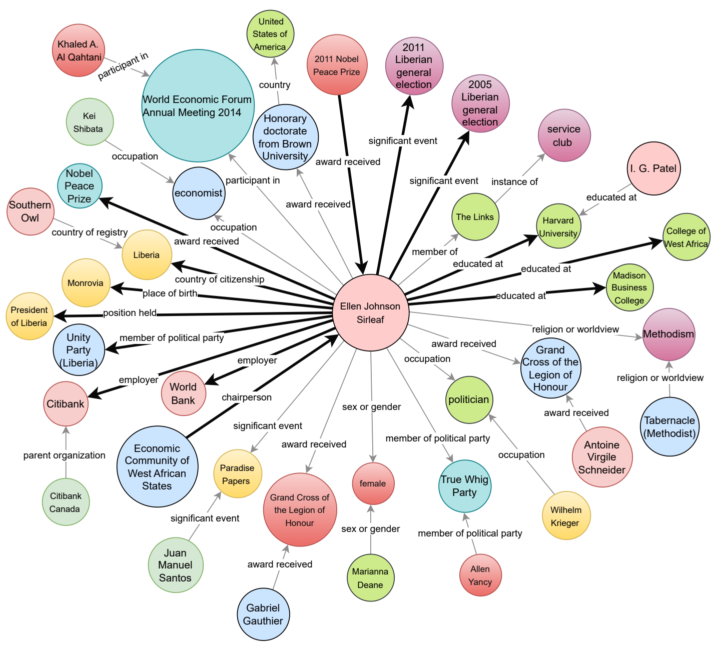

Figure 1: KG subgraph of entity `Ellen Johnson Sirleaf`: arrows depict the subgraph of relationships to other entities, and labels indicate their roles. Selecting the bold edges as entity summaries of the most relevant triples may reduce information overload while concisely describing the entity.

Despite advances in entity summarization techniques Liu et al. [2021], the development and evaluation of these methods are hindered by a number of limitations in the benchmarks and datasets Liu et al. [2020], Cheng et al. [2023]. The first limitation of the current benchmarks is the small dataset size, encompassing only a few hundred entities. Second, the generation of ground-truth summaries for testing mostly relies on expensive and lengthy manual annotation. Moreover, the dependence on a few human annotators often biases the data towards the annotators' preferences and knowledge. Third, existing benchmarks often disregard the wealth of information in the knowledge graph structure.

To address the above limitations, we propose:

- **Novel WIKES benchmark for ES** based on summaries and graphs from Wikidata and Wikipedia.
- **Subgraph extraction method** preserving the complexity of real-world KGs; subsampling using random walks and proportionally preserving node degrees, WIKES captures the structure of the entities up to the second-hop neighborhood, thereby ensuring that the connections in WIKES accurately reflect those in the source KG.
- **Comprehensive summaries for *any* entity in the KG**, ensuring that summaries are both relevant and contextually rich by deriving them directly from corresponding Wikipedia abstracts, minimizing human bias, as these abstracts are created and reviewed by several experts. In this manner, WIKES is scalable, enabling it to generate large benchmark resources efficiently with high-quality annotation.
- **Automatic entity summarization dataset generator** allows for the creation of arbitrarily large datasets, encompassing various domains of knowledge.

2 Existing Datasets

Here, we review the existing datasets for entity summarization. Table 1 provides an overview and statistics of the current datasets in this field. FACES and INFO datasets have a higher density than the entities in the Entity Summarization Benchmark (ESBM). It is also clear that LMDB and FACES are not connected graphs, that challenge graph-based learning methods where the information cannot easily propagate in disconnected networks. Specifically, FACES consists of 12 connected components, which complicates the learning process for graph embedding methods by limiting the richness of information that can be leveraged from the graph.

Table 1: Entity summarization datasets in terms of number of entities $|\mathcal{V}|$, triples $|\mathcal{E}|$, number of ground-truth summaries (target entities), density as $|\mathcal{E}|/\binom{|\mathcal{V}|}{2}$, graph connectivity, number of components, sampling method to select entities and subgraph, and minimum / maximum node degree.

Metric	DBpedia (ESBM)	LMDB (ESBM)	FACES	INFO
Entities ($\lvert\mathcal{V}\rvert$)	2 721	1 853	1 379	1 410
Relations ($\lvert\mathcal{E}\rvert$)	4 436	2 148	2 152	2 019
Target Entities	125	50	50	100
Density	0.0005	0.0006	0.0011	0.0010
Sampling method	Not specified	Not specified	Not specified	Not specified
Connected-graph	Yes	No	No	Yes
Num-comp	1	2	12	1
Min Degree	1	1	1	1
Max Degree	125	208	88	100

We provide here a comprehensive description of each dataset or benchmark:

