Fast-and-Frugal Text-Graph Transformers are Effective Link Predictors

Andrei C. Coman
Idiap Research Institute, EPFL
andrei.coman@idiap.ch

Christos Theodoropoulos
KU Leuven
christos.theodoropoulos@kuleuven.be

Marie-Francine Moens
KU Leuven
sien.moens@kuleuven.be

James Henderson
Idiap Research Institute
james.henderson@idiap.ch

Abstract

We propose Fast-and-Frugal Text-Graph (FnF-TG) Transformers, a Transformer-based framework that unifies textual and structural information for inductive link prediction in text-attributed knowledge graphs. We demonstrate that, by effectively encoding ego-graphs (1-hop neighbourhoods), we can reduce the reliance on resource-intensive textual encoders. This makes the model both fast at training and inference time, as well as frugal in terms of cost. We perform a comprehensive evaluation on three popular datasets and show that FnF-TG can achieve superior performance compared to previous state-of-the-art methods. We also extend inductive learning to a fully inductive setting, where relations don't rely on transductive (fixed) representations, as in previous work, but are a function of their textual description. Additionally, we introduce new variants of existing datasets, specifically designed to test the performance of models on unseen relations at inference time, thus offering a new test-bench for fully inductive link prediction.

1 Introduction

Knowledge graphs (KGs) represent complex information as a structured collection of entities and their relations. They are a fundamental component of various applications, including information extraction (Mintz et al., 2009; Bosselut et al., 2019; Theodoropoulos et al., 2021) and retrieval (Dalton et al., 2014; Gupta et al., 2019), question answering (Saxena et al., 2022; Yu et al., 2022; Coman et al., 2023), reasoning (Zhang et al., 2020; Jiang et al., 2022; Niu et al., 2022), fact-aware language modelling (Logan et al., 2019; Yang et al., 2023), and many others (Fensel et al., 2020; Schneider et al., 2022). Text-attributed KGs extend KGs by associating each entity and relation with a corresponding textual description, which provide a richer representation of the knowledge encoded in the graph. In particular, the text associated with an entity may provide a description of its relationships to other entities. This combination of explicit structural and implicit textual information makes modelling text-attributed KGs particularly challenging.

Initial attempts to model KGs focused on their graph nature, typically addressing a *transductive* setting (Bordes et al., 2013; Nickel et al., 2015; Wang et al., 2017). These models could only make predictions for entities observed during training and only considered the structural information of the KG, ignoring any textual information.

To overcome this limitation, later work focused on using the textual descriptions in KGs to address an *inductive* setting (Xie et al., 2016; Shi and Weninger, 2018; Wang et al., 2021b), meaning that predictions can be made even for entities not observed during training, using entity representations computed based on their textual descriptions.

Combining information from textual descriptions and graph structures has proven crucial (Schlichtkrull et al., 2017). An entity's ego-graph, which represents it's 1-hop neighbourhood, provides valuable context that can help disambiguate its role and distinguish it from similar entities. While there has been progress in leveraging ego-graphs, we believe that there is significant room for more effective approaches.

Modelling text-attributed KGs in an inductive setting poses several challenges, particularly when it comes to effectively integrating textual and structural information in embeddings. Transformers (Vaswani et al., 2017) have shown remarkable success at modelling unstructured (text) data (Devlin et al., 2019; Raffel et al., 2019; Brown et al., 2020; Touvron et al., 2023). While their ability to model structured (graph) data is less evident, the inherent graph processing abilities of a Transformer's self-attention mechanism make it a natural fit for modelling graph structures (Henderson et al., 2023). We leverage recent advances in using Transformers for graph encoding (Mohammadshahi and Hender-

Findings of the Association for Computational Linguistics: ACL 2025, pages 11828–11841

son, 2020, 2021; Miculicich and Henderson, 2022; Coman et al., 2024), and propose Fast-and-Frugal Text-Graph (FnF-TG) Transformers, which unify textual and structural information in a framework based solely on Transformers.

Another challenge is that text encoders are resource-intensive, especially when the textual descriptions of both entities and their ego-graph neighbours need to be encoded, leading to considerably increased training and inference time (Markowitz et al., 2022). This cost can be reduced by using smaller text encoders, but they can be considerably less effective. We demonstrate that we can reduce the dependence on large text encoders with a more effectively encoding of ego-graphs, using our FnF-TG Transformers and their more appropriate inductive biases. This makes the overall framework both fast in terms of time, as well as frugal in terms of cost.

A third challenge is that previous models fail to leverage the textual descriptions of relation labels. They still assume a fixed (transductive) inventory of relations, meaning that they cannot handle relations which they did not see during training and thus are not fully inductive. We propose an extension of this method to address the challenge of being fully inductive, by computing a relation embedding from the text describing that relation. This embedding serves as both the relation representation for link prediction, analogous to the transductive case, and also as the relation label which is input to the self-attention mechanism of the FnF-TG's graph encoder component.

We showcase the effectiveness of our proposed model on three popular datasets for inductive link prediction in text-attributed KGs from the experimental setting of Daza et al. (2021) and Wang et al. (2021b), namely WN18RR$_{\text{IND}}$, FB15k-237$_{\text{IND}}$, and Wikidata-5M$_{\text{IND}}$. We show that it improves over the state-of-the-art in all cases.

Additionally, we introduce new variants of existing datasets which are specifically designed to evaluate the performance of models on relations which are unseen until test time, thus offering a new test-bench for fully inductive link prediction.

Contributions:

1. We propose a KG embedding model which leverages the intrinsic graph processing capabilities of Transformers to effectively capture the information in both the KG's textual descriptions and the KG's graph structure.
2. We demonstrate that Fast-and-Frugal Text-Graph (FnF-TG[1]) Transformers achieve superior performance compared to previous state-of-the-art results on three popular datasets, even with small and efficient text encoders.
3. We extend inductive KG learning to a fully inductive setting, where both entity and relation representations are computed as functions of their textual descriptions.
4. We introduce a new test-bench for fully inductive link prediction by modifying existing datasets to specifically test models' performance on unseen relations.

2 Related Work

Transductive Link Prediction In this setting, link prediction aims to identify missing links within a fixed and fully observable graph where all entities and their other connections are known during training. Typically, it involves learning embeddings within a geometric space, as demonstrated by models like RESCAL (Nickel et al., 2011), NTN (Socher et al., 2013), TransE (Bordes et al., 2013), DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), TorusE (Ebisu and Ichise, 2018), RotatE (Sun et al., 2019), and SimplE (Kazemi and Poole, 2018). Additionally, there are approaches that incorporate convolutional layers such as R-GCN (Schlichtkrull et al., 2017), ConvE (Dettmers et al., 2018), HypER (Balazevic et al., 2018), and ConvR (Jiang et al., 2019). Moreover, recent advances have seen the integration of Transformers in models such as CoKE (Wang et al., 2019b) and HittER (Chen et al., 2021).

