Leveraging Graph Structure in Seq2Seq Models for Knowledge Graph Link Prediction

Luu Huu Phuc[1,2], Ratan Bahadur Thapa[1], Mojtaba Nayyeri[1], Jingcheng Wu[1], Evgeny Kharlamov[2] and Steffen Staab[1,3]

[1]Analytic Computing, KI, University of Stuttgart, Stuttgart, Germany
[2]Bosch Center for Artificial Intelligence, Stuttgart, Germany
[3]WAIS, University of Southampton, United Kingdom

Abstract

We introduce Graph-Augmented Sequence-to-Sequence (GA-S2S), a novel framework that integrates a T5-small encoder-decoder with a Relational Graph Attention Network (RGAT) to improve link prediction in knowledge graphs. While existing Seq2Seq models rely solely on surface-level textual descriptions of entities and relations and at best, flatten the neighborhoods of a query entity into a single linear sequence, thereby discarding the inherent graph structure, GA-S2S jointly encodes both textual features and the full k-hop subgraph topology surrounding the query entity. By integrating raw encoder outputs with RGAT's relation-aware embeddings, our model captures and leverages richer multi-hop relational patterns and textual information. Our preliminary experiments on the CoDEx dataset demonstrate that GA-S2S outperforms competitive Seq2Seq-based baseline models, achieving up to a 19% relative gain in link prediction accuracy.

Keywords

Knowledge Graph, Link Prediction, Sequence-to-Sequence Model, Graph Neural Network

1. Introduction

Knowledge graphs (KGs) use graph-based data models to represent structured knowledge about real-world entities (e.g., people, places, or things) and their connectedness. They store facts about entities as triples (e, r, e'), where e and e' denote entities, and r represents their connectedness, also known as a relation. Reasoning over these structured facts is known to be useful for many artificial intelligence applications, ranging from question answering and recommendation systems to natural language processing [1, 2], and so on.

However, the structured facts stored in knowledge graphs in real-world applications often represent only a partial view of the world. This incompleteness arises from practical constraints such as limitations in data acquisition, privacy concerns, contractual restrictions on third-party data sharing, noisy or inaccurate observations (e.g.,, due to sensor, IoT, or edge device failures), and errors introduced during (human or procedural) data integration [3]. As a result, knowledge graphs, whether newly constructed from such data or pre-existing, inherently contain partial information about the world we want to represent, and it is therefore expected that many factual assertions about real-world entities may be either incomplete or entirely absent. For instance, 71% of person entities in Freebase lack a recorded place of birth, and 75% lack a recorded nationality [4]; moreover, between 69% and 99% of entities in YAGO, DBpedia, and similar KGs are missing at least one property that other entities within the same class possess [5]. Link prediction for KGs [6], a problem that is early studied under the umbrella of *statistical relational learning* [7], aims to infer these missing facts by predicting whether a triple is likely to be true, particularly, by estimating the likelihood of missing entities for a given incomplete triple assertion (i.e.,, triple query). Formally, given an incomplete triple assertion $(e, r, ?)$ or $(?, r, e)$, the

ESWC'26: 5th Workshop on LLM-Integrated Knowledge Graph Generation from Text (TEXT2KG), May 10-14, 2026, Dubrovnik, Croatia

huuphuc.luu@de.bosch.com (L. H. Phuc); ratan.thapa@ki.uni-stuttgart.de (R. B. Thapa); mojtaba.nayyeri@ki.uni-stuttgart.de (M. Nayyeri); jingcheng.wu@ki.uni-stuttgart.de (J. Wu); evgeny.kharlamov@de.bosch.com (E. Kharlamov); steffen.staab@ki.uni-stuttgart.de (S. Staab)

objective is to predict the missing entity e' such that the newly formulated triple (e, r, e') or (e', r, e) corresponds to a plausible missing fact in the KGs.