- **ESBM** Liu et al. [2020]: The Entity Summarization Benchmark (ESBM) is the first benchmark to evaluate the performance of entity summarization methods. ESBM has three versions; v1.2 is the latest and most extensive version. This version comprises 175 entities, with 150 from DBpedia Lehmann et al. [2015] and 25 from LinkedMDB Hassanzadeh and Consens [2009]. The summaries comprise triples selected by 30 "researchers and students" annotators. Each entity has exactly 6 summaries. Despite encompassing two datasets, ESBM has several limitations. First, the entity sampling method is not explained. In particular, some triples in the neighborhood of the entity are missing in the datasets. Second, there are no connections among the entities in the neighborhood, nor any two-hop neighborhood. Third, the expertise and background of the annotators are not assessed nor disclosed. Due to the expensive annotation process, the dataset size is small.
- **FACES** Gunaratna et al. [2015] is a dataset from DBpedia (version 3.9) Auer et al. [2007] and includes 50 randomly selected entities, each with at least 17 different types of relations. Similar to ESBM, the FACES ground-truth is also generated manually.
- **INFO** Cheng et al. [2023] contains 100 randomly selected entities from 10 classes in DBpedia. It comprises two sets of ground-truth summaries, REF-E and REF-W. REF-E summaries comprise a selection of triples from five experts adhering to a 140-character limit, similar to typical Google search result snippets. REF-W summaries are obtained by one expert who reads the abstract sections of the respective entities on Wikipedia and selects neighboring entities that closely match the Wikipedia abstracts. The number of ground-truth summaries per entity varies, as some experts evaluate multiple entities. This inconsistency complicates the evaluation process. The expertise of the annotators remains unspecified.

In contrast, our benchmark uses Wikidata to automatically map entities from Wikipedia to Wikidata. This automation allows us to efficiently generate summaries for any number of entities. Unlike previous work, we use the Wikipedia abstract as a summary instead of manual annotators. Each abstract is a collaboration of many users; as such, it should not introduce obvious biases. Additionally, with this process, we ensure high-quality and cost-effective summaries. Furthermore, we present the characteristics of our dataset in Table 3. The WikES benchmark includes a larger number of entities and relations than existing datasets. It is a connected graph containing approximately 500 seed nodes. Further details regarding the specific characteristics of our dataset are provided in Section 3.4.

3 The WIKES Benchmark

A *Knowledge Graph* $\mathcal{KG} = (\mathcal{V}, \mathcal{R}, \mathcal{T})$ is a directed multigraph consisting of entities $\mathcal{V} = \{v_1, \ldots, v_n\}$, relationships $\mathcal{R}$, and triples $\mathcal{T} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$. The set of edges $\mathcal{E} = \{(i, j) \mid v_i, v_j \in \mathcal{V} \wedge \exists r \in \mathcal{R} \text{ s.t. } (v_i, r, v_j) \in \mathcal{T}\}$ contains pairs of nodes connected by a relationship.

The *t-hop neighborhood* $\mathcal{N}_t(v_i)$ of node v_i is the set of nodes reachable from v_i within t edges when ignoring edge directions.

A *summary* for an entity v_i is a subset $\mathcal{S}(v_i) \subseteq \Delta_t(v_i)$ of triples from the t-description of v_i, where the *t-description* of an entity $v_i \in \mathcal{V}$ in a knowledge graph $\mathcal{KG}$ is the set $\Delta_t(v_i) = \{(s, p, o) \in \mathcal{T} \mid s \in \mathcal{N}_t(v_i) \vee o \in \mathcal{N}_t(v_i)\}$ of triples in which one of the entities is in the t-hop neighborhood of v_i.

Entity summarization for an entity $v_i \in \mathcal{V}$ in a knowledge graph $\mathcal{KG}$ aims to find a summary $\mathcal{S}(v_i)$ that maximizes some score among all possible summaries for v_i, i.e.,

$$\underset{\substack{\mathcal{S}(v_i) \subseteq \Delta_t(v_i) \\ |\mathcal{S}(v_i)| = k}}{\arg\max} \ \text{score}(\mathcal{S}(v_i)), \tag{1}$$

3.1 Extracting Summaries from Wikidata using Wikipedia Abstracts

We extract summaries for each Wikidata item using Wikipedia abstracts and infoboxes. Each abstract is a joint effort of many users and experts, which ensures quality and accuracy. Leveraging Wikipedia, we avoid time-consuming manual annotation and enable the automatic generation of large-scale datasets.

Wikidata is a free and collaborative knowledge base that collects structured data to support Wikipedia and other Wikimedia projects. It includes descriptions and labels for entities. The descriptions offer in-depth details, while the labels serve as concise identifiers, facilitating efficient data retrieval and integration in subsequent steps. We load all Wikidata items XML dump files published on 2023/05/01[2] as entities $\mathcal{V}$ alongside their properties as relationships $\mathcal{R}$ into a graph database[3]. The result is a graph that connects all Wikidata items and statements. We include items if they (1) are not marked as redirects, (2) belong to the main Wikidata namespace, and (3) have an English label or description. Additionally, we load metadata for each Wikidata item and property, including labels and descriptions, into a relational database[4]. **Wikipedia** pages contain infoboxes, abstracts, page content, categories, references, and more. Links to other Wikipedia pages are referred to as mentions. We detect these mentions in the abstracts and infoboxes of Wikipedia pages to use them later for labeling the summaries in Wikidata. We extract and load all the content from the XML dump files of Wikipedia pages, published on 2023/05/01[5], into a relational database under the same conditions as Wikidata: the pages must be in English and not redirected.