Inductive Link Prediction In this setting, link prediction involves predicting missing links in a dynamic graph where only partial information is available during training. Much work has explored leveraging limited relational knowledge between novel entities and those already present in the training graph (Bhowmik and de Melo, 2020; Wang et al., 2020). Examples include LAN (Wang et al., 2019a), IndTransE (Dai et al., 2021), OpenWorld (Shah et al., 2019), GraIL (Teru et al., 2020), NBFNet (Zhu et al., 2021), NodePiece (Galkin et al., 2021), and BERTRL (Zha et al., 2021). Moreover, approaches such as DKRL (Xie et al., 2016), Commonsense (Malaviya et al., 2020), KG-BERT (Yao et al., 2019), KEPLER (Wang et al., 2021b),

[1] https://github.com/idiap/fnf-tg

BLP (Daza et al., 2021), StAR (Wang et al., 2021a), SimKGC (Wang et al., 2022), StATIK (Markowitz et al., 2022), iHT (Chen et al., 2023), and KnowC (Yang et al., 2024) use language models to encode entities based on their textual descriptions. Among these, StATIK (Markowitz et al., 2022) stands out as it combines both a language model and a graph encoder, specifically employing a Message Passing Neural Network (MPNN) (Gilmer et al., 2017) to create entity embeddings. This makes StATIK particularly relevant to our approach and we will use it as the state-of-the-art method of reference and compare our proposed method against it to demonstrate its effectiveness.

Transformers and Graphs Graph Transformers (GTs) represent a significant evolution in graph input methods within the Transformer architecture (Henderson et al., 2023). Early work such as G2GT (Mohammadshahi and Henderson, 2020, 2021; Miculicich and Henderson, 2022) laid the foundation by incorporating explicit graphs into Transformer's latent attention graph. Later work introduced RoFormer (Su et al., 2021), which uses a rotation matrix to encode absolute positions, and Graphormer (Ying et al., 2021), which uses node centrality encoding and soft attention biases. Other models, like SSAN (Xu et al., 2021), JointGT (Ke et al., 2021), TableFormer (Yang et al., 2022), and GADePo (Coman et al., 2024), have applied GTs to various tasks such as document-level relation extraction, knowledge-to-text generation, table-based question answering, and graph-aware declarative pooling.

We continue to advance graph input methods and show that GTs, when combined with an effective inductive bias in the input and the latent attention graph, achieve superior performance compared to the previous state-of-the-art.

3 Background

3.1 Inductive Representation Learning

A text-attributed knowledge graph can be defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{D})$ where $\mathcal{E}$ represents the set of entities, $\mathcal{R}$ denotes the set of relation labels, $\mathcal{T}$ consists of the set of relation triples $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, and $\mathcal{D}$ contains the textual descriptions associated with entities and relation labels. In each triple, h and t represent the head and tail entities, respectively, which are connected by a directional relation r. Inductive link prediction involves completing missing triples in the graph by leveraging the textual descriptions associated with the entities and relation labels. If the textual description of an entity mentions the target relation, then this resembles an open relation extraction task (Banko et al., 2007), and if not then it resembles a knowledge graph completion task (Lin et al., 2015), with a continuum of difficulties in between. Our goal is to learn an embedder that maps these descriptions and the partial graph to a representation space where the missing triples can be inferred. Specifically, given a training graph $\mathcal{G}_{train} = (\mathcal{E}_{train}, \mathcal{R}, \mathcal{T}_{train}, \mathcal{D}_{train})$, where $\mathcal{E}_{train} \subset \mathcal{E}$, $\mathcal{D}_{train} \subset \mathcal{D}$ and $\mathcal{T}_{train} \subset \mathcal{T}$ only includes triples involving entities in $\mathcal{E}_{train}$, the goal is to infer the missing triples in $\mathcal{T} \setminus \mathcal{T}_{train}$. During evaluation, for a given query triple $\mathcal{T}_i = (h, r, t)$, the model is tasked with performing head or tail prediction on the graph $\mathcal{G} \setminus \mathcal{T}_i$. This involves two types of queries: tail prediction, where the query is of the form $(h, r, \hat{e})$ and head prediction, where the query is of the form $(\hat{e}, r, t)$. In both cases, the model must rank all possible candidate entities $\hat{e} \in \hat{\mathcal{E}}$ to identify the correct entity $\hat{e}_t$ or $\hat{e}_h$ and place it at the top of the ranked list.

3.2 Structural Objective and Loss Function

We adopt the margin-based ranking loss from Bordes et al. (2013) as our optimisation criterion. We construct two sets of triples: a set of true triples T and a set of negative triples T', where a negative triple consists of a corrupted version of a true triple with either the head or tail entity replaced by a random entity (target excluded) from the training minibatch. We define the structural objective function f using the TransE (Bordes et al., 2013) model, which represents each triple (h, r, t) as:

$$f_{\text{TransE}}(h, r, t) = -||\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}||_1$$

where $\boldsymbol{h}$, $\boldsymbol{r}$, and $\boldsymbol{t}$ are the vector representations of the head entity, relation, and tail entity, respectively. The loss function is then defined as:

$$Loss = \sum_{(t,t') \in (T \times T')} max(0, 1 - f(t) + f(t'))$$

where $f(t)$ and $f(t')$ are the scores assigned to the true triple t and the negative triple t', respectively.

3.3 Evaluation and Metrics

Evaluation Scenarios We assess our models in two inductive scenarios, following Bordes et al. (2013). In the first setting, called *dynamic* evaluation, new entities may appear in the head or tail positions, and the candidates set is defined as $\hat{\mathcal{E}} = \mathcal{E}_{train} \cup \mathcal{E}_{eval}$. In the second setting, called

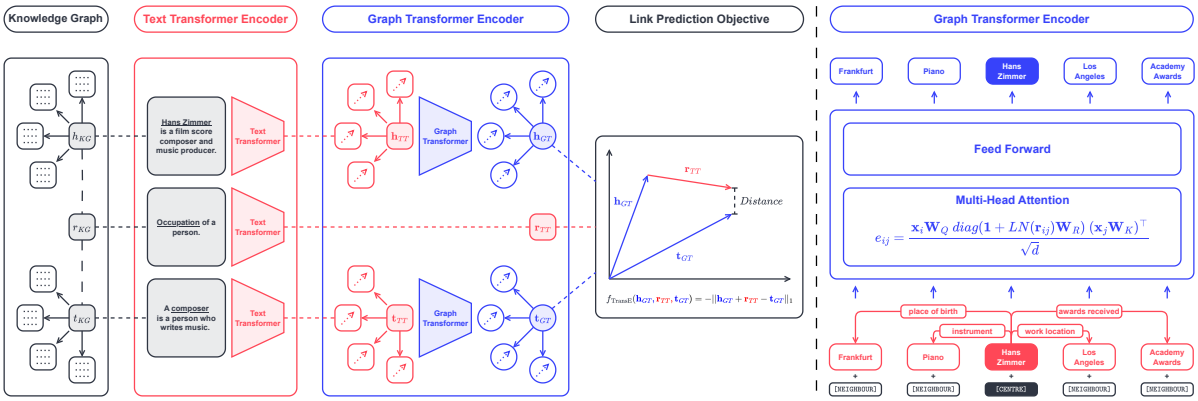


Figure 1: Architecture of the proposed Fast-and-Frugal Text-Graph (FnF-TG) Transformer model.

transfer evaluation, both head and tail entities are new and unseen during training, and the candidates set is defined as $\hat{\mathcal{E}} = \mathcal{E}_{eval}$, where $\mathcal{E}_{eval}$ is disjoint from the training set of entities $\mathcal{E}_{train}$.