Conventional link prediction methods for KGs can be classified into *embedding-based* approach [8] and *Graph Neural Network* (GNN)-based approach [9]. Embedding-based methods assign distinct vectors to each entity and relation, employing scoring functions to evaluate triple plausibility [10, 11, 12]. Their parameter sizes scale linearly with the number of entities and relations, making them memory-intensive to train on large-scale KGs. Moreover, by treating each triple in isolation, static embeddings often fail to capture higher-order or multi-hop structures, thereby limiting their ability to generalize over broader subgraph patterns. GNN-based methods, on the other hand, address this limitation by leveraging the graph structure through iterative neighborhood message passing and aggregation [13, 14, 15]. Nonetheless, neither embedding-based nor GNN-based approaches are primarily designed to process unstructured information (i.e.,, textual descriptions, literal attributes, *etc*) without integrating specialized or auxiliary modules [16]. Both embedding and GNN approaches typically answer a query, for example, a head-relation query $(e, r, ?)$, by computing a score $f(e, r, e')$ for every tail entity candidate e' in the knowledge graph (KG) and then ranking them to select the top-k, a process that is $O(|E|)$ per query - where $|E|$ is the number of entities in the KG - and becomes computationally expensive for large KGs (i.e.,, $|E| \geq$ millions).

To address the computationally expensive inference for large KGs, recent generative approaches leveraging transformer-based Sequence-to-Sequence (Seq2Seq) models, such as KGT5 [17], KG-S2S [18], KGT5-context [19], and UniLP [20], have achieved state-of-the-art results on standard link-prediction benchmarks, such as Wikidata5M [21] and WikiKG90Mv2 [22]. These methods, by contrast to the embedding and GNN-based approaches, assume that entities and relations can be verbalized with unique text mentions and recast link prediction as a text-generation task. For example, given the verbalized forms "`Michael Jackson`", "`occupation`", and "`musician`" for a triple, the query $(e_{\text{Michael Jackson}}, r_{\text{occupation}}, ?)$ is reformulated as input sequence "`Predict tail: Michael Jackson | occupation`", and the Seq2Seq model is then trained to generate the target mention "`musician`". Since Seq2Seq-based methods employ a fixed parameter budget independent of the number of entities and relations, and replace the scoring-ranking step by directly generating the text mentions of target entities, they overcome the linear parameter growth and high inference cost in previous approaches. However, since these models operate solely on verbalized text mentions, they do not explicitly model topological patterns (e.g.,, multi-hop neighborhood), which trades off the possibility of exploiting subgraph structural information in KGs.

In this paper, we address this limitation of the Seq2Seq-based approach by proposing a hybrid model that integrates a Seq2Seq backbone based on the T5-small model [23], initialized without pre-trained weights, with a relation-aware GNN module, i.e.,, Relational Graph Attention Network (RGAT) [14]. Rather than constraining the input to a single flat sequence, our model is able to process the neighborhood topological structure of a query entity. The proposed model encodes each query and its full k-hop neighborhood into a shared latent space using the T5 encoder. The resulting representations are then enriched via relation-aware message passing over the neighborhood surrounding the query entity, using RGAT. Finally, the augmented representation is decoded by the T5-decoder to generate the verbalized text mention of the target entity. This design simultaneously leverages textual information as well as local graph structures. We benchmark our approach on the CoDEX dataset, which contains 3 KGs ranging from tens of thousands to hundreds of thousands of triples, and show that it outperforms competitive Seq2Seq-based baseline methods in link prediction accuracy.

2. Background

A knowledge graph G is a (multi)set of triples, each drawn from the Cartesian product $E \times R \times E$, where E and R are two disjoint sets of entities and relations. Each triple $(h, r, t) \in G$ represents the fact that a relation $r \in R$ exists between head entity $h \in E$ and tail entity $t \in E$.