Summary annotation. We annotate the summaries in Wikidata using the corresponding Wikipedia pages. For each Wikipedia page corresponding to a Wikidata entity, we iterate through all connected Wikidata items using Wikidata properties. If a connected Wikidata item is mentioned in the Wikipedia abstract and infobox, we annotate the Wikidata item with the corresponding Wikidata property as part of the summary.

Wikidata is a directed multigraph, which means that each entity (Wikidata item) can be connected to another entity via multiple relations (Wikidata properties). Yet, links in Wikipedia are not labeled; as such, we need to select one of the relations for the summary. To annotate the correct Wikidata

[2] `https://dumps.wikimedia.org/wikidatawiki/`
[3] `https://neo4j.com`
[4] `https://www.postgresql.org/`
[5] `https://dumps.wikimedia.org/enwiki/`

property as part of the summary, we employ the DistilBERT model Sanh et al. [2019]. DistilBERT is a fast and lightweight model with a reduced number of parameters compared to the original BERT model. This way, we can efficiently process large amounts of data while maintaining high-quality embeddings for accurate relation selection.

Concretely, we first embed the abstract of the Wikidata item for which we are generating summaries using DistilBERT. We then calculate the cosine similarity between the embedding of the abstract and the embeddings of each candidate relation. Finally, we add the relation with the highest cosine similarity to the abstract embedding to the summary. This approach ensures that the most relevant Wikidata property is selected for the summary based on its semantic similarity to the Wikipedia abstract.

3.2 Capturing the Graph Structure

Here we introduce the WIKES generator algorithm. The main idea is to sample a connected graph that preserves the original graph structure. To this end, we employ random walks Pearson [1905].

A random walk is a stochastic process defined as a sequence of steps, where the direction and magnitude of each step are determined by the random variable $\mathrm{X}_{t+1} = \mathrm{X}_t + \mathrm{S}_t$ where X_t represents the position at time t, and S_t is the step taken from position X_t.

The process is a Markov process, characterized by its memoryless property:

$$P(X_{t+1} = x | X_t = x_t, X_{t-1} = x_{t-1}, \ldots, X_0 = x_0) = P(X_{t+1} = x | X_t = x_t) \tag{2}$$

In adapting this concept to our work, we redefine the number of random walks assigned to nodes based on their degrees, ensuring the distribution remains proportional to real data. This is achieved through logarithmic transformation and normalization. The logarithmic transformation is applied to reduce the impact of high-degree nodes and also low-degree nodes, making it more manageable for the random walk. Given a graph with node degrees $\{d_1, d_2, \ldots, d_i\}$, the log-transformed degree for node i is $L_i = \log(d_i)$. These values are then normalized:

$$N_i = \frac{L_i - \min(\{L\})}{\max(\{L\}) - \min(\{L\})} \tag{3}$$

where N_i is the normalized logarithmic degree of node i. Finally, the number of random walks R_i assigned to each node is:

$$R_i = \text{round}\left(\text{minRW} + N_i \times (\text{maxRW} - \text{minRW})\right) \tag{4}$$

Here, minRW and maxRW are the user-defined minimum and maximum limits for random walks. This adaptation ensures that the random walks are proportional to the normalized logarithmic degree of each node, reflecting the true structure of the network. For a small dataset we set minRW $= 100$ and maxRW $= 300$; for a medium dataset minRW $= 150$ and maxRW $= 600$; for a large dataset, minRW $= 300$ and maxRW $= 1800$. This ensures that the random walks are tailored to both the scale and the complexity of the dataset. Importantly, our approach can be used to extract further subgraphs at the scale needed for benchmarking in a given scenario.

Moreover, the random walk sampling process requires a set of seed nodes as a starting point. In our case, the seed nodes represent the target entities we are interested in. The seed nodes can be any Wikidata Item Identifier, Wikipedia title, or Wikipedia ID of the Wikipedia pages. We collect the seed nodes on the condition that they have at least k (default $k = 5$) common entities with the abstract section and the infobox in the Wikipedia pages. Therefore, this model is flexible, allowing you to choose any seed nodes from any domain as an input. In the datasets that we generated, we collect seed nodes from Laouenan et al. [2022]. This paper has published information about individuals from various domains. The authors collected data from multiple Wikipedia editions and Wikidata, using deduplication and cross-verification techniques to compile a database of 1.6 million individuals with English Wikipedia pages. The seed nodes that we use include actor, athletic, football, journalist, painter, player, politician, singer, sport, writer, lawyer, film, composer, novelist, poet, and screenwriter. Using combinations of these seed nodes, we generate four sets of datasets, with each set having small, medium, and large versions. In Table 4 in Section 6 in the supplementary material, we present the seed nodes and their proportions for each dataset and their corresponding train-test-val splits.

