On Masked Language Models for Contextual Link Prediction

Angus Brayne **Maciej Wiatrak** **Dane Corneil**
BenevolentAI
4-8 Maple St, London
{angus.brayne, maciej.wiatrak, dane.corneil}@benevolent.ai

Abstract

In the real world, many relational facts require context; for instance, a politician holds a given elected position only for a particular timespan. This context (the timespan) is typically ignored in knowledge graph link prediction tasks, or is leveraged by models designed specifically to make use of it (i.e. *n-ary* link prediction models). Here, we show that the task of n-ary link prediction is easily performed using language models, applied with a basic method for constructing cloze-style query sentences. We introduce a pre-training methodology based around an auxiliary entity-linked corpus that outperforms other popular pre-trained models like BERT, even with a smaller model. This methodology also enables n-ary link prediction without access to any n-ary training set, which can be invaluable in circumstances where expensive and time-consuming curation of n-ary knowledge graphs is not feasible. We achieve state-of-the-art performance on the primary n-ary link prediction dataset WD50K and on WikiPeople facts that include literals - typically ignored by knowledge graph embedding methods.

1 Introduction

Large-scale knowledge graphs (KGs) have gained prominence over the past several decades as a means for representing complex structured data at scale, leading to the development of machine learning models designed to predict new or unknown information from a KG (Ji et al., 2021). A subclass of such models deals with *link prediction*, i.e. inferring new facts from a given KG consisting of (*subject*, *relation*, *object*) *triples*. For instance, a link prediction model might reason from a KG containing the triple (*USA*, *ElectedPresident*, *JFK*) to infer that the triple (*JFK*, *BornInCountry*, *USA*) also likely exists (i.e. *JFK was born in the country USA*).

The triple format is often too restrictive to represent a query effectively. For instance, the query

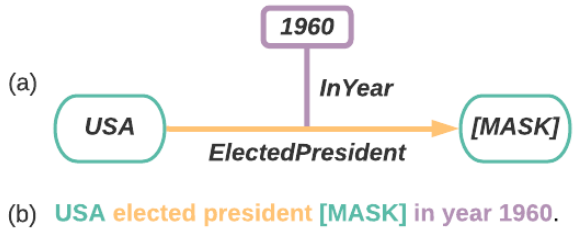

Figure 1: N-ary query representation in KG vs. natural language frameworks. (a) In a knowledge graph, the primary triple query (*USA*, *ElectedPresident*, [MASK]) is augmented with an auxiliary link for qualifer information (*InYear*, *1960*). Each entity or relationship is represented by a unique identifier. Qualifiers require the use of specialised encoder architectures; literal qualifiers like *1960* typically cannot be used at all. (b) We instead represent the query in a templated language model, where the qualifier detail can be directly appended.

Who was elected President of the United States in 1960? permits multiple correct answers when simplified to the triple format (*USA*, *ElectedPresident*, [MASK]), in the absence of the context *1960* (also referred to as a *qualifier* (Vrandečić and Krötzsch, 2014)). Recently, several KG completion models have been developed aimed specifically at link prediction in the presence of qualifiers, collectively referred to as hyper-relational or *n-ary* link prediction models (Wen et al., 2016; Zhang et al., 2018; Guan et al., 2019; Liu et al., 2020; Rosso et al., 2020; Galkin et al., 2020; Yu and Yang, 2021; Wang et al., 2021b). Usage of qualifiers becomes particularly difficult when they include literals, i.e. values that cannot be efficiently represented as discrete graph entities. Examples of literals include years (like *1960*), times, or numerals. Existing KG completion algorithms typically remove literals (Rosso et al., 2020; Galkin et al., 2020) or use specialised techniques to leverage them (Kristiadi et al., 2019).

The need for new models to leverage qualifiers and literals reveals some fundamental weaknesses in discrete, triple-based knowledge graph represen-

Proceedings of Deep Learning Inside Out (DeeLIO 2022):
The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 87 - 99

tations. Unlike graphs, written languages clearly permit the use of qualifiers and literals to represent facts and queries. Pre-trained language models like BERT (Devlin et al., 2019) have already shown competitive performance compared to existing KG link prediction approaches on triple-based KGs (Clouâtre et al., 2021; Yao et al., 2019). As such, it is natural to ask whether Language Models (LMs) present a better alternative for inferring facts with qualifiers and literals compared to n-ary KG inference models.

Apart from their ability to represent qualifiers and literals, using LMs with novel pre-training methodologies on vast corpora also presents opportunities to enable n-ary link prediction without access to any n-ary training set. The need to construct large, partially complete n-ary knowledge graphs in new domains is an expensive and time-consuming requirement of link prediction (Nicholson and Greene, 2020).

Here, we present Hyper Relational Link Prediction using an auxiliary Entity Linked Corpus (Hyper-ELC), the first fully natural-language-based approach applied to KG link prediction benchmarks containing qualifiers and literals. We make use of model pre-training to leverage the large corpora directly available to language models, applying a simple entity-linking approach to prime the model for later inference on named KG entities and to enable link prediction without access to any n-ary training set. To our knowledge, this is the first approach to link prediction without KG supervision. We also use fine-tuning to specifically focus Hyper-ELC on the types of queries represented in the training set. By using KG link prediction datasets, we can directly compare language models to KG models specifically designed to take advantage of additional context in form of qualifiers and literals. Our results show competitive performance compared to these link prediction models, suggesting that language models provide a performant and practical alternative to KG models for link prediction beyond triple-based datasets.

2 Related Work

2.1 N-Ary Link Prediction

Several models have been developed over the past decade to learn from and infer on n-ary relationships. This has been driven by the recognition that knowledge bases like Freebase (Bollacker et al., 2008) contain a sizeable number of relationships involving more than two named entities. Wen et al. (2016) generalized the triple-based translational embedding model *TransH* (Wang et al., 2014) to hyper-relational facts. Zhang et al. (2018) extended this approach using a binary loss learned from the probability that any two entities participate in the same n-ary fact.

Unlike these earlier embedding-based models, **NaLP** (Guan et al., 2019) addressed the n-ary link prediction problem with a neural network, representing n-ary facts as permutation-invariant sets of role-value pairs. Liu et al. (2020) developed the first tensor decomposition-based approach to the problem, adapting earlier tensor decomposition methods applied to link prediction in triple-based KGs. **HINGE** (Rosso et al., 2020) applied a convolutional network to the underlying triples and qualifiers in an n-ary fact.

More recently, several specialised n-ary prediction models have been developed by combining knowledge graph embeddings with attention-based transformer architectures (Vaswani et al., 2017); namely **StarE** (Galkin et al., 2020), **Hy-Transformer** (Yu and Yang, 2021) and **GRAN** (Wang et al., 2021b). In the StarE model, embeddings are fed through a graph neural network before entering the transformer layer. Hy-Transformer and GRAN instead feed the processed embeddings into the transformer directly. Hy-Transformer also adds a qualifier prediction-based auxiliary task, while GRAN modifies the transformer attention model to represent the link structure of the n-ary input. Together, these three transformer-based models have achieved state-of-the-art performance on the n-ary link prediction task.

Hyper-ELC differs from other n-ary link prediction models in that it represents facts in natural language, eliminating the need for specialised encoders or graph-based methods and introducing the ability to pre-train on massive natural language corpora. By representing facts as token sequences, earlier modelling constraints can be avoided; e.g. multiple arities can be supported with the same model (unlike Liu et al. (2020)), and structural information can be retained in token positional encodings, unlike Wen et al. (2016) and Guan et al. (2019). The pre-training introduced here also enables prediction on the downstream task without access to any n-ary training set. Nonetheless, like the most recent approaches, we also use a transformer architecture. In particular, Hyper-ELC is

most similar to Hy-Transformer and GRAN, with named graph entities exchanged for word tokens with positional embeddings.

2.2 Literals in Link Prediction

Parallel to research on incorporating qualifiers, several groups have investigated leveraging numerical attributes of entities in triple-based KG completion tasks (García-Durán and Niepert, 2017; Tay et al., 2017; Wu and Wang, 2018; Kristiadi et al., 2019). In these models, the numerical literals are general attributes associated with one of the entities involved in the triple (e.g. the latitude of a city entity); conversely, in the tasks we consider here, literals directly participate in n-ary facts. Nonetheless, we note that our approach could be straightforwardly applied to numerical attributes as well, by inserting them into the textual templates.

Hyper-ELC also differs from previous models by using a standard word-piece tokenisation approach to efficiently parse the literal data. While some literals, like *1962*, are single tokens in the BERT base uncased vocabulary, less commonly discussed dates are split into multiple tokens - for example *1706* becomes *170* and *##6*. Additionally, pre-training gives the model additional context to learn the relationships between dates - e.g. that similar people and events are discussed in sentences containing *1961* and sentences containing *1962*, revealing a similarity.

Notably, literal attributes composed of textual descriptions have also been investigated in KG completion, e.g. Xie et al. (2016); Xu et al. (2016). While we focus on numerical literals here, our natural language-based approach could also be extended to general textual attributes.

2.3 Language Models for Link Prediction

The success of large pre-trained language models has motivated multiple investigations into whether they can be used as knowledge bases. Petroni et al. (2019) proposed a benchmark for evaluating factual knowledge present in LMs with cloze-style queries. Their work has been further extended to probing areas including semantic (Ettinger, 2020; Wallace et al., 2019), commonsense (Tamborrino et al., 2020; Forbes et al., 2019; Roberts et al., 2020), and linguistic (Lin et al., 2019; Tenney et al., 2019) knowledge. Furthermore, in order to improve the performance of LMs in extracting factual knowledge, Jiang et al. (2020) and Shin et al. (2020) proposed methods for automatic discovery and creation of cloze-style queries. This body of work focuses mainly on predicting tokens for filling in blanks, rather than ranking unique entity IDs, as we do here, and therefore requires an entity disambiguation post-processing step. It also focuses on comparison to open-domain question answering or relation extraction approaches rather than link prediction.

Several groups have proposed using LMs for triple-based link prediction. Yao et al. (2019) proposed KG-BERT, which encodes a triple as a sequence, where the entities and relation are separated by a [SEP] token and represented by their textual descriptions. They train to classify whether an individual triplet is correct or not, scoring every (h, r, ?) and (?, r, t) triplet to be ranked. This approach can involve millions of inference steps for a single completion. This work was extended for improved efficiency and performance in Kim et al. (2020); Wang et al. (2021a). This methodology, including entity separation and precise entity descriptions, diverges from plain masked text and is therefore incompatible with our simple pre-training approach that enables n-ary link prediction without access to a training knowledge graph.

An alternative approach to triple-based link prediction is MLMLM (Clouâtre et al., 2021), which also improves on KG-BERT's inference complexity with respect to the number of entities in the KG. They instead use the MLM setup to generate the logits for the tokens required to rebuild all of the entities. These logits are used alongside mean likelihood sampling to rank all entities. The head entity prediction input includes the head entity mask, relation, tail entity and tail entity definition. The tail entity prediction input is analogous. Unlike KG-BERT and its extensions, this method shares the MLM setup with our approach, however they predict tokens rather than unique entity ids. The maximum number of tokens of all of the entities is predicted for each example - predicting the pad token if necessary. This has the benefit that they can predict previously unseen entities (as long as they have fewer than the maximum number of tokens). However, again, this work requires entity disambiguation to go from tokens to a unique entity.

Finally, none of the language model approaches discussed above have been adapted to higher order link prediction with qualifiers and literals. Hyper-ELC additionally extends upon these approaches with a task-specific pre-training approach that en-

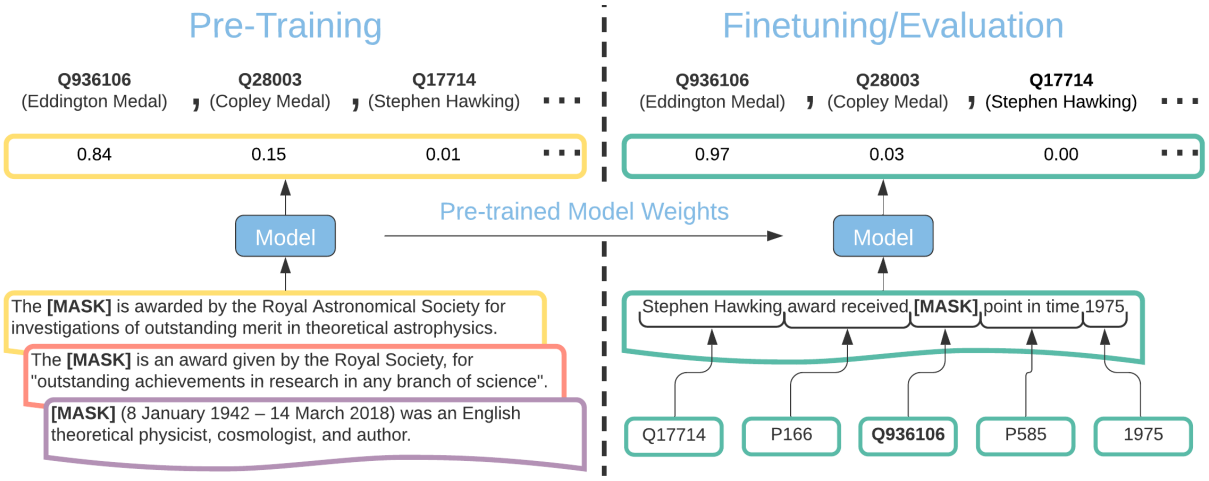

Figure 2: Overview of the training procedure. The names in brackets below the labels are purely informative; as in the typical link prediction setup, we rank the unique identifiers. [Left] Entities of interest in the pre-training corpus are linked and replaced with mask tokens; the model is trained to predict the corresponding named entity of interest. [Right] The finetuning task is the same, but performed on automatically generated sentences from the train set. Surface forms are used for the other entities in each fact.

ables us to perform this task without access to a training knowledge graph.

3 Definitions

A hyper-relational (n-ary) graph, made up of hyper-relational facts, can be defined as $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices (entities), $\mathcal{R}$ is the set of relations, and $\mathcal{E}$ is a set $(e_1, \ldots, e_n)$ of edges with $e_j \in \mathcal{V} \times \mathcal{R} \times \mathcal{V} \times \mathcal{P}(\mathcal{R} \times \mathcal{V})$ for $1 \leq j \leq n$. Here, $\mathcal{P}$ denotes the power set.

A hyper-relational fact $e_j \in \mathcal{E}$ is written as a tuple $(s, r, o, \mathcal{Q})$, with $s, o \in \mathcal{V}$ and $r \in \mathcal{R}$. Here, $\mathcal{Q}$ is the set of qualifier pairs (qr_i, qv_i) with qualifier relations $qr_i \in \mathcal{R}$ and qualifier values $qv_i \in \mathcal{V}$. An example of a fact in this representation would be (*StephenHawking* (s), *AwardReceived* (r), *EddingtonMedal* (o), (*PointInTime* (qr_1), *1975* (qv_1))).

4 Methods

Our approach consists of three stages:

1. **Pre-training** to predict the unique identifier of a masked entity in the sentences of an auxiliary *entity linked corpus*.

2. **Finetuning** on sentence-like natural language templates created from the training set of the *n-ary link prediction dataset*.

3. **Evaluation** on the test set of the n-ary link prediction dataset using the same format of natural language templates.

For a visual representation of the process, see Figure 2.

4.1 Pre-Training

Our method may use any corpus that references the entities of interest and any entity linking methodology for recognising them within the corpus. As we use the entity linked corpus only in pre-training and not for evaluation, we do not require it to be gold standard. However, increased coverage and precision of the linking may result in better downstream performance.

Each pre-training example is a tuple consisting of a unique entity ID and a masked sentence in which that entity occurs. In the sentence, the span of every occurrence of the entity of interest is replaced by a "[MASK]" token. A single unique entity is masked in each example while all other entities are left as plain text. For example, the label for the entity *StephenHawking* is Q17714 and a masked sentence would be: "[MASK] (8 January 1942 – 14 March 2018) was an English theoretical physicist, cosmologist, and author."

4.2 Finetuning and Evaluation

In order to use our pre-trained language model for the n-ary link prediction task, we must format the query in natural language as a cloze-style sentence. This may be done in any way that represents the query, but linguistic alignment with the pre-training corpus may benefit performance (Jiang et al., 2020; Shin et al., 2020).

Dataset	Statements		Statements w/ Qualifiers (%)		Statements w/ Literals (%)		Entities
	Train	Test	Train	Test	Train	Test	
WikiPeople Pre	37.4M	380,396	—	—	—	—	29,720
WikiPeople	294,439	37,712	2.6	2.6	0	0	34,839
WikiPeople Lit	294,439	3,906	12.1	100	10.9	100	34,839
WD50K Pre	48.6M	494,881	—	—	—	—	42,800
WD50K	166,435	46,159	13.8	13.1	0	0	47,155
WD50K (100)	22,738	5,297	100	100	0	0	18,791

Table 1: Statistics of the datasets used in the experiments. The "Pre" and "Lit" labels on the datasets indicate pre-training and literal datasets, respectively. "M" indicates million. Validation set statistics have been left out for brevity, but they follow a similar pattern to the test set statistics. In the original WikiPeople source data, 10.9% of statements have literals in the qualifiers. The source data also includes 12,363 (3.3%) statements with a literal in the tail position, which are removed from all datasets.

One simple approach is to space separate the entities, relationships and roles in the $(s, r, o, \mathcal{Q})$ order (Figure 2) described in Section 3. This requires that each of the entities have associated textual names, which is usually the case in knowledge graphs.

4.3 Model

Our models are all based on the Transformer architecture (Vaswani et al., 2017), more specifically BERT (Devlin et al., 2019). However, we found a smaller version of the BERT architecture to be more stable during pre-training, which enabled a higher learning rate and larger batch size (see Table 5 in the Appendix). We use the BERT base uncased word-piece tokenisation for all text-based models.

We use a single linear layer as a decoder, followed by a softmax. For optimisation, we leverage a standard categorical cross-entropy loss. All of our models are trained with the Adam optimiser, and are regularised via dropout and gradient clipping. We follow the same setup during pre-training and finetuning. We believe that this alignment between pre-training and the downstream task is part of what makes this approach so powerful. Note that the pre-trained model can also be applied on the downstream task even without additional finetuning on a training graph (Section 6.3).

5 Datasets

5.1 WikiPeople and WD50K

For finetuning and evaluation we use two n-ary link prediction datasets: WikiPeople[1] (Guan et al., 2019) and WD50K[2] (Galkin et al., 2020). Both WikiPeople and WD50K are extracted from Wikidata and contain a mixture of binary and higher-order facts. WikiPeople is a commonly used benchmark containing facts related to entities representing humans.

WD50K was created by Galkin et al. (2020) from the 2019/08/01 Wikidata dump[3]. It was developed with the goal of containing a higher proportion of non-literal higher-order relationships. It is based on the entities from FB15K-237 (Bordes et al., 2013) that have a direct mapping in Wikidata.

In order to transform the facts in these datasets into natural language queries, we use the English Wikidata names for each of the entity and relationship/role IDs[4]. We then create templates in the simple manner described in Section 4.2. We find that while the queries are not particularly natural in their structure and vocabulary, their meaning remains largely the same (an example template is shown in Figure 2, right).

5.2 Non-Named Entity Qualifiers

Galkin et al. (2020) noted that most of the qualifier values in WikiPeople are literals, in this case datetime instances. Literals appear in approximately 13% of the statements in the WikiPeople dataset, but they are typically ignored in knowledge graph embedding approaches (Rosso et al., 2020). If the literals are ignored, only 2.6% of statements in WikiPeople are higher-order. None of the previous approaches to this dataset encode literals.

Note that, for evaluation purposes, alternative correct entities are filtered from the ranking at evaluation time when assessing a given potential answer (Bordes et al., 2013). This has implications

[1]Downloaded from: https://github.com/gsp2014/NaLP/tree/master/data/WikiPeople

[2]Downloaded from: https://zenodo.org/record/4036498

[3]https://dumps.wikimedia.org/wikidatawiki