Generative Se2Seq-based methods excel at leveraging textual data for link prediction [24]. They

reformulate queries such as $(e, r, ?)$ as text-generation tasks, i.e.,, generating the text mention $s(e')$ for a target entity e' that best completes the query triple (e, r, e'). It is therefore assumed that all entities e and all relations r in KGs are associated with a unique text mention, denoted by $s(e)$ and $s(r)$, respectively. That is, for any $e, e' \in E$ and $r, r' \in R$, we have $e = e' \leftrightarrow s(e) = s(e')$ and $r = r' \leftrightarrow s(r) = s(r')$.

KGT5 [17] is one of the pioneering works to cast knowledge graph link prediction as a Seq2Seq task, translating pairs of input queries and target entities into text sequences and training a T5-small model from scratch to predict the missing entity. Building on this, KG-S2S [18] unifies static, temporal, and inductive link prediction under a single Seq2Seq paradigm. KGT5-context [19] further enhances KGT5 by prepending linearized 1-hop neighbor triples of the query entity to the input sequence. For example, an input might read: "`Predict tail: Michael Jackson | occupation | Context: genre | pop music | record label | Sony Music | . . .`", where the "Context" segment verbalizes all 1-hop triples linked to "Michael Jackson". Going beyond 1-hop triples, UniLP [20] introduces soft prompts, learnable embeddings representing 1-hop relations, into each transformer layer. This approach allows the model to implicitly capture structural information, improving its ability to encode the topology of the knowledge graph without relying solely on textual context.

3. The GA-S2S Model

The GA-S2S model, as shown in Figure 1, is composed of 3 sequential components: a text encoder, an RGAT-based GNN module, and a text decoder. We employ the encoder and decoder from the T5-small model without pre-trained weights. We initialize the T5 model from scratch because pre-trained weights are heavily optimized for natural language syntax. Given that our inputs are highly structured, template-based verbalizations mixed with graph embeddings, training from scratch prevents potential negative transfer from natural language priors.

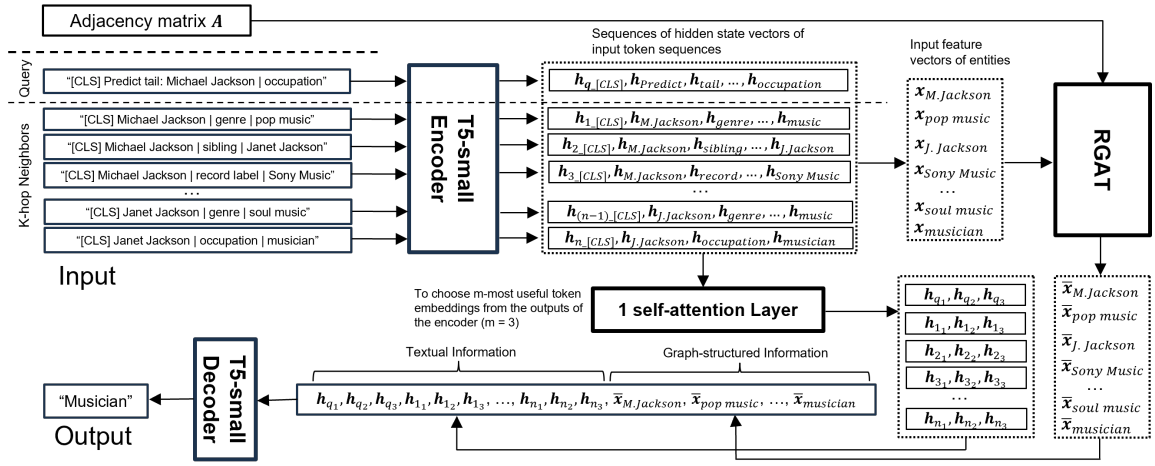

Figure 1: The proposed model architecture integrates an RGAT module between the encoder and decoder of the T5-small model, enabling the fusion of textual and structural information from KGs. All components of the T5 model and the RGAT module are fully trainable and initialized from scratch, and no pre-trained weights are frozen during training.