3.3 WIKES Generator

We discuss how WIKES is created, and how further benchmarks can be generated without the need for manual annotators. Algorithm 1 details the generator, which consists of the following steps.

Step1: Retrieve summaries of each seed node (explained in Section 3.1)

Step2: Expand the graph using the random walk method in Section 3.2. Set the random walk's length n (default $n = 2$), which means it explores up to the n-hop neighborhood of each seed node.

Step3: Check if the graph is connected. If it is, done. If not, identify all disconnected components and sort them by size, from largest to smallest. In each iteration, connect smaller components to the largest component using h connections. Utilize the shortest path method, selecting paths that are equal to or less than a minimum path length l. Continue connecting nodes from the smaller component to the larger one until h nodes are connected. After each iteration, check graph connectivity again. If all components are connected to the largest component, the algorithm ends. Otherwise, re-sort components and increase l by 1. Repeat until the graph is a single connected component.

Algorithm 1 WIKES Generator

```
Input: Graph G, seed nodes S, random walk length n, minimum path length l
Output: A connected graph
procedure GenerateGraph(G, S, n, l)
    summaries ← RetrieveSummaries(S)
    G ← RandomWalkExpansion(G, S, n) mentioned in section 3.2
    is_connected ← CheckConnectivity(G)
    while not is_connected do
        components ← FindComponents(G)
        Sort components by size in descending order
        largest ← components[0]
        for comp in components[1 :] do
            Connect comp to largest using h connections via shortest paths ≤ l
            G ← UpdateGraph(G, comp, largest)
            is_connected ← CheckConnectivity(G)
            if is_connected then
                break
            end if
        end for
        l ← l + 1
    end while
    return G
end procedure
```

3.4 WIKES Datasets

We generate three sizes for each of the four datasets, obtaining 12 datasets. We present their characteristics in Table 3 in section 6. The number of entities in the small datasets ranges from approximately $70k$ to $85k$, and the number of relations ranges from around $120k$ to $135k$. In the medium datasets, the number of entities ranges from $100k$ to $130k$, and the number of relations ranges from $195k$ to $220k$. The number of entities in the large datasets ranges from approximately $185k$ to $250k$, and the number of relations ranges from around $397k$ to $470k$. The average runtime for generating small graphs is approximately 128 seconds; for medium-sized graphs, it is approximately 216 seconds; and for large graphs, it is approximately 512 seconds. We construct the train-test-validation split for each dataset with 70% for training, 15% for testing, and 15% for validation. Detailed information about the run time, as well as the number of nodes and relations for these splits, is available on our GitHub repository. All graphs in each train-test-validation splits are connected.

4 Empirical Evaluation

We study the quality of WIKES using the following metrics:

F-Score. Let $\mathcal{S}_m$ the summary obtained by a summarization method and $\mathcal{S}_h$ the ground-truth summary. We compare $\mathcal{S}_m$ with $\mathcal{S}_h$ using the F1-score based on precision P and recall R:

$$\mathrm{F1} = \frac{2 \cdot P \cdot R}{P + R}, \text{ where P} = \frac{|\mathcal{S}_m \cap \mathcal{S}_h|}{|\mathcal{S}_m|} \text{ and R} = \frac{|\mathcal{S}_m \cap \mathcal{S}_h|}{|\mathcal{S}_h|} \tag{5}$$

The F1 score lies within [0,1]. High F1 indicates that $\mathcal{S}_m$ is closer to the ground-truth $\mathcal{S}_h$.

Mean Average Precision (MAP). This metric is particularly suitable for evaluating ranking tasks because it takes into account the order of the predicted triples. MAP calculates precision at each position i in the predicted summary and averages these values over all relevant summary triples. It reflects both the relevance and the ranking quality of the predicted summaries. MAP, unlike F1-score, does not depend on a specific value of k. This makes it a robust metric for assessing how well a summarization method ranks the relevant triples.

$$\mathrm{MAP} = \frac{1}{N} \sum_{n=1}^{N} \frac{\sum_{i=1}^{|\mathcal{S}_m^{(n)}|} \begin{cases} \mathrm{Precision@}i(\mathcal{S}_h^{(n)}) & \text{if } \mathrm{Rel}(n,i) \\ 0 & \text{otherwise} \end{cases}}{|\mathcal{S}_h^{(n)}|} \tag{6}$$

where N is the total number of entities, $\mathcal{S}_h^{(n)}$ is the set of ground-truth summary triples for a particular entity v_n, $\mathcal{S}_m^{(n)}$ is the set of predicted summary triples for the entity v_n, $\mathrm{Precision@}i$ is the precision at the i-th position in the predicted summary, and $\mathrm{Rel}(n,i)$ indicates whether the i-th predicted triple for entity v_n is relevant (i.e., it belongs to $\mathcal{S}_h^{(n)}$). MAP scores are in the range [0,1], where a higher MAP indicates better performance in terms of correctly predicting relevant summary triples. To account for the varying lengths of the ground-truth summaries in real-world data, we also calculate MAP and F-score (which we refer to as dynamic MAP and dynamic F-score) by setting the length of the generated summary ($|\mathcal{S}_m|$) equal to the length of the corresponding ground-truth summary ($|\mathcal{S}_h|$).