Metrics For each evaluation triple, we create two types of queries: $(h, r, \hat{e})$ for predicting tails and $(\hat{e}, r, t)$ for predicting heads, where $\hat{e} \in \mathcal{E}$ represents all possible candidate entities, as described in Subsection 3.1. We rank candidate triples by their scores and evaluate the ranking of the correct triple. We report Mean Reciprocal Rank (MRR) and Hits@k (H@k) with $k \in \{1, 3, 10\}$ averaged across head and tail prediction tasks. We adopt the *filtered* setting as in Bordes et al. (2013), removing valid triples from the set of negative candidate triples when ranking candidate targets.

3.4 Proposed Architecture

Figure 1 shows the overall architecture of our proposed model. There are three main components: *Knowledge Graph* (KG), *Text Transformer Encoder* (TT) and *Graph Transformer Encoder* (GT).

Knowledge Graph The text-attributed KG component contains a set of triples of type (h_{KG}, r_{KG}, t_{KG}), along with their corresponding textual descriptions. For each head h_{KG} and tail t_{KG}, we also extract their ego-graphs (1-hop neighbourhood), denoted as $E(h_{KG})$ and $E(t_{KG})$, respectively. Then we encode each of these nodes with the text encoder, discussed below. Specifically, for each centre entity, we encode its textual description, as well as encoding the textual descriptions of its neighbouring entities and the relations that connect the centre entity to its neighbours.

Text Transformer Encoder The textual descriptions from the KG module are passed to the *Text Transformer Encoder* (TT), which produces vector representations $\boldsymbol{x}_{TT}$ for each entity and relation textual description x_{KG} in the ego-graph. More formally, we apply the following function:

$$\boldsymbol{x}_{TT} = \sigma(\text{BERT}_{\text{SIZE}}(x_{KG})_{\texttt{[CLS]}}\boldsymbol{W}_0)\boldsymbol{W}_1$$

where $\boldsymbol{W}_0, \boldsymbol{W}_1 \in \mathbb{R}^{d \times d}$ are two linear projection matrices and σ is the SiLU (Elfwing et al., 2017) activation function. We employ BERT (Devlin et al., 2019) as our encoder and use the `[CLS]` vector representation output by the encoder as the embedding. $\text{BERT}_{\text{SIZE}}$ indicates that we employ different sizes of this model released by Turc et al. (2019).

When encoding candidate entities $\hat{e} \in \hat{\mathcal{E}}$, we simply pass the text associated with the entity through the TT component. The same is true for any neighbouring entities required by the graph encoder (discussed below). In contrast, when encoding queries $(h, r, \cdot)$ and $(\cdot, r, t)$ we condition h and t on the relation type r. Similarly to StAR (Wang et al., 2021a) and StATIK (Markowitz et al., 2022), for tail prediction queries $(h, r, \cdot)$ we concatenate the text associated with h_{KG} and r_{KG}, resulting in $[h||r]_{KG}$. For head prediction queries $(\cdot, r, t)$, we create an inverse version of the relation text by prepending its textual description with the text `"inverse of"`, denoted as r^{-1}_{KG}. We then concatenate the text associated with t_{KG} and r^{-1}_{KG}, resulting in $[t||r^{-1}]_{KG}$.

Graph Transformer Encoder The outputs of the TT component, together with the ego-graphs $E(\boldsymbol{h}_{TT})$ and $E(\boldsymbol{t}_{TT})$ are then input to the *Graph Transformer Encoder* (GT). The input embedding for each node of the graph is the vector output by the TT encoder for that entity, as described above. In addition, we add learnable segment embeddings to each node input, indicated as $s_{\texttt{[CENTRE]}}$ and $s_{\texttt{[NEIGHBOUR]}}$, to disambiguate between the centre and neighbour nodes in the ego-graph. These

embeddings indicate to the model which input nodes will be used subsequently as the embedding representation of the ego-graph.

To encode the graph relations, we follow Mohammadshahi and Henderson (2020, 2021); Miculicich and Henderson (2022); Coman et al. (2024) in leveraging the intrinsic graph processing capabilities of the Transformer model by incorporating graph relations as relation embeddings input to the self-attention function. But unlike in that previous work, our relation embeddings are computed from the text associated with the relation, rather than coming from a fixed set of relations. For every pair of input nodes ij, the pre-softmax attention score $e_{ij} \in \mathbb{R}$ is computed from both the respective node embeddings $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$, and the embedding of the relation r_{ij} between the i-th and j-th nodes, as:

$$e_{ij} = \frac{\boldsymbol{x}_i \boldsymbol{W}_Q \ \mathrm{diag}(\mathbf{1} + \mathrm{LN}(\boldsymbol{r}_{ij})\boldsymbol{W}_R) \ (\boldsymbol{x}_j \boldsymbol{W}_K)^\top}{\sqrt{d}}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K \in \mathbb{R}^{d \times d}$ represent the query and key matrices, respectively, $\boldsymbol{r}_{ij}$ represents the relation embedding output by the TT module when it encodes the text associated with the relation between the i-th and j-th nodes, and $\boldsymbol{W}_R \in \mathbb{R}^{d \times d}$ is the relation matrix. Thus, $\mathrm{LN}(\boldsymbol{r}_{ij})\boldsymbol{W}_R$ is the embedding of the relation between i and j, where LN stands for the $LayerNorm$ operation. Finally, $\mathrm{diag}(\mathbf{1} + \ldots)$ maps this vector into a diagonal matrix plus the identity matrix.

When encoding candidate entities and queries in the GT, it is crucial to ensure that no information regarding the target triple (h, r, t) leaks into the $E(\hat{e})$, $E(\boldsymbol{h}_{TT})$, or $E(\boldsymbol{t}_{TT})$ ego-graphs. This precaution prevents the model from learning trivial solutions or biases from leaked information.