Input Representation. Given a query $q = (e, r, ?)$, the input to our model consists of the text sequence $s(q)$ corresponding to the query, a set of text sequences $s(G_{e,k})$ of a sampled k-hop subgraph $G_{e,k}$ of the query entity e, and the associated adjacency matrix $\mathbf{A}$ of $G_{e,k}$.

We first sample a k-hop subgraph $G_{e,k} = \{(h, r, t) \in G \mid h \in E_{e,k} \wedge t \in E_{e,k}\}$ of the query entity e and integer $k \geq 0$, where $E_{e,k} = \{e' \mid$ there exists a path of length $\leq k$ between e and e' in $G\}$. Each triple in the induced subgraph $G_{e,k} \subseteq G$ of the original G under the query q is then verbalized as, for

any $i \leq n = |G_{e,k}| = |\{(h,r,t) \in G \mid h \in E_{e,k} \wedge t \in E_{e,k}\}|$,

$$s(h_i, r_i, t_i) = \text{“[CLS]”}s(h_i)\text{“} \mid \text{”}s(r_i)\text{“} \mid \text{”}s(t_i),$$

where “[CLS]” denotes a special token of the Seq2Seq model. The connectivity of $G_{e,k}$, i.e., its adjacency matrix $\mathbf{A}$, is encoded via 2 tensors: a $n \times 2$ tensor of node-index pairs $[[h_1, t_1], \ldots, [h_n, t_n]]$, and a length-$n$ tensor of relation indices $[r_1, \ldots, r_n]$.

Encoding and Subgraph Representation. All text sequences $s(q) \cup s(G_{e,k})$ are fed to the T5-encoder, producing a sequence of hidden state vectors $H = \{\mathbf{h}_{\texttt{tok}_1}, \ldots, \mathbf{h}_{\texttt{tok}_l}\}$ for each input text sequence, where $l \leq 512$. Since each entity $e' \in E_{e,k}$ in the sampled subgraph $G_{e,k}$ can appear in one or more triples, we aggregate its representation by averaging over the $\mathbf{h}_{\texttt{[CLS]}}$ vectors of all triples it participates in, denoted by $\mathbf{x}_{e'}$:

$$\mathbf{x}_{e'} = \frac{1}{|G_{e',k}|} \sum_{(h,r,t) \in G_{e',k}} \mathbf{h}_{\texttt{[CLS]}}^{(h,r,t)},$$

where $G_{e',k} = \{(h,r,t) \in G_{e,k} \mid h = e' \vee t = e'\}$ and $\mathbf{h}_{\texttt{[CLS]}}^{(h,r,t)}$ is the hidden state vector of “[CLS]” token of $s(h,r,t)$. We then pass the set of entity features $\{\mathbf{x}_{e'}\}$ and adjacency matrix $\mathbf{A}$ into the RGAT module, which propagates and aggregates information across the subgraph, yielding updated embeddings $\{\overline{\mathbf{x}}_{e'}\}$.

Decoding with Combined Representations. We add a skip connection from the encoder to the decoder so that the decoder receives both the raw encoder output and graph-enhanced embeddings from the GNN module. For each input sequence $s \in \{s(q)\} \cup s(G_{e,k})$, the encoder produces a corresponding sequence of token-level hidden state vectors H. We apply a lightweight self-attention layer to distill H into the m most salient vectors $\{\mathbf{h}_1, \ldots, \mathbf{h}_m\}$, as not all tokens in s contain useful information. Empirically, we find that setting $m = 3$ provides a good balance between performance and efficiency. The decoder input is the concatenation of these distilled vectors for the query and its k-hop neighborhood triples with the RGAT's output node features $\{\overline{\mathbf{x}}_{e'} \mid e' \in E_{e,k}\}$.

The entire pipeline is trained end-to-end by minimizing the cross-entropy loss between the decoder's generated token distributions and the ground-truth entity mentions.