We analyze our dataset and compare it with the ESBM benchmark using statistical measures such as frequency and inverse frequency of entities and relations. We calculate the F-score and MAP score for the top-5 and top-10 of both the ESBM dataset and our WikiProFem. We choose top-5 and top-10 because we only have ground-truth summaries for top-5 and top-10 in the ESBM dataset. The F-score and MAP results for ESBM are presented in Figure 2. The statistics show that for DBpedia, the F-score using inverse relation frequency outperforms the random baseline by 0.15 for top-5 and by 0.34 for top-10. Furthermore, when using inverse entity frequency, DBpedia achieves an even higher F-score, surpassing the random baseline by 0.07 for top-5 and by 0.15 for top-10. For LMDB, we observe a similar trend when using inverse frequency. The F-score surpasses the random baseline by 0.10 for top-5 and by approximately 0.15 for top-10. Additionally, when employing entity frequency, LMDB achieves an F-score that is around 0.17 higher than the baseline for top-5 and 0.07 higher for top-10. The results demonstrate that ESBM exhibits a strong bias towards entity, reverse entity, and relation frequency. For Map score, we are exactly observing the same behavior for ESBM. We believe that the bias comes from the fact that the datasets are small, their second-hop neighborhood is not considered, and the relations between their first-hop neighbors are not considered. On the other hand, Figure 3 shows the F-score for top-5, top-10 and dynamic F-score on WIKES. Since the length of summaries varies with the abstract, we calculate the F-score for each seed node based on its summary length. Results show that WIKES F-score is close to random for different statistics, thus rejecting the hypothesis of obvious biases. We observe a minor bias towards node frequency in small datasets. Yet, as we increase the size of the dataset, this bias disappears. We observe a similar behavior with MAP in Figure 4 Furthermore, we use *the entire* Wikidata to measure the F-score for our seed nodes. Thus, importantly, we observe that our dataset's F-score trend is comparable to that of the entire data, especially our large dataset. We also extracted the first-hop neighborhood of all our seed nodes and observed a small bias in the F-score top-5 and dynamic F-score. We conclude that

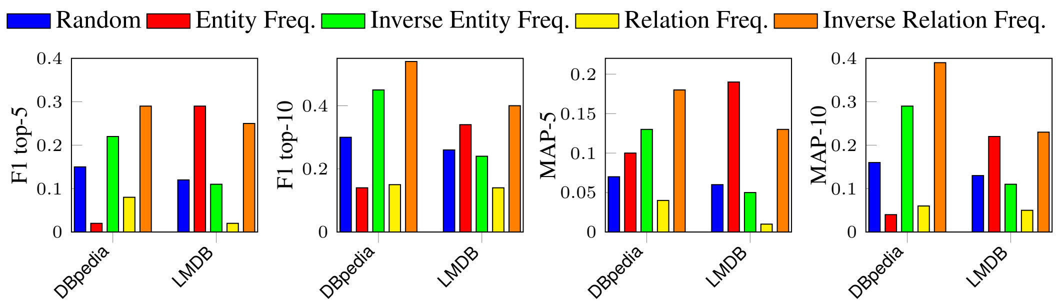

Figure 2: F1 score and MAP for frequency statistics on ESBM datasets.

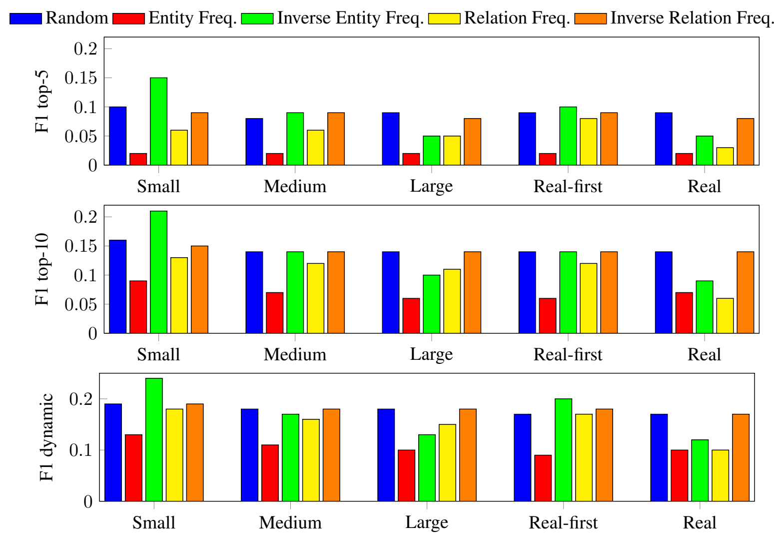

Figure 3: F1 for frequency statistics on WikiProFem.

adding the two-hop neighborhood makes the sample follow the graph distribution. Thus, WIKES is an unbiased benchmark that retains the source KG distribution.