3.5 Datasets and Setting

Daza et al. (2021) introduced the WN18RR$_{\text{IND}}$ and FB15k-237$_{\text{IND}}$ inductive variants of the well-known WN18RR$_{\text{TRA}}$ (Dettmers et al., 2018) and FB15k-237$_{\text{TRA}}$ (Toutanova and Chen, 2015) transductive KGs. The inductive setting simulates a *dynamic* scenario where new entities and triples are dynamically added to the graph. The training graph is constructed as $\mathcal{G}_{train} = \{(h, r, t) \in \mathcal{T}_{train} : h, t \in \mathcal{E}_{train}\}$. The validation and test graphs are constructed by incrementally adding entities and triples, such that $\mathcal{G}_{val} = \{(h, r, t) \in \mathcal{T}_{val} : h, t \in \mathcal{E}_{train} \cup \mathcal{E}_{val}\}$ and $\mathcal{G}_{test} = \{(h, r, t) \in \mathcal{T}_{test} : h, t \in \mathcal{E}_{train} \cup \mathcal{E}_{val} \cup \mathcal{E}_{test}\}$.

In contrast, the Wikidata5M$_{\text{IND}}$ KGs curated by Wang et al. (2021b) provide a *transfer* learning scenario in which the evaluation graphs are constructed such that the validation and test entity and triple sets, $\mathcal{E}_{val}$ and $\mathcal{E}_{test}$, and $\mathcal{T}_{val}$ and $\mathcal{T}_{test}$ are disjoint from the training entity and triple set $\mathcal{E}_{train}$ and $\mathcal{T}_{train}$. The validation and test graphs are constructed as $\mathcal{G}_{val} = \{(h, r, t) \in \mathcal{T}_{val} : h, t \in \mathcal{E}_{val}\}$ and $\mathcal{G}_{test} = \{(h, r, t) \in \mathcal{T}_{test} : h, t \in \mathcal{E}_{test}\}$. Because the graphs $\mathcal{G}_{val}$ and $\mathcal{G}_{test}$ do not include the entities from $\mathcal{G}_{train}$, they are much smaller graphs (see Appendix Table 6), which poses challenges for generalisation with graph-aware models, as will be discussed further below. We evaluate our model's ability to generalise to entirely new entities and triples in this setting.

We conduct our experiments in the above-mentioned settings of Daza et al. (2021) and Wang et al. (2021b), where textual information extraction is an integral part. Our method is directly comparable to DKRL (Xie et al., 2016), BLP (Daza et al., 2021), KEPLER (Wang et al., 2021b), StAR (Wang et al., 2021a), and the state-of-the-art method, StATIK (Markowitz et al., 2022) which employ the same textual encoder, structural objective, and loss function. Similar to StATIK, our work aims to jointly model the text and the structure of knowledge graphs, including extracting information about KG links from the text. This sets us apart from the setting of Teru et al. (2020), which uses different KGs splits and is employed in GraIL (Teru et al., 2020), NBFNet (Zhu et al., 2021), and NodePiece (Galkin et al., 2021), that solely focus on using the structure of the graph without incorporating any textual information extraction component.

3.6 Controlled Experimental Setup

When comparing the performance of different models on link prediction tasks, it is crucial to establish a fair and consistent baseline. Our experiments in Table 2 highlight the importance of carefully setting this baseline, as various factors can greatly influence the results.

Specifically, we demonstrate that the computational budget, which determines training hyperparameters, can have a substantial impact on model performance. Starting with the baseline model BLP$_{\text{BERT}_{\text{BASE}}}$ (Daza et al., 2021), we introduce improvements such as using inductive relations, increasing the number of negative triples to match the batch size (negatives batch tying), increasing the embedding dimension from 128 to 768, doubling the batch size from 64 to 128, and modifying the

Model	WN18RR$_{IND}$ MRR	H@1	H@3	H@10	FB15k-237$_{IND}$ MRR	H@1	H@3	H@10
Text-Only Models								
DKRL$^{*}_{BERT_{BASE}}$	0.139	0.048	0.169	0.320	0.144	0.084	0.151	0.263
BOW$^{*}_{BERT_{BASE}}$	0.180	0.045	0.244	0.450	0.173	0.103	0.184	0.316
BLP$^{*}_{BERT_{BASE}}$	0.285	0.135	0.361	0.580	0.195	0.113	0.213	0.363
FnF-T$_{BERT_{BASE}}$	**0.373**	**0.238**	**0.442**	**0.647**	**0.266**	**0.174**	**0.297**	**0.453**
FnF-T$_{BERT_{MEDIUM}}$	0.342	0.213	0.405	0.603	0.253	0.164	0.281	0.431
FnF-T$_{BERT_{SMALL}}$	0.320	0.197	0.379	0.572	0.239	0.152	0.265	0.411
FnF-T$_{BERT_{MINI}}$	0.268	0.156	0.318	0.498	0.204	0.128	0.223	0.354
FnF-T$_{BERT_{TINY}}$	0.193	0.098	0.230	0.385	0.164	0.100	0.176	0.289
Structure-Informed Models								
StAR$^{\star}_{BERT_{BASE}}$	0.321	0.192	0.381	0.576	0.163	0.092	0.176	0.309
StATIK$^{\star}_{BERT_{BASE}}$	0.516	0.425	0.558	0.690	0.224	0.143	0.248	0.381
FnF-TG$_{BERT_{BASE}}$	0.732	0.652	0.785	**0.875**	0.316	0.214	0.350	**0.524**
FnF-TG$_{BERT_{MEDIUM}}$	**0.737**	**0.661**	**0.789**	0.873	0.314	0.214	0.353	0.515
FnF-TG$_{BERT_{SMALL}}$	0.727	0.648	0.781	0.867	**0.316**	**0.216**	**0.354**	0.518
FnF-TG$_{BERT_{MINI}}$	0.713	0.632	0.768	0.857	0.302	0.204	0.337	0.502
FnF-TG$_{BERT_{TINY}}$	0.638	0.543	0.700	0.808	0.288	0.195	0.318	0.475

Table 1: WN18RR$_{IND}$ and FB15k-237$_{IND}$ test set results. *Daza et al. (2021); *Markowitz et al. (2022).

Model	MRR WN18RR$_{IND}$	FB15k-237$_{IND}$
BLP$_{BERT_{BASE}}$	0.285	0.195
BLP$^{\bullet}_{BERT_{BASE}}$	0.280	0.205
+ inductive relations	0.281	0.219
+ negatives batch tying	0.300	0.221
+ bigger embedding size	0.339	0.254
+ bigger batch size	0.366	0.260
+ better sampling method	0.373	0.266
FnF-T$_{BERT_{BASE}}$ (ours)	**0.373**	**0.266**

Table 2: WN18RR$_{IND}$ and FB15k-237$_{IND}$ test set results with cumulative additions over the baseline model BLP$_{BERT_{BASE}}$ (Daza et al., 2021) that lead to our improved baseline model FnF-T$_{BERT_{BASE}}$. BLP$^{\bullet}_{BERT_{BASE}}$ indicates our reimplementation of BLP$_{BERT_{BASE}}$.