4. Experimental Settings

We use CoDEx dataset [25] in our experiments, which is extracted from Wikidata and Wikipedia. Unlike earlier benchmarks, CoDEx is particularly designed to mitigate train/test leakage and avoid trivial relational patterns. CoDEx also provides multilingual textual labels for entities and relations; in our experiments, we use English textual labels as the text mentions. Detailed statistics of the dataset are shown in Table 1.

We train our model with a batch size of 64 and set all hidden dimensions to 512. We tune the RGAT dropout rate over $\{0.0, 0.1, 0.2, 0.5\}$ and the number of attention heads over $\{1, 2, 4\}$ via grid search. Our RGAT implementation is based on *PyTorch Geometric* [26], and we use the T5-small encoder, decoder, and transformer layers from *HuggingFace* [27]. For each query, we sample a 1-hop neighborhood of size 75 for CoDEx-L, and for CoDEx-S/M, we use neighborhood sizes of 75 for 1-hop and (10, 5) for 2-hop neighborhoods. We restrict to 2-hop neighborhoods only for the smaller KGs due to the computational limitations of our current model implementation.

We evaluate using Mean Reciprocal Rank (MRR) and Hit@k ($k = 1, 3, 10$), following the standard scoring and ranking procedure described in [17]. For baseline comparison, we evaluate against the current state-of-the-art generative models for KG link prediction, namely KGT5 [17], KG-S2S [18], and KGT5-context [19]. UniLP [20] is not included in our evaluation due to the unavailability of its codebase.

Table 1
Statistics of the 3 KGs in CoDEx dataset

	CoDEx-S	CoDEx-M	CoDEx-L
#Entity	2034	17050	77951
#Relation	42	51	69
Train	32888	185584	551193
Valid	1827	10310	30622
Test	1828	10311	30622
Avg Deg	32.3	21.7	14.1
Median Deg	17	12	7
Density	0.0159	0.0012	0.0002

5. Overall Evaluation

Table 2 shows link prediction performance on the CoDEx dataset. At first glance, GA-S2S outperforms all other baselines on CoDEx-M/L and ranks second on CoDEx-S. GA-S2S with 1-hop neighborhoods improves the MRR scores over KGT5-context (also using 1-hop neighborhoods) by 11.2%, 4.8%, and 1.7% on CoDEx-S/M/L respectively. Incorporating 2-hop neighborhoods further increases MRR by 19.5% on CoDEx-S and 6.6% on CoDEX-M. While existing Seq2Seq methods are constrained to 1-hop (flattened) neighborhoods, these results suggest that k-hop neighborhoods with $k \geq 2$ are advantageous for the task. The improvement is most noticeable on CoDEx-S and decreases as the KG size increases. This is expected since the bigger KGs are much sparser with lower degrees of connectivity as shown in Table 1.

Table 2
Comparison of link prediction performance on CoDEx dataset. The best results are written in bold. The second-best results are underlined. All results are averaged over 4 runs. Comparison is done using a t-test with p-value = 0.01.

	CoDEx-S				CoDEx-M				CoDEx-L			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
KG-S2S	0.269	0.182	0.314	0.454	0.240	0.175	0.267	0.376	0.235	0.181	0.260	0.343
KGT5	**0.344**	**0.269**	**0.387**	<u>0.492</u>	0.276	0.214	0.314	0.398	0.267	0.212	0.299	0.375
KGT5-context	0.277	0.186	0.313	0.471	0.271	0.206	0.304	0.402	<u>0.292</u>	<u>0.236</u>	<u>0.321</u>	<u>0.400</u>
GA-S2S (1-hop)	0.308	0.210	0.359	**0.512**	<u>0.284</u>	<u>0.219</u>	<u>0.317</u>	<u>0.415</u>	**0.297**	**0.241**	**0.327**	**0.404**
GA-S2S (2-hop)	<u>0.331</u>	<u>0.252</u>	<u>0.381</u>	0.489	**0.289**	**0.223**	**0.322**	**0.421**	-	-	-	-

Although KGT5-context extends KGT5 with 1-hop neighborhood information, its performance is lower than that of KGT5 on the CoDEX-S/M datasets. This suggests that naively appending linearized 1-hop neighborhoods of query entities to input sequences can negatively impact model performance.

It is equally noteworthy that KGT5, even without any explicit neighborhood graph structure input, remains the top performer on CoDEx-S. We hypothesize that this is due to the excessively long and potentially redundant decoder input sequences in both our model and KGT5-context. Even when limiting each neighbor triple to three embedding vectors, large neighborhoods can still result in unwieldy decoder inputs in GA-S2S. As discussed above, flattening all information into one unstructured sequence may confuse the decoder and impede correct generation. A decoder redesign that accepts structured input could therefore yield further performance gains.

6. Limitations

While Se2Seq models often outperform embedding methods on large-scale KGs with millions of entities [17, 19], they do not consistently exhibit a performance advantage over embedding methods on small- to mid-sized KGs. This is reflected in the CoDEx benchmark: GA-S2S, along with other existing

Seq2Seq-based models, lags behind leading KG embedding methods. For instance, ComplEx [12] achieves MRR scores of up to 0.465 and 0.337 on CoDEx-S and CoDEx-M, respectively, and TuckER [28] achieves an MRR score of 0.309 on CoDEx-L. Nevertheless, the performance gap between GA-S2S and embedding-based methods narrows as the KG size increases from CoDEx-S to CoDEx-L.

Additionally, GA-S2S currently incurs higher computational costs than standard Seq2Seq models and faces scalability challenges on very large KGs. This overhead arises from the richer processing of neighborhood structures via the GNN module, which leads to a higher number of processed input tokens per query (up to $512\times$ neighborhood size for GA-S2S versus only up to 512 for baselines). However, this complexity directly contributes to the model's ability to jointly capture graph topology with textual information. Thus, our preliminary exploration with the GNN module establishes a basic hybrid Seq2Seq architecture for further improvement. In particular, incorporating graph transformers [29, 30], whose designs are more compatible with transformer-based encoders and decoders, might offer a promising direction. By natively processing graph-structured data via topology-aware attention mechanisms, graph transformers can inherently avoid the need to flatten neighborhoods into linear sequences, thereby enabling deeper integration of structural and textual data.

7. Conclusion and Open Questions

We have introduced GA-S2S, a hybrid Seq2Seq-GNN framework that directly integrates graph structure into generative Seq2Seq link prediction models. In contrast to prior Seq2Seq methods that simply linearize neighborhood triples into input sequences, GA-S2S employs a relation-aware GNN (RGAT) to encode both local neighborhood triples and k-hop subgraph topology, enabling richer multi-hop relational reasoning. On the CoDEx benchmarks, our approach consistently outperforms competitive Seq2Seq baselines, particularly when incorporating 2-hop neighborhood, demonstrating the clear advantages of explicitly encoding graph structure.

Despite these promising results, several open questions remain for advancing hybrid Seq2Seq models, as highlighted in Sect. 6. First, how can more balanced and rigorously evaluated neighborhood sampling strategies be designed to manage computational overhead without sacrificing structural context? Second, could efficient encoder-decoder architectures, such as hierarchical transformers [31], serve as a more scalable backbone for massive KGs? Finally, what are the optimal techniques for seamlessly integrating textual and structural information throughout both the encoding and decoding stages? Addressing these questions will be critical for the next generation of knowledge graph reasoning models.

Acknowledgments

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Jingcheng Wu. Jingcheng Wu and Ratan Bahadur Thapa has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - SFB 1574 - Project number 471687386.

Declaration on Generative AI

During the preparation of this work, the authors used Gemini in order to: Grammar and spelling check. Specifically, GenAI tools were used only for minor improvements to the authors' written text. After using these tools, the authors reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] H. Zhou, L. Halilaj, S. Monka, S. Schmid, Y. Zhu, J. Wu, N. Nazer, S. Staab, Seeing and knowing in the wild: Open-domain visual entity recognition with large-scale knowledge graphs via contrastive learning, in: AAAI, AAAI Press, 2026, pp. 13638–13646. doi:10.1609/AAAI.V40I16.38370.