We evaluate the performance of different entity summarization methods on our benchmark, and provide all implementations in the WIKES GitHub repository.

- **PageRank** Ma et al. [2008] ranks nodes in a graph based on the structure of incoming links, with the idea that more important nodes are likely to receive more links from other nodes.
- **RELIN** Cheng et al. [2011] is a weighted PageRank algorithm that evaluates the relevance of triples within a graph structure. We have re-implemented this model according to the specifications in the referenced paper. On our smaller dataset version, RELIN takes approximately 6 hours to compute all summaries.
- **LinkSum** Thalhammer et al. [2016] is a two-step, relevance-centric method that combines PageRank with an adaptation of the Backlink algorithm to identify relevant connected entities. We have re-implemented it according to the paper. The LinkSum method initially takes 10 hours to compute the backlinks for each node in the small version of our dataset. By parallelizing the implementation, we reduced this to one hour. Additionally, the Backlink algorithm itself initially takes 100 minutes, but with parallelization, this was reduced to 10 minutes for the small version of our dataset.

Due to the inefficiency of the methods, we use a smaller version of WIKES for evaluation. The results in Table 2 show that LinkSum outperforms both RELIN and PageRank. These findings suggest that

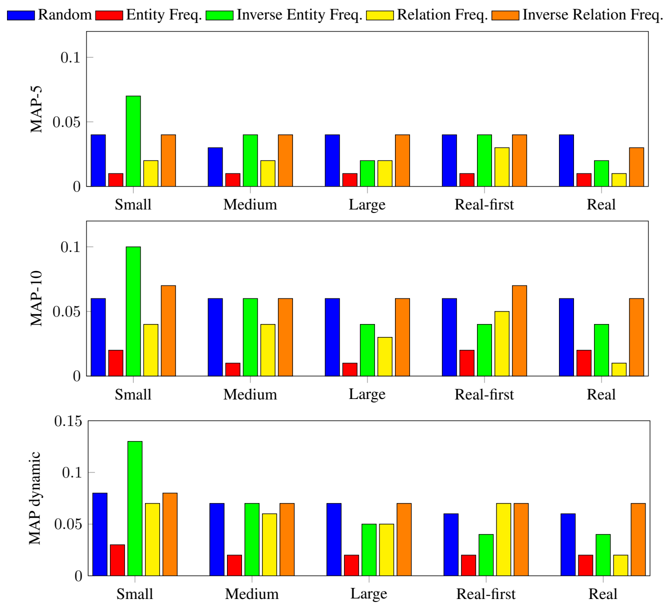

Figure 4: MAP for frequency statistics on WikiProFem.

models capable of exploiting the graph structure while handling large-scale datasets and maintaining high accuracy in entity summarization are valuable for such real-world KGs, such as WIKES.

		topK = 5		topK = 10		Dynamic	
Model	Dataset	F-Score	MAP	F-Score	MAP	F-Score	MAP
PageRank	WikiLitArt	0.024	0.01	0.081	0.02	0.175	0.046
	WikiCinema	0.003	0.001	0.041	0.005	0.146	0.028
	WikiPro	0.060	0.02	0.169	0.049	0.288	0.109
	WikiProFem	0.032	0.01	0.093	0.024	0.145	0.036
RELIN	WikiLitArt	0.093	0.035	0.148	0.054	0.208	0.080
	WikiCinema	0.071	0.023	0.127	0.038	0.209	0.068
	WikiPro	0.125	0.053	0.200	0.086	0.273	0.127
	WikiProFem	0.111	0.050	0.179	0.081	0.219	0.095
LinkSum	WikiLitArt	0.184	0.080	0.239	0.109	0.225	0.127
	WikiCinema	0.119	0.048	0.152	0.060	0.135	0.068
	WikiPro	0.249	0.127	0.347	0.190	0.350	0.242
	WikiProFem	0.195	0.097	0.236	0.127	0.213	0.136

Table 2: Performance comparison of entity summarization models on the small version of WIKES. The models are evaluated with different topK values (5 and 10) and a dynamic setting.