Model	MRR	H@1	H@3	H@10
Text-Only Models				
KEPLER$^{\diamond}_{BERT_{BASE}}$	0.402	0.222	0.514	0.730
BLP$^{*}_{BERT_{BASE}}$	0.478	0.241	0.660	0.871
FnF-T$_{BERT_{BASE}}$	**0.597**	**0.427**	**0.722**	**0.896**
FnF-T$_{BERT_{MEDIUM}}$	0.588	0.418	0.712	0.890
FnF-T$_{BERT_{SMALL}}$	0.588	0.417	0.714	0.889
FnF-T$_{BERT_{MINI}}$	0.562	0.391	0.683	0.870
FnF-T$_{BERT_{TINY}}$	0.526	0.348	0.649	0.849
Structure-Informed Models				
StATIK$^{\star}_{BERT_{BASE}}$	0.770	**0.765**	0.771	0.779
FnF-TG$_{BERT_{BASE}}$	**0.799**	0.741	**0.833**	**0.911**
FnF-TG$_{BERT_{MEDIUM}}$	0.785	0.727	0.817	0.900
FnF-TG$_{BERT_{SMALL}}$	0.781	0.721	0.816	0.898
FnF-TG$_{BERT_{MINI}}$	0.779	0.719	0.814	0.894
FnF-TG$_{BERT_{TINY}}$	0.761	0.697	0.799	0.883

Table 3: Wikidata5M$_{IND}$ test set results. $^{\diamond}$Wang et al. (2021b); *Daza et al. (2021); *Markowitz et al. (2022).

negative sampling strategy to two-sided reflexive, where both head and tail entities are considered as potential negatives. These cumulative improvements lead to the development of a new text-only model baseline, FnF-T$_{BERT_{BASE}}$, which shows substantial improvements on both the WN18RR$_{IND}$ and FB15k-237$_{IND}$ datasets.

To ensure a fair comparison, we fixed our computational budget to a constant in this paper, using a consumer-grade GPU (NVIDIA RTX3090 24GB). This allows for a consistent and reproducible experimental setup, enabling a more accurate assessment of performance. For more details, see Appendix A.

3.7 Inductive Link Prediction Results

As shown in the top half of Table 1, for both the WN18RR$_{IND}$ and the FB15k-237$_{IND}$ datasets, our inductive relation embeddings and the enhanced controlled experimental setup result in improved text-only models. These models rely heavily on having powerful text encoders, as shown by the degradation in performance when using smaller versions of BERT as the text encoder.

The addition of our graph encoder to the model (bottom half of Table 1) leads to a substantial increase in link prediction accuracy over the text-only model. We also see that our TG (text-graph) encoder results in substantially better accuracy than the previous state-of-the-art model, StATIK. Interestingly, this more effective use of graph context also has a big impact on the model's dependence on powerful text encoders. Reducing the size of the text encoder (BERT$_{BASE}$ > BERT$_{MEDIUM}$ > BERT$_{SMALL}$ > BERT$_{MINI}$ > BERT$_{TINY}$) does re-

sult in some degradation of accuracy, but the differences are much smaller than in the text-only case. Even with a $\text{BERT}_{\text{TINY}}$ text encoder, the graph-aware model performs better than the text-only model with a $\text{BERT}_{\text{BASE}}$ encoder. This shows that the inductive bias of explicit graph relations can be an effective alternative to extracting the same information from text with a powerful text encoder.

This pattern of results is repeated in the transfer case, shown in Table 3. Here, the training set is much larger, but the graph in the test set is relatively small with each entity having fewer neighbours (see Appendix Table 6). This reduces the advantage gained from adding an effective graph encoder and the margin of our models' improvement over the text-only models, and over the previous state-of-the-art model, StATIK.[2] But we still see the same pattern where the size of the text encoder has less effect on accuracy for the graph-aware model.

3.8 Ablation Study

Table 4 presents results from our ablation studies, showing the impact of removing various design features from our graph-aware model on its accuracy.

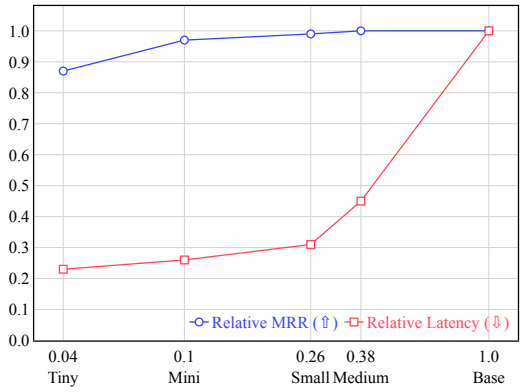
	MRR	
Model	**WN18RR$_{\text{IND}}$**	**FB15k-237$_{\text{IND}}$**
FnF-TG$_{\text{BERT}_{\text{MEDIUM | SMALL}}}$	0.737	0.316
$-\ r_{ij}$	0.733	0.306
$-\ s_{\texttt{[CENTRE]}}, s_{\texttt{[NEIGHBOUR]}}$	0.677	0.298
$-\ E(\boldsymbol{h}_{TT}), E(\boldsymbol{t}_{TT})$	0.480	0.251
$-\ [h\|\|r]_{KG}, [t\|\|r^{-1}]_{KG}$	0.342	0.239

Table 4: Ablation studies on the WN18RR$_{\text{IND}}$ and FB15k-237$_{\text{IND}}$ test sets using the top FnF-TG model. Each row indicates the performance after cumulatively removing a specific feature.

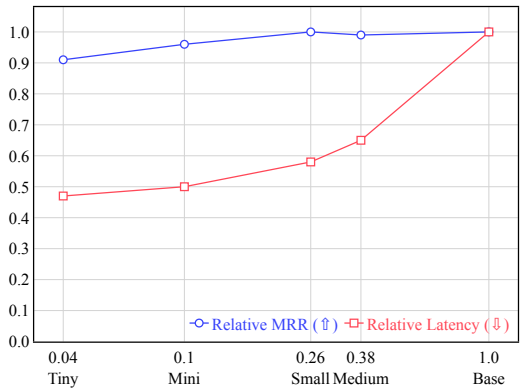
Removing the r_{ij} relation embeddings in the pre-softmax attention score leads to a decline in model performance, with a more substantial drop observed on the FB15K-237$_{\text{IND}}$ dataset compared to the WN18RR$_{\text{IND}}$ dataset. Note that with this modification the model still knows that there is some relation to the neighbours, but does not know its label. Removing the learnable segment embeddings $s_{\texttt{[CENTRE]}}$ and $s_{\texttt{[NEIGHBOUR]}}$ then removes this unlabelled graph structure, which considerably impacts the model's performance. Eliminating the ego-graph neighbours altogether results in an even more substantial performance drop. Despite this, the model remains competitive as a text-only model compared to the BLP baseline, owing to its ability to leverage relation conditioning features to represent candidate relations. Finally, removing the relation conditioning $[h||r]_{KG}$ and $[t||r^{-1}]_{KG}$, results in a further notable decrease in performance. Without relation conditioning, the model loses its ability to anticipate the query relation, severely impacting its accuracy.

