StATIK+: Structure and Text for Inductive Knowledge Graph Modeling and Paths towards Enterprise Implementations

Elan S. Markowitz[1], Aram Galstyan[1]

[1]*University of Southern California - Information Sciences Institute*

Abstract

While many enterprise knowledge Graphs (KGs) are updated frequently, most KG models require retraining to incorporate these updates. Inductive models are able to adapt to new edges and entities in the KG. This extended abstract presents a prior work **StATIK**–**St**ructure **A**nd **T**ext for **I**nductive **K**nowledge Completion– **and** a roadmap towards industry implementations. StATIK uses a Language Model to extract the semantic information from text descriptions, while using Message Passing Neural Networks to capture the structural information in the graph. While StATIK was evaluated for inductive knowledge graph completion, many applications have different end tasks. This work provides background and a roadmap of some of the opportunities in applying StATIK to industry tasks.

Keywords

Large Language Models, LLM, Knowledge Graphs, KG, Enterprise Knowledge Graphs

Knowledge graphs (KGs) are used to represent knowledge across many domains. These domains include commonsense reasoning [1, 2, 3], question answering [4, 5, 6, 7, 8, 9], recommendation systems [10, 11, 12, 13], and many others [14]. In knowledge graphs, the nodes, called *entities*, often possess textual descriptions, while edges are typically labeled with one of many *relation* types, which may also possess textual descriptions. Effective KG models should learn to leverage this textual information in order to correctly complete the knowledge base. Additionally, such knowledge graphs are usually dynamic [15, 16] as a result of the underlying knowledge base being dynamic. Meaning, entities and edges are frequently added and removed from the knowledge graph. Thus, another quality we desire of knowledge graph models, is that they be inductive and generalize to unseen entities.

StATIK [17] is a completely inductive, hybrid model that effectively leverages both the structure of a knowledge graph as well as the underlying textual descriptions of the entities and relations. Structure is incorporated through a Message Passing Neural Network (MPNN) [18] that aggregates information from a neighborhood defined around each entity, while textual information is incorporated through a pretrained language model[19]. A high level depiction of StATIK is provided in Figure 1. While proven in inductive knowledge graph completion, this paper proposes paths to applying StATIK in other industrial tasks.

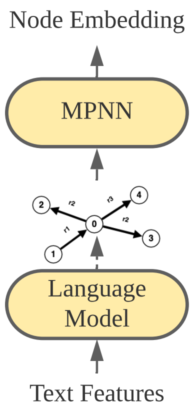

Figure 1: StATIK takes as input text features for a centroid node ⓪ around which a local subgraph has been sampled. A fine-tuned language model encodes the text features for the centroid node while the representations for neighboring nodes come from a frozen pretrained language model (preprocessing step). A message passing neural network is applied to generate the final representation for the centroid node.

1. Inductive Representation Learning on Knowledge Graphs

We can define a knowledge graph with textual information as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{D})$ where $\mathcal{E}$ is the set of entities, $\mathcal{R}$ is the set of relation types, $\mathcal{T}$ is the set of triples $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, and $\mathcal{D}$ is the set of entity and relation descriptions. The inductive knowledge graph modeling task is defined as follows. Let the training graph be $\mathcal{G}_{train} = (\mathcal{E}_{train}, \mathcal{R}, \mathcal{T}_{train}, \mathcal{D}_{train})$ where $\mathcal{E}_{train}$ is a subset of $\mathcal{E}$, $\mathcal{D}_{train}$ is the corresponding subset of $\mathcal{D}$, and $\mathcal{T}_{train}$ is the subset of $\mathcal{T}$ containing triples only involving entities

---

CIKM 2023 Workshop on Enterprise Knowledge Graphs Using Large Language Models

*Corresponding author.
esmarkow@usc.edu (E. S. Markowitz)
https://elanmarkowitz.github.io/ (E. S. Markowitz)

Table 1
Related works comparison table (reproduced from [17]). N is number of entities, Q is number of queries, R is number of relation types. †Model uses domain adaptation but does not train end-to-end. References are TransE [20], OpenWorld[21], Glove-DKRL[22], Commonsense[23], IndTransE[24], LAN[25], GralL[26], KGBert[27], BLP[28], StAR[29]