5 Conclusion

We introduce WIKES (Wiki Entity Summarization Benchmark), a benchmark for KG entity summarization which provides a scalable dataset generator that eschews the need for costly human annotation. WIKES uses Wikipedia abstracts for automatic summary generation, ensuring contextually rich and unbiased summaries. It preserves the complexity and integrity of real-world KGs through a random walk sampling method that captures the structure of entities down to their second-hop neighborhoods. Empirical evaluations demonstrate that WIKES provides high-quality large-scale datasets for entity summarization tasks, and that it captures the complexities of knowledge graphs in terms of topology, making it a valuable resource for evaluating and improving entity summarization algorithms.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

Gong Cheng, Thanh Tran, and Yuzhong Qu. Relin: Relatedness and informativeness-based centrality for entity summarization. In *The Semantic Web – ISWC 2011*, pages 114–129, 2011.

Gong Cheng, Qingxia Liu, and Yuzhong Qu. Generating characteristic summaries for entity descriptions. *TKDE*, 35(5):4825–4835, 2023.

Kalpa Gunaratna, Krishnaprasad Thirunarayan, and Amit P Sheth. Faces: diversity-aware entity summarization using incremental hierarchical conceptual clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 116–122, 2015.

Oktie Hassanzadeh and Mariano P Consens. Linked movie data base. In *LDOW*, 2009.

Sen Hu, Lei Zou, Jeffrey Xu Yu, Haixun Wang, and Dongyan Zhao. Answering natural language questions by subgraph matching over knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, 30(5): 824–837, 2018.

Yunshi Lan, Shuohang Wang, and Jing Jiang. Multi-hop knowledge base question answering with an iterative sequence matching model. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 359–368, 2019. doi: 10.1109/ICDM.2019.00046.

Morgane Laouenan, Palaash Bhargava, Jean-Benoît Eyméoud, Olivier Gergaud, Guillaume Plique, and Etienne Wasmer. A cross-verified database of notable people, 3500bc-2018ad. *Scientific Data*, 9(1):290, 2022.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

Qingxia Liu, Gong Cheng, Kalpa Gunaratna, and Yuzhong Qu. ESBM: an entity summarization benchmark. In *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17*, pages 548–564. Springer, 2020.

Qingxia Liu, Gong Cheng, Kalpa Gunaratna, and Yuzhong Qu. Entity summarization: State of the art and future challenges. *Journal of Web Semantics*, 69:100647, 2021.

Nan Ma, Jiancheng Guan, and Yi Zhao. Bringing pagerank to the citation analysis. *Information Processing & Management*, 44(2):800–810, 2008.

Karl Pearson. The problem of the random walk. *Nature*, 72(1867):342–342, 1905.

Hadas Raviv, Oren Kurland, and David Carmel. Document retrieval using entity-based language models. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 65–74, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340694. doi: 10.1145/2911451.2911508.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019. URL `https://api.semanticscholar.org/CorpusID:203626972`.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 697–706, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936547. doi: 10.1145/1242572.1242667.

Andreas Thalhammer, Nelia Lasierra, and Achim Rettinger. Linksum: Using link analysis to summarize entity data. In *ICWE*, pages 244–261, 2016.

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

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Dkn: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, page 1835–1844. International World Wide Web Conferences Steering Committee, 2018. ISBN 9781450356398. doi: 10.1145/3178876.3186175.

319 # 6 Appendix

Table 3: Generated Datasets in terms of number of entities $|\mathcal{V}|$, triples $|\mathcal{E}|$, ground-truth summaries, density as $|\mathcal{E}|/\binom{|\mathcal{V}|}{2}$, graph connectivity, number of components, sampling method to select the entities and the subgraph, minimum and maximum node degree and, running time.

(a) Small Datasets

Metric	WikiLitArt	WikiCinema	WikiPro	WikiProFem
Entities ($\|\mathcal{V}\|$)	85 346	70 753	79 825	79 926
Relations ($\|\mathcal{E}\|$)	136 950	126 915	125 912	123 193
Target Entities	494	493	493	468
Density	0.000018	0.000018	0.000019	0.000019
Sampling method	Random Walk	Random Walk	Random Walk	Random Walk
Connected-graph	Yes	Yes	Yes	Yes
Num-comp	1	1	1	1
Min Degree	1	1	1	1
Max Degree	2172	3005	2060	3142
Run-time (seconds)	91.934	118.014	126.119	177.63

(b) Medium Datasets

Metric	WikiLitArt	WikiCinema	WikiPro	WikiProFem
Entities ($\|\mathcal{V}\|$)	128 061	101 529	119 305	122 728
Relations ($\|\mathcal{E}\|$)	220 263	196 061	198 663	196 838
Target Entities	494	493	493	468
Density	0.000013	0.000019	0.000014	0.000013
Sampling method	Random Walk	Random Walk	Random Walk	Random Walk
Connected-graph	Yes	Yes	Yes	Yes
Num-comp	1	1	1	1
Min Degree	1	1	1	1
Max Degree	3726	5124	3445	5282
Run-time (seconds)	155.36	196.413	208.157	301.718

(c) Large Datasets

Metric	WikiLitArt	WikiCinema	WikiPro	WikiProFem
Entities ($\|\mathcal{V}\|$)	239 491	185 098	230 442	248 012
Relations ($\|\mathcal{E}\|$)	466 905	397 546	412 766	413 895
Target Entities	494	493	493	468
Density	0.000008	0.00001	0.000008	0.000007
Sampling method	Random Walk	Random Walk	Random Walk	Random Walk
Connected-graph	Yes	Yes	Yes	Yes
Num-comp	1	1	1	1
Min Degree	1	1	1	1
Max Degree	8599	12189	7741	12939
Run-time (seconds)	353.113	475.679	489.409	768.99

Dataset	Seed Nodes Categories
WikiLitArt	**Entire graph:** actor=150, composer=35, film=41, novelist=24, painter=59, poet=39, screenwriter=17, singer=72, writer=57
	Train: actor=105, composer=24, film=29, novelist=17, painter=42, poet=27, screenwriter=12, singer=50, writer=40
	Val: actor=23, composer=5, film=6, novelist=4, painter=9, poet=6, screenwriter=2, singer=11, writer=8
	Test: actor=22, composer=6, film=6, novelist=3, painter=8, poet=6, screenwriter=3, singer=11, writer=9
WikiCinema	**Entire graph:** actor=405, film=88
	Train: actor=284, film=61
	Val: actor=59, film=14
	Test: actor=62, film=13
WikiPro	**Entire graph:** actor=58, football=156, journalist=14, lawyer=16, painter=23, player=25, politician=125, singer=27, sport=21, writer=28
	Train: actor=41, football=109, journalist=10, lawyer=11, painter=16, player=17, politician=87, singer=19, sport=15, writer=20
	Val: actor=9, football=23, journalist=2, lawyer=3, painter=3, player=4, politician=19, singer=4, sport=3, writer=4
	Test: actor=8, football=24, journalist=2, lawyer=2, painter=4, player=4, politician=19, singer=4, sport=3, writer=4
WikiProFem	**Entire graph:** actor=141, athletic=25, football=24, journalist=16, painter=16, player=32, politician=81, singer=69, sport=18, writer=46
	Train: actor=98, athletic=18, football=17, journalist=9, painter=13, player=22, politician=57, singer=48, sport=14, writer=34
	Val: actor=21, athletic=4, football=3, journalist=4, painter=1, player=5, politician=13, singer=11, sport=1, writer=5
	Test: actor=22, athletic=3, football=4, journalist=3, painter=2, player=5, politician=11, singer=10, sport=3, writer=7

Table 4: Seed nodes categories for each dataset. "Entire graph" refers to using the seed nodes and generating the data without train-test-val splits. In train-test-val, each of the datasets is a single weakly connected graph.

Table 5 presents the versions of the technologies and configurations that we use in this work.

Table 5: Technology and Configuration Details for Daatset Generations

(a) Technologies Used: Software Versions and Data Sources

Technology	Version/Details
Java	Version 21
Spring Boot	Version 3
Docker	Version 24.0.8
Python	Version 3.10
PostgreSQL	Version 16.3
Neo4j	Version 5.20.0-community
Wikipedia XML Article Dump Files	Published by Wikimedia on 2023/05/01
Wikidata XML Article Dump Files	Published by Wikimedia on 2023/05/01

(b) Pre-processing Setup: Specifications of the AWS EC2 Instance (r5a.4xlarge) Used for Dataset Pre-processing

Specification	Details
vCPU	16 (AMD EPYC 7571, 16 MiB cache, 2.5 GHz)
Memory	128 GB (DDR4, 2667 MT/s)
Storage	500 GB (EBS, 2880 Max Bandwidth)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: To support our claims in the introduction and abstract, we provide experiments in section 4

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

   Justification: We do not have this information because assessing the limitations of a dataset can be challenging. One clear limitation is that our data is generated from an encyclopedic knowledge graph, and we are uncertain about its suitability for specific domains. However, we have made a concerted effort to diversify the topics covered.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer:[N/A]

   Justification: We do not have proofs, as our focus is on empirical evaluations. We compare our dataset with real-world data in section 4.

   Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The model, dataset, and instructions for running the models are available in our GitHub repository which is public.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification:The model, dataset, and instructions for running the models are available in our GitHub repository which is public.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyperparameters and data splits are explained both in the paper and our github repository. We detail the characteristics of the methods and provide implementations.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We did not repeat the experiments to report the confidence intervals.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: In 3.4 and the appendix, we provide information about the running time for producing datasets. Additionally, our GitHub repository contains detailed information about the technologies we used.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer:[Yes]

   Justification:We have reviewed the NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper does not involve societal impacts as it primarily focuses on foundational research in knowledge graph entity summarization, without direct application to scenarios that would cause societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper poses no risks associated with the release of data or models, as it focuses on foundational research in knowledge graph entity summarization without generating or releasing high-risk data or models.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the paper and re-implement the techniques.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide our Github repository and datasets publicly.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.