3.9 Efficient Text Encoders

Being able to reduce the size of the text encoder with minimal degradation in accuracy is important because the text encoder is a substantial part of the training cost. In Figure 2 we plot the relative reduction in accuracy against the relative reduction in training time as we reduce the size of the text encoder, for the WN18RR$_{\text{IND}}$ and the FB15k-237$_{\text{IND}}$ datasets. We see that reducing the encoder size by a factor of four reduces the training time by a factor of three for WN18RR$_{\text{IND}}$ (and nearly two for FB15k-237$_{\text{IND}}$) with very little reduction in accuracy.



WN18RR$_{\text{IND}}$



FB15k-237$_{\text{IND}}$

Figure 2: Accuracy and training time plotted as a function of text encoder size, relative to the largest text encoder with the highest accuracy, shown as $(1.0, 1.0)$.

[2] StATIK has a surprisingly high H@1 score, almost identical to its H@3, H@10 and MRR scores. It is not clear why this is the case. Regardless, our model's MRR, H@3, and H@10 scores are better than StATIK. MRR is the primary evaluation measure since it summarises the entire ranking.

		WN18RR				FB15k-237				Wikidata5M			
Training	**Evaluation**	**MRR**	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**	**MRR**	**H@1**	**H@3**	**H@10**
FnF-$T_{BERT_{TINY}}$													
IND	IND	0.193	0.098	0.230	0.385	0.164	0.100	0.176	0.289	0.526	0.348	0.649	0.849
FIR	IND	0.169	0.080	0.198	0.346	0.143	0.087	0.153	0.253	0.478	0.306	0.591	0.793
IND	IND \ FIR	0.307	0.210	0.353	0.495	0.171	0.102	0.185	0.308	0.594	0.451	0.697	0.849
FIR	IND \ FIR	0.064	0.012	0.082	0.159	0.024	0.007	0.026	0.052	0.219	0.054	0.334	0.499
FnF-$TG_{BERT_{TINY}}$													
IND	IND	0.638	0.543	0.700	0.808	0.288	0.195	0.318	0.475	0.761	0.697	0.799	0.883
FIR	IND	0.573	0.480	0.629	0.738	0.249	0.165	0.274	0.418	0.711	0.644	0.749	0.837
IND	IND \ FIR	0.585	0.483	0.652	0.769	0.282	0.176	0.311	0.514	0.726	0.646	0.781	0.867
FIR	IND \ FIR	0.108	0.028	0.147	0.242	0.050	0.023	0.051	0.099	0.401	0.287	0.480	0.589
Random baseline													
–	–	0.0003	–	–	–	0.0007	–	–	–	0.0013	–	–	–

Table 5: Fully inductive link prediction results.

3.10 Fully Inductive Link Prediction Results

The experimental setting of Daza et al. (2021) and Wang et al. (2021b) do not support evaluation on unseen relations. One distinctive advantage of our model is that it is not restricted to a fixed set of relation labels learned during training. Although we do show that conditioning on relation texts improves accuracy even on seen relations (see Table 2), it is important to evaluate our model in a fully inductive setting, where relations are also unseen, in addition to entities.

To this end, we propose a new experimental setting for a fully inductive relations (FIR) evaluation, by converting the $\text{WN18RR}_{\text{IND}}$, $\text{FB15k-237}_{\text{IND}}$, and $\text{Wikidata-5M}_{\text{IND}}$ evaluations to their respective FIR versions. More specifically, we focus on the long tail of relations and remove the least frequent relation labels until 10% of edges have been removed from the training graph $\mathcal{G}_{train}$. We then train a new set of models on this new version so they have not seen the removed relation labels, and evaluate them on both the full set of test relations (IND) and specifically on the relations for the unseen labels (IND \ FIR). Given that all the previous models (DKRL, BLP, KEPLER, StAR, and StATIK) are inductive in entities but transductive in relations, none of them can make informed predictions in this setting, so we compare to a random baseline which computes the expected MRR for random rankings of candidate entities $\hat{\mathcal{E}}$ as follows:

$$\mathbb{E}[\text{MRR}_{\text{random}}] = \frac{1}{|\hat{\mathcal{E}}|} \sum_{i=1}^{|\hat{\mathcal{E}}|} \frac{1}{i}$$

The results in Table 5 show the performance of our model on this new setting. Our approach shows promising results as it outperforms the random baseline by a significant margin. However, the performance drops considerably when training on FIR and evaluating on IND \ FIR, indicating that the model struggles with unseen relations. Notably, the results on the Wikidata-5M dataset are considerably better than those obtained on the WN18RR and FB15k-237 datasets, probably due to having relations with more descriptive texts. Nevertheless, these results highlight the need for further research in developing models that can effectively generalize to unseen relations.

4 Conclusion

We presented a new Transformers-based approach to link prediction in text-attributed knowledge graphs that combines textual descriptions and graph structure in a fully inductive setting. Our Fast-and-Frugal Text-Graph (FnF-TG) Transformers outperform previous state-of-the-art models on three popular datasets, showcasing the importance of capturing rich structured information about entities and their relations. Our approach achieves superior performance while maintaining efficiency and scalability, making it a promising solution for large-scale knowledge graph applications. Moreover, our ablation studies provide insights into the key factors contributing to its effectiveness, demonstrating the value of each component in our model. Additionally, we proposed a new evaluation setting for fully inductive link prediction, where relations are also inductive, and demonstrated the potential of our approach in this setting.

Limitations

While our approach has achieved promising results, there are opportunities for further improvement.

One area for exploration is optimising the scalability of our *Graph Transformer Encoder* component (see Figure 1), which currently requires computing fully quadratic attention over the entire ego-graph of a given entity. In fact it could still require considerable resources if the number of nodes in the ego-graph is scaled to the order of thousands, hundreds of thousands, or even millions.

Our work demonstrates that effectively capturing even local neighbourhood information is both non-trivial and under-explored and that it can significantly enhance performance. Indeed, our simplification to a 1-hop neighbourhood (ego-graph) was a careful decision to balance effectiveness and complexity. This approach not only allows for a fair comparison with the current state-of-the-art method, StATIK (Markowitz et al., 2022), but also mitigates the exponential increase in computational complexity (see Appendix Subsection A.2) associated with larger neighbourhoods. While this predefined 1-hop neighbourhood provides a solid starting point, there is room to explore better alternatives. For instance, investigating multi-hop neighbourhoods or adaptive neighbourhood definitions could uncover more nuanced insights from the graph structure, potentially leading to even better results.