Model	TransE	OpenWorld	Glove-DKRL	Commonsense	IndTransE	LAN	GraIL	KGBert	BLP	StAR	**ours**
Inductive - Seen2Unseen	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
Inductive - Unseen2Unseen	✗	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓
End-to-end LM	✗	✗	✗	✗†	✗	✗	✗	✓	✓	✓	✓
No Support Set Required	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
Graph features	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✓
Structure Objective	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
Inference Scalability	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(NQ)$	$\mathcal{O}(NQ)$	$\mathcal{O}(N)$	$\mathcal{O}(NR+Q)$	$\mathcal{O}(N+Q)$

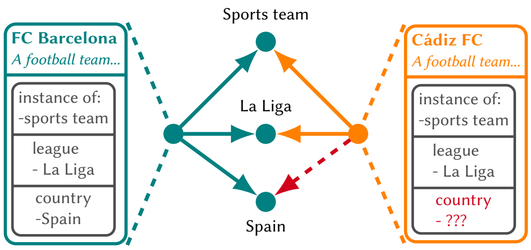

Figure 2: Problem addressed by inductive learning (reproduced from [17]). During training, only the **blue** portion of the graph exists, including the entities **FC Barcelona**, **sports team**, **La Liga**, and **Spain**. Later, the entity **Cádiz FC** is added to the graph. When added, an entity contains a description and some number of edges (possibly zero). Since StATIK is inductive, it requires no retraining or retroactive processing in any way to make predictions about Cádiz FC. This could include predicting Cádiz FC's country i.e. the query (Cádiz FC, country, ?). The correct prediction, (Cádiz FC, country, Spain), is displayed in dashed red.

in $\mathcal{E}_{train}$. For any KG task, the goal is to make predictions on $\mathcal{G}$ having only trained the model on $\mathcal{G}_{train}$. StATIK was trained and evaluated on a knowledge graph completion task. Figure 2 demonstrates a motivating example.

In order to learn the inductive objective, StATIK uses text features instead of the commonly used embedding tables and extends prior work by also incorporating structural information through message passing neural networks, a type of graph neural network.

2. Related Work

Much of the work in the area of KG modeling has focused on the transductive setting i.e. making predictions on entities seen at training time. Generally, these methods learn simple entity embeddings in a geometric space such as TransE [20], ComplEx [30], DistMult [31], RotatE [32], and SimplE [33], or through a machine learning decoder such as ConvE and HypER [34, 35]. There has also been effort in using graph neural networks for knowledge graph completion. R-GCN [36] brings the original GCN [37] to the multi-relational knowledge graph setting. Wang et al. [38] looked at using a modified version of GAT [39] to get strong results in the transductive setting.

2.1. Inductivity

Recently, there has been increased focus on the inductive setting. Some works learn embeddings for new entities by translating from existing entities in the training graph [25, 24, 40, 41]. This requires a sufficient number of edges from nodes seen during training to the new nodes (**seen-to-unseen**). Other methods [26, 42], have been able to achieve inductivity without such requirements, and as a result, can operate on **unseen-to-unseen** entities. There have also been some works on open domain knowledge graph completion, a similar learning task [21, 43]. Some of their techniques, such as using text to enable generalization to new entities, have continued in the works analyzed here.

2.2. Language Models

Transformers [44], have created a renaissance in language modeling over recent years. Combined with self-supervised pretraining, language models are able to capture the contextual and semantic information of natural language [19].

As many KGs contain text associated with each entity, researchers have sought to use that information for improved performance or inductivity. KGBert [27] looked into using transformers for link prediction, treating it as a text classification task. Bert for Link Prediction (BLP) [28] and StAR [29] have sought to incorporate language models while improving on some of the flaws of KGBert. Commonsense [23] initializes an embedding table using the language model before training with a message passing neural network (similar but not inductive). Older model DKRL [22] uses a simpler language model with GloVe embeddings [45].

2.3. Structural Objective/Graph Features

Most KG completion models use some form of structural objective; The scoring function uses spatial or geometric transformations to learn the graph structure.

Structural objectives alone have some limitations with regard to capturing graph structure. Being able to explicitly use the local graph structure and topology as a feature (through message passing) is beneficial for both general performance and inductivity. Many of the models mentioned [23, 24, 25, 26] make use of such features.

2.4. Scalability

Scalability is incredibly important for KG models as enterprise knowledge graphs can include millions to billions of entities and edges. When dealing with complex encoders such as MPNNs or LMs, the number of encoder passes becomes an especially pressing issue.

In KG completion, every entity is considered a possible solution to a query. This is one of the more challenging tasks with regard to scalability. If each of the possible triples is evaluated independently, the problem becomes a combinatorial mess. This is the case for KGBert [27] and GraiL [26] which can only evaluate a single triple at a time. This makes the task quadratic in complexity.

There are other scalability issues that arise when combining language models with graph models. Namely, the neighborhood for the message passing neural network contains many nodes, each with associated text. Encoding each of the neighbors with a language model at inference time would be costly when only a single entity encoding is used in the final output. `StATIK` solves this by separating the language encoding of the neighbors into a preprocessing step and only applying the finetuned language model to the entity being encoded.

In addition, the size of the neighborhood of some entities makes it infeasible to process the entire neighborhood at once. `StATIK` relies on frameworks for efficiently sampling neighborhoods at inference time [46].

Table 1 gives a comparison of the most relevant related works. `StATIK` has many advantages in terms of the factors discussed.

3. Toward Enterprise Implementation

There are many enterprise domains in which KG entities are associated with text descriptions including biomedical [47], chemistry [48], geological [49], financial [50], and many others [51]. `StATIK` is applicable whenever this condition is met. Here we outline several potential directions for leveraging `StATIK` for enterprise-relevant KG use cases.

3.1. Beyond KG Completion

While `StATIK` was developed for and evaluated on inductive knowledge graph completion, the architecture can be applied to other tasks that are relevant for various enterprise use cases. Daza et al. [28] showed that a text-based model trained on inductive knowledge graph completion could transfer to other tasks, such as entity classification and information retrieval.

There are two approaches that can be used to adapt `StATIK` to other tasks (1) pretrain then fine-tune: `StATIK` can be pretrained on inductive knowledge graph completion to learn rich entity representations and then fine-tuned on other downstream tasks. (2) Task-specific training: `StATIK` can be trained from scratch on a task such as entity classification.

Both approaches can be tried for any task. However, there are some cases in which one may be better. In some use cases where the primary task is tangential to the knowledge graph, such as knowledge-enhanced recommendation systems, it is likely more important to pretrain the KG model such that it learns rich KG representations [52, 53]. In other uses where the output of `StATIK` is more directly optimized by the primary task (e.g. entity classification), then either approach can be used.

3.2. Tasks Conditioned on Text

One of the advantages of incorporating a fine-tuned language model is that the model can learn to condition on text, rather than just represent the text. `StATIK` was trained on text for single hop queries to the KG–e.g. (Inception, director, ?). However, it does not need to be limited to that. `StATIK` can be conditioned on any text that is associated with the entity being encoded or task being performed. Here are a few examples of how this can be applied. Note that determining the true effectiveness of each of the following approaches is an area of future work.

- Adding in user features as text for a personalized knowledge graph

- Performing more complex KG question answering where the question can be encoded in text.
- Conditioning a knowledge graph model for recommendation systems on the genre a user wants
- Adding doctors notes to a patient diagnosis when encoding that ailment to improve patient healthcare recommendations
- Adding a summary of reviews for products being recommended

This ability to condition on text unlocks intriguing possibilities for various industry applications and could lead to better unification between language and KG domains.

4. Conclusion

StATIK is a powerful inductive model for representing knowledge graph entities. In return for the added complexity of using both message passing neural networks and language models, StATIK is able to utilize the rich information present in both text and graph features. This makes it appealing for many enterprise use cases where such information is available. In addition, its structure unlocks possibilities for applying StATIK in as-of-yet unexplored ways by utilizing the power of finetuned language models.

References

[1] L. Bauer, Identify, align, and integrate: Matching knowledge graphs to commonsense reasoning tasks, in: EACL, 2021.

[2] J. Yan, M. Raman, T. Zhang, R. A. Rossi, H. Zhao, S. Kim, N. Lipka, X. Ren, Learning contextualized knowledge structures for commonsense reasoning, ArXiv abs/2010.12873 (2021).

[3] H. Zhang, Z. Liu, C. Xiong, Z. Liu, Grounded conversation generation as guided traverses in commonsense knowledge graphs, in: ACL, 2020.

[4] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, J. Leskovec, Qa-gnn: Reasoning with language models and knowledge graphs for question answering, ArXiv abs/2104.06378 (2021).

[5] Y. Feng, X. Chen, B. Y. Lin, P. Wang, J. Yan, X. Ren, Scalable multi-hop relational reasoning for knowledge-aware question answering, in: EMNLP, 2020.

[6] B. Y. Lin, X. Chen, J. Chen, X. Ren, Kagnet: Knowledge-aware graph networks for commonsense reasoning, ArXiv abs/1909.02151 (2019).

[7] P. Christmann, R. S. Roy, A. Abujabal, J. Singh, G. Weikum, Look before you hop: Conversational question answering over knowledge graphs using judicious context expansion, Proceedings of the 28th ACM International Conference on Information and Knowledge Management (2019).

[8] A. Saxena, S. Chakrabarti, P. P. Talukdar, Question answering over temporal knowledge graphs, in: ACL, 2021.

[9] B. Hixon, P. Clark, H. Hajishirzi, Learning knowledge graphs for question answering through conversational dialog, in: NAACL, 2015.

[10] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, ArXiv abs/2003.00911 (2020).

[11] B. Wang, W. Cai, Knowledge-enhanced graph neural networks for sequential recommendation, Inf. 11 (2020) 388.

[12] J. Huang, W. X. Zhao, H.-J. Dou, J. rong Wen, E. Y. Chang, Improving sequential recommendation with knowledge-enhanced memory networks, The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (2018).

[13] H. Wang, F. Zhang, X. Xie, M. Guo, Dkn: Deep knowledge-aware network for news recommendation, Proceedings of the 2018 World Wide Web Conference (2018).

[14] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A. N. Ngomo, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, Communications of the ACM 64 (2021) 96 – 104.

[15] R. Das, T. Munkhdalai, X. Yuan, A. Trischler, A. McCallum, Building dynamic knowledge graphs from text using machine reading comprehension, CoRR abs/1810.05682 (2018). URL: http://arxiv.org/abs/1810.05682. arXiv:1810.05682.

[16] S. Liao, S. Liang, Z. Meng, Q. Zhang, Learning dynamic embeddings for temporal knowledge graphs, WSDM '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 535–543. URL: https://doi.org/10.1145/3437963.3441741. doi:10.1145/3437963.3441741.

[17] E. Markowitz, K. Balasubramanian, M. Mirtaheri, M. Annavaram, A. G. Galstyan, G. V. Steeg, Statik: Structure and text for inductive knowledge graph completion, in: NAACL-HLT, 2022. URL: https://api.semanticscholar.org/CorpusID:250520198.

[18] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, G. E. Dahl, Neural message passing for quantum chemistry, CoRR abs/1704.01212 (2017). URL: http://arxiv.org/abs/1704.01212. arXiv:1704.01212.

[19] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, 2019.

[20] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for model-

ing multi-relational data, in: NIPS, 2013.
[21] H. Shah, J. Villmow, A. Ulges, U. Schwanecke, F. Shafait, An open-world extension to knowledge graph completion models, ArXiv abs/1906.08382 (2019).
[22] R. Xie, Z. Liu, J. Jia, H. Luan, M. Sun, Representation learning of knowledge graphs with entity descriptions, in: AAAI, 2016.
[23] C. Malaviya, C. Bhagavatula, A. Bosselut, Y. Choi, Commonsense knowledge base completion with structural and semantic context, Proceedings of the 34th AAAI Conference on Artificial Intelligence (2020).
[24] D. Dai, H. Zheng, F. Luo, P. Yang, B. Chang, Z. Sui, Inductively representing out-of-knowledge-graph entities by optimal estimation under translational assumptions, ArXiv abs/2009.12765 (2021).
[25] P. Wang, J. Han, C. Li, R. Pan, Logic attention based neighborhood aggregation for inductive knowledge graph embedding, in: AAAI, 2019.
[26] K. K. Teru, E. Denis, W. L. Hamilton, Inductive relation prediction by subgraph reasoning, in: ICML, 2020.
[27] L. Yao, C. Mao, Y. Luo, Kg-bert: Bert for knowledge graph completion, ArXiv abs/1909.03193 (2019).
[28] D. Daza, M. Cochez, P. T. Groth, Inductive entity representations from text via link prediction, Proceedings of the Web Conference 2021 (2021).
[29] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, Y. Chang, Structure-augmented text representation learning for efficient knowledge graph completion, in: Proceedings of the Web Conference 2021, 2021, pp. 1737–1748.
[30] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: ICML, 2016.
[31] B. Yang, W. tau Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, CoRR abs/1412.6575 (2015).
[32] Z. Sun, Z. Deng, J.-Y. Nie, J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space, ArXiv abs/1902.10197 (2019).
[33] S. M. Kazemi, D. Poole, Simple embedding for link prediction in knowledge graphs, in: NeurIPS, 2018.
[34] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, CoRR abs/1707.01476 (2017). URL: http://arxiv.org/abs/1707.01476. arXiv:1707.01476.
[35] I. Balazevic, C. Allen, T. M. Hospedales, Hypernetwork knowledge graph embeddings, CoRR abs/1808.07018 (2018). URL: http://arxiv.org/abs/1808.07018. arXiv:1808.07018.
[36] M. Schlichtkrull, T. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, ArXiv abs/1703.06103 (2018).
[37] T. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, ArXiv abs/1609.02907 (2017).
[38] R. Wang, B. Li, S. Hu, W. Du, M. Zhang, Knowledge graph embedding via graph attenuated attention networks, IEEE Access 8 (2020) 5212–5224.
[39] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio', Y. Bengio, Graph attention networks, ArXiv abs/1710.10903 (2018).
[40] H. Wang, H. Ren, J. Leskovec, Entity context and relational paths for knowledge graph completion, CoRR abs/2002.06757 (2020). URL: https://arxiv.org/abs/2002.06757. arXiv:2002.06757.
[41] R. Bhowmik, G. de Melo, Explainable link prediction for emerging entities in knowledge graphs, in: SEMWEB, 2020.
[42] Z. Zhu, Z. Zhang, L.-P. Xhonneux, J. Tang, Neural bellman-ford networks: A general graph neural network framework for link prediction, in: Neural Information Processing Systems, 2021. URL: https://api.semanticscholar.org/CorpusID:235422273.
[43] B. Shi, T. Weninger, Open-world knowledge graph completion, in: AAAI, 2018.
[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
[45] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014.
[46] E. Markowitz, K. Balasubramanian, M. Mirtaheri, S. Abu-El-Haija, B. Perozzi, G. V. Steeg, A. Galstyan, Graph traversal with tensor functionals: A meta-algorithm for scalable learning, ArXiv abs/2102.04350 (2021).
[47] W. Choi, H. Lee, Inference of biomedical relations among chemicals, genes, diseases, and symptoms using knowledge representation learning, IEEE Access 7 (2019) 179373–179384. URL: https://api.semanticscholar.org/CorpusID:209459645.
[48] F. Farazi, M. Salamanca, S. Mosbach, J. Akroyd, A. Eibeck, L. K. Aditya, A. Chadzynski, K. Pan, X. Zhou, S. Zhang, M. Q. Lim, M. Kraft, Knowledge graph approach to combustion chemistry and interoperability, ACS Omega 5 (2020) 18342 – 18348. URL: https://api.semanticscholar.org/CorpusID:220883899.
[49] Y. Zhu, W. Zhou, Y. Xu, J. Liu, Y. Tan, Intelligent learning for knowledge graph towards geological data, Sci. Program. 2017 (2017) 5072427:1–5072427:13. URL: https://api.semanticscholar.org/CorpusID:29772448.
[50] Y. Liu, Q. Zeng, J. B. O. Meré, H. Yang, An-

ticipating stock market of the renowned companies: A knowledge graph approach, Complex. 2019 (2019) 9202457:1–9202457:15. URL: https://api.semanticscholar.org/CorpusID:201263701.

[51] B. Abu-Salih, Domain-specific knowledge graphs: A survey, ArXiv abs/2011.00235 (2020). URL: https://api.semanticscholar.org/CorpusID:226227432.

[52] E. Markowitz, Z. Jiang, F. Yang, X. Fan, T. Chen, G. V. Steeg, A. G. Galstyan, Multi-task knowledge enhancement for zero-shot and multi-domain recommendation in an ai assistant application, ArXiv abs/2306.06302 (2023). URL: https://api.semanticscholar.org/CorpusID:259137694.

[53] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, M. Guo, Multi-task feature learning for knowledge graph enhanced recommendation, The World Wide Web Conference (2019). URL: https://api.semanticscholar.org/CorpusID:59291937.