[2] Z. Ding, J. Wu, J. Wu, Y. Xia, B. Xiong, V. Tresp, Temporal fact reasoning over hyper-relational knowledge graphs, in: EMNLP (Findings), Findings of ACL, Association for Computational Linguistics, 2024, pp. 355–373. doi:10.18653/V1/2024.FINDINGS-EMNLP.20.

[3] Y. Zhu, J. Wu, Y. Wang, H. Zhou, J. Chen, E. Kharlamov, S. Staab, Certainty in uncertainty: Reasoning over uncertain knowledge graphs with statistical guarantees, in: EMNLP, Association for Computational Linguistics, 2025, pp. 8730–8752. doi:10.18653/V1/2025.EMNLP-MAIN.441.

[4] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, Knowledge vault: A web-scale approach to probabilistic knowledge fusion, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 601–610.

[5] S. Razniewski, F. Suchanek, W. Nutt, But what do we actually know?, in: Proceedings of the 5th Workshop on Automated Knowledge Base Construction, 2016, pp. 40–44.

[6] S. M. Kazemi, D. Poole, Simple embedding for link prediction in knowledge graphs, Advances in neural information processing systems 31 (2018).

[7] L. Getoor, B. Taskar, Introduction to statistical relational learning, MIT press, 2007.

[8] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, P. Merialdo, Knowledge graph embedding for link prediction: A comparative analysis, ACM Trans. Knowl. Discov. Data 15 (2021). URL: https://doi.org/10.1145/3424672. doi:10.1145/3424672.

[9] Z. Ye, Y. J. Kumar, G. O. Sing, F. Song, J. Wang, A comprehensive survey of graph neural networks for knowledge graphs, IEEE Access 10 (2022) 75729–75741. doi:10.1109/ACCESS.2022.3191784.

[10] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Curran Associates Inc., Red Hook, NY, USA, 2013, p. 2787–2795.

[11] M. Nickel, V. Tresp, H.-P. Kriegel, A three-way model for collective learning on multi-relational data, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML'11, Omnipress, Madison, WI, USA, 2011, p. 809–816.

[12] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: M. F. Balcan, K. Q. Weinberger (Eds.), Proceedings of The 33rd International Conference on Machine Learning, volume 48 of *Proceedings of Machine Learning Research*, PMLR, New York, New York, USA, 2016, pp. 2071–2080. URL: https://proceedings.mlr.press/v48/trouillon16.html.

[13] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, M. Alam (Eds.), The Semantic Web, Springer International Publishing, Cham, 2018, pp. 593–607.

[14] D. Busbridge, D. Sherburn, P. Cavallo, N. Y. Hammerla, Relational graph attention networks, CoRR abs/1904.05811 (2019). URL: http://arxiv.org/abs/1904.05811. arXiv:1904.05811.

[15] S. Vashishth, S. Sanyal, V. Nitin, P. Talukdar, Composition-based multi-relational graph convolutional networks, in: International Conference on Learning Representations, 2020. URL: https://openreview.net/forum?id=BylA_C4tPr.

[16] F. Lu, P. Cong, X. Huang, Utilizing textual information in knowledge graph embedding: A survey of methods and applications, IEEE Access 8 (2020) 92072–92088. doi:10.1109/ACCESS.2020.2995074.

[17] A. Saxena, A. Kochsiek, R. Gemulla, Sequence-to-sequence knowledge graph completion and question answering, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),

Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2814–2828. URL: https://aclanthology.org/2022.acl-long.201/. doi:10.18653/v1/2022.acl-long.201.

[18] C. Chen, Y. Wang, B. Li, K.-Y. Lam, Knowledge is flat: A Seq2Seq generative framework for various knowledge graph completion, in: N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, S.-H. Na (Eds.), Proceedings of the 29th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 4005–4017. URL: https://aclanthology.org/2022.coling-1.352/.

[19] A. Kochsiek, A. Saxena, I. Nair, R. Gemulla, Friendly neighbors: Contextualized sequence-to-sequence link prediction, in: B. Can, M. Mozes, S. Cahyawijaya, N. Saphra, N. Kassner, S. Ravfogel, A. Ravichander, C. Zhao, I. Augenstein, A. Rogers, K. Cho, E. Grefenstette, L. Voita (Eds.), Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 131–138. URL: https://aclanthology.org/2023.repl4nlp-1.11/. doi:10.18653/v1/2023.repl4nlp-1.11.

[20] B. Liu, M. Peng, W. Xu, X. Jia, M. Peng, Unilp: Unified topology-aware generative framework for link prediction in knowledge graph, in: Proceedings of the ACM Web Conference 2024, WWW '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 2170–2180. URL: https://doi.org/10.1145/3589334.3645592. doi:10.1145/3589334.3645592.

[21] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, J. Tang, KEPLER: A unified model for knowledge embedding and pre-trained language representation, Transactions of the Association for Computational Linguistics 9 (2021) 176–194. URL: https://aclanthology.org/2021.tacl-1.11/. doi:10.1162/tacl_a_00360.

[22] W. Hu, M. Fey, H. Ren, M. Nakata, Y. Dong, J. Leskovec, Ogb-lsc: A large-scale challenge for machine learning on graphs, arXiv preprint arXiv:2103.09430 (2021).

[23] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020).

[24] Z. Ding, Y. Li, Y. He, A. Norelli, J. Wu, V. Tresp, M. M. Bronstein, Y. Ma, Dygmamba: Efficiently modeling long-term temporal dependency on continuous-time dynamic graphs with state space models, Trans. Mach. Learn. Res. 2025 (2025). URL: https://openreview.net/forum?id=sq5AJvVuha.

[25] T. Safavi, D. Koutra, CoDEx: A Comprehensive Knowledge Graph Completion Benchmark, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 8328–8350. URL: https://aclanthology.org/2020.emnlp-main.669/. doi:10.18653/v1/2020.emnlp-main.669.

[26] M. Fey, J. E. Lenssen, Fast graph representation learning with PyTorch Geometric, in: ICLR Workshop on Representation Learning on Graphs and Manifolds, 2019.

[27] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Q. Liu, D. Schlangen (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://aclanthology.org/2020.emnlp-demos.6/. doi:10.18653/v1/2020.emnlp-demos.6.

[28] I. Balazevic, C. Allen, T. Hospedales, TuckER: Tensor factorization for knowledge graph completion, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5185–5194. URL: https://aclanthology.org/D19-1522/. doi:10.18653/v1/D19-1522.

[29] J. Yang, Z. Liu, S. Xiao, C. Li, D. Lian, S. Agrawal, A. Singh, G. Sun, X. Xie, Graphformers: Gnn-nested transformers for representation learning on textual graph, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 28798–28810. URL: https://proceedings.neurips.cc/

paper_files/paper/2021/file/f18a6d1cde4b205199de8729a6637b42-Paper.pdf.

[30] J. Liu, Q. Mao, W. Jiang, J. Li, Knowformer: revisiting transformers for knowledge graph reasoning, in: Proceedings of the 41st International Conference on Machine Learning, ICML'24, JMLR.org, 2024.

[31] P. Nawrot, S. Tworkowski, M. Tyrolski, L. Kaiser, Y. Wu, C. Szegedy, H. Michalewski, Hierarchical transformers are more efficient language models, in: M. Carpuat, M.-C. de Marneffe, I. V. Meza Ruiz (Eds.), Findings of the Association for Computational Linguistics: NAACL 2022, Association for Computational Linguistics, Seattle, United States, 2022, pp. 1559–1571. URL: https://aclanthology.org/2022.findings-naacl.117/. doi:10.18653/v1/2022.findings-naacl.117.