By building upon our framework, future work could refine these aspects, ultimately enhancing the effectiveness and versatility of our approach.

Ethics Statement

We do not anticipate any ethical concerns related to our work, as it primarily presents an alternative approach to a previously proposed method. Our main contribution lies in introducing a new approach for link prediction. In our experiments, we use the same datasets and pretrained models as previous research, all of which are publicly available. However, it is important to acknowledge that these datasets and models may still require further examination for potential fairness issues and the knowledge they encapsulate.

Acknowledgements

We extend our special gratitude to the Swiss National Science Foundation (SNSF) and Research Foundation – Flanders (FWO) for funding this work under grants 200021E_189458 and G094020N.

References

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2018. Hypernetwork knowledge graph embeddings. In *International Conference on Artificial Neural Networks*.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *CACM*.

Rajarshi Bhowmik and Gerard de Melo. 2020. Explainable link prediction for emerging entities in knowledge graphs. In *The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part I 19*, pages 39–55. Springer.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sanxing Chen, Hao Cheng, Xiaodong Liu, Jian Jiao, Yangfeng Ji, and Jianfeng Gao. 2023. Pre-training transformers for knowledge graph completion. *ArXiv*, abs/2303.15682.

Sanxing Chen, Xiaodong Liu, Jianfeng Gao, Jian Jiao, Ruofei Zhang, and Yangfeng Ji. 2021. HittER: Hierarchical transformers for knowledge graph embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10395–10407, Online and Punta Cana,

Dominican Republic. Association for Computational Linguistics.

Andrei Coman, Gianni Barlacchi, and Adrià de Gispert. 2023. Strong and efficient baselines for open domain conversational question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6305–6314, Singapore. Association for Computational Linguistics.

Andrei Coman, Christos Theodoropoulos, Marie-Francine Moens, and James Henderson. 2024. GADePo: Graph-assisted declarative pooling transformers for document-level relation extraction. In *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Damai Dai, Hua Zheng, Fuli Luo, Pengcheng Yang, Tianyu Liu, Zhifang Sui, and Baobao Chang. 2021. Inductively representing out-of-knowledge-graph entities by optimal estimation under translational assumptions. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 83–89, Online. Association for Computational Linguistics.

Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, page 365–374, New York, NY, USA. Association for Computing Machinery.

Daniel Daza, Michael Cochez, and Paul Groth. 2021. Inductive entity representations from text via link prediction. In *Proceedings of the Web Conference 2021*, WWW '21, page 798–808, New York, NY, USA. Association for Computing Machinery.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Takuma Ebisu and Ryutaro Ichise. 2018. Toruse: knowledge graph embedding on a lie group. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2017. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks : the official journal of the International Neural Network Society*, 107:3–11.

William Falcon and The PyTorch Lightning team. 2019. PyTorch Lightning.

Dieter A. Fensel, Umutcan Simsek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, and Alexander Wahler. 2020. Knowledge graphs: Methodology, tools and selected use cases. *Knowledge Graphs*.

Mikhail Galkin, Jiapeng Wu, E. Denis, and William L. Hamilton. 2021. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. *ArXiv*, abs/2106.12144.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1263–1272. JMLR.org.

Swapnil Gupta, Sreyash Kenkre, and Partha Talukdar. 2019. CaRe: Open knowledge graph embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 378–388, Hong Kong, China. Association for Computational Linguistics.

James Henderson, Alireza Mohammadshahi, Andrei Coman, and Lesly Miculicich. 2023. Transformers as graph-to-graph models. In *Proceedings of the Big Picture Workshop*, pages 93–107, Singapore. Association for Computational Linguistics.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2022. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations*.

Xiaotian Jiang, Quan Wang, and Bin Wang. 2019. Adaptive convolution for multi-relational learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 978–987, Minneapolis, Minnesota. Association for Computational Linguistics.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Pei Ke, Haozhe Ji, Yu Ran, Xin Cui, Liwei Wang, Linfeng Song, Xiaoyan Zhu, and Minlie Huang. 2021. JointGT: Graph-text joint representation learning for text generation from knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2526–2538, Online. Association for Computational Linguistics.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971, Florence, Italy. Association for Computational Linguistics.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. Commonsense knowledge base completion with structural and semantic context. *Proceedings of the 34th AAAI Conference on Artificial Intelligence*.

Elan Markowitz, Keshav Balasubramanian, Mehrnoosh Mirtaheri, Murali Annavaram, Aram Galstyan, and Greg Ver Steeg. 2022. StATIK: Structure and text for inductive knowledge graph completion. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 604–615, Seattle, United States. Association for Computational Linguistics.

Lesly Miculicich and James Henderson. 2022. Graph refinement for coreference resolution. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2732–2742, Dublin, Ireland. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Alireza Mohammadshahi and James Henderson. 2020. Graph-to-graph transformer for transition-based dependency parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3278–3289, Online. Association for Computational Linguistics.

Alireza Mohammadshahi and James Henderson. 2021. Recursive non-autoregressive graph-to-graph transformer for dependency parsing with iterative refinement. *Transactions of the Association for Computational Linguistics*, 9:120–138.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. 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, page 809–816, Madison, WI, USA. Omnipress.

Guanglin Niu, Bo Li, Yongfei Zhang, and Shiliang Pu. 2022. CAKE: A scalable commonsense-aware framework for multi-view knowledge graph completion. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2867–2877, Dublin, Ireland. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2814–2828, Dublin, Ireland. Association for Computational Linguistics.

M. Schlichtkrull, Thomas Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. In *Extended Semantic Web Conference*.

Phillip Schneider, Tim Schopf, Juraj Vladika, Mikhail Galkin, Elena Simperl, and Florian Matthes. 2022. A decade of knowledge graphs in natural language processing: A survey. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 601–614,

Online only. Association for Computational Linguistics.

Haseeb Shah, Johannes Villmow, Adrian Ulges, Ulrich Schwanecke, and Faisal Shafait. 2019. An open-world extension to knowledge graph completion models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3044–3051.

Noam M. Shazeer. 2020. Glu variants improve transformer. *ArXiv*, abs/2002.05202.

Baoxu Shi and Tim Weninger. 2018. Open-world knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Komal K. Teru, Etienne G. Denis, and William L. Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Christos Theodoropoulos, James Henderson, Andrei Catalin Coman, and Marie-Francine Moens. 2021. Imposing relation structure in language-model embeddings using contrastive learning. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 337–348, Online. Association for Computational Linguistics.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv: Computation and Language*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, Ying Wang, and Yi Chang. 2021a. Structure-augmented text representation learning for efficient knowledge graph completion. In *Proceedings of the Web Conference 2021*, WWW '21, page 1737–1748, New York, NY, USA. Association for Computing Machinery.

Hongwei Wang, Hongyu Ren, and Jure Leskovec. 2020. Relational message passing for knowledge graph completion. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294, Dublin, Ireland. Association for Computational Linguistics.

Peifeng Wang, Jialong Han, Chenliang Li, and Rong Pan. 2019a. Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Quan Wang, Pingping Huang, Haifeng Wang, Songtai Dai, Wenbin Jiang, Jing Liu, Yajuan Lyu, Yong Zhu, and Hua Wu. 2019b. Coke: Contextualized knowledge graph embedding. *arXiv preprint arXiv:1911.02168*.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame,

Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *AAAI Conference on Artificial Intelligence*.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. On layer normalization in the transformer architecture. *ArXiv*, abs/2002.04745.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *AAAI Conference on Artificial Intelligence*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Guangqian Yang, Yi Liu, Lei Zhang, Licheng Zhang, Hongtao Xie, and Zhendong Mao. 2024. Knowledge context modeling with pre-trained language models for contrastive knowledge graph completion. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8619–8630, Bangkok, Thailand. Association for Computational Linguistics.

Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. 2022. TableFormer: Robust transformer modeling for table-text encoding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 528–537, Dublin, Ireland. Association for Computational Linguistics.

Lin F. Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. 2023. Give us the facts: Enhancing large language models with knowledge graphs for fact-aware language modeling. *IEEE Transactions on Knowledge and Data Engineering*, 36:3091–3110.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *ArXiv*, abs/1909.03193.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform bad for graph representation? In *Neural Information Processing Systems*.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4961–4974, Dublin, Ireland. Association for Computational Linguistics.

Hanwen Zha, Zhiyu Chen, and Xifeng Yan. 2021. Inductive relation prediction by bert. *ArXiv*, abs/2103.07102.

Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2031–2043, Online. Association for Computational Linguistics.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. 2021. Neural bellman-ford networks: A general graph neural network framework for link prediction. In *Neural Information Processing Systems*.

A Appendix

A.1 Datasets statistics

Table 6 provides the statistics of the datasets used in our experiments. $\mathcal{E}$ represents the set of entities, $\mathcal{R}$ denotes the set of relation labels, $\mathcal{T}$ consists of the set of relation triples $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, and $E(e)$ shows the mean and standard deviation (μ_σ) of the number of neighbours in an entity's ego-graph.

A.2 Complexity of FnF-TG

Our method exhibits identical computational complexity to StATIk (Markowitz et al., 2022), with $O(N + Q)$ complexity, where N denotes the number of nodes in the graph and Q represents the number of queries $(h, r, \hat{e})$ and $(\hat{e}, r, t)$.

A.3 Training and Implementation Details

We provide details on the training and implementation of our models on three datasets: $\text{WN18RR}_{\text{IND}}$, $\text{FB15k-237}_{\text{IND}}$, and $\text{Wikidata5M}_{\text{IND}}$.

Seeds and Epochs We run our experiments with five different seeds $(73, 21, 37, 3, 7)$ for $\text{WN18RR}_{\text{IND}}$ and $\text{FB15k-237}_{\text{IND}}$, and two seeds $(73, 21)$ for $\text{Wikidata5M}_{\text{IND}}$ due to its large scale (see Table 6). We train our models for 40 epochs on $\text{WN18RR}_{\text{IND}}$ and $\text{FB15k-237}_{\text{IND}}$, and 5 epochs on $\text{Wikidata5M}_{\text{IND}}$, following previous works (Daza et al., 2021; Markowitz et al., 2022).

Dataset	$\mathcal{R}$	$\mathcal{E}_{train}$	$\mathcal{T}_{train}$	$E(e)_{train}$	$\mathcal{E}_{val}$	$\mathcal{T}_{val}$	$E(e)_{val}$	$\mathcal{E}_{test}$	$\mathcal{T}_{test}$	$E(e)_{test}$
WN18RR$_{\text{IND}}$	11	$32,755$	$69,585$	$2,12_{3,15}$	$4,094$	$11,381$	$1,17_{1,33}$	$4,456$	$12,037$	$1,18_{1,35}$
FB15K-237$_{\text{IND}}$	237	$11,633$	$215,082$	$18,49_{28,91}$	$1,454$	$42,164$	$4,70_{10,63}$	$2,416$	$52,870$	$4,97_{12,29}$
Wikidata-5M$_{\text{IND}}$	822	$4,579,609$	$20,496,514$	$4,48_{4,41}$	$7,374$	$6,699$	$0,91_{0,78}$	$7,475$	$6,894$	$0,92_{0,81}$

Table 6: WN18RR$_{\text{IND}}$, FB15K-237$_{\text{IND}}$, and Wikidata-5M$_{\text{IND}}$ datasets statistics. $\mathcal{E}$ represents the set of entities, $\mathcal{R}$ denotes the set of relation labels, $\mathcal{T}$ consists of the set of relation triples $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, and $E(e)$ shows the mean and standard deviation (μ_σ) of the number of neighbours in an entity's ego-graph.

Hyperparameters We set the number of sampled neighbors per entity based on the dataset statistics (Table 6): 10 for WN18RR$_{\text{IND}}$, 40 for FB15k-237$_{\text{IND}}$, and 1 for Wikidata5M$_{\text{IND}}$. We use 24 words of text for each x_{KG} in WN18RR$_{\text{IND}}$ and FB15k-237$_{\text{IND}}$, and 64 words for Wikidata5M$_{\text{IND}}$.

Graph Transformer Encoder We implement the *Graph Transformer Encoder* layer using a pre-LayerNorm Transformer (Xiong et al., 2020) with a SwiGLU-type pointwise feed-forward network (Shazeer, 2020). We use a single GT layer, as multiple layers did not improve performance while increasing latency.

Optimisation We set the learning rate to $1e^{-5}$ for a batch size of 32 and scale it proportionally with the batch size following a power-of-2 rule to fit the GPU budget. We use RAdam (Liu et al., 2020) as our optimiser and a cosine learning rate decay throughout the training process.

Libraries We develop our models using PyTorch (Paszke et al., 2019), Lightning (Falcon and The PyTorch Lightning team, 2019), and Hugging Face's Transformers (Wolf et al., 2020) libraries.

A.4 Computational Budget

We fix our computational budget to a constant consumer-grade GPU (NVIDIA RTX3090 24GB) as stated in Subsection 3.6 and report the GPU budget per run for each dataset on FnF-TG$_{\text{BERT}_{\text{BASE}}}$ relative to the largest text encoders. The GPU budget per run is 4 GPU/h for WN18RR$_{\text{IND}}$, 6 GPU/h for FB15k-237$_{\text{IND}}$, and 40 GPU/h for Wikidata5M$_{\text{IND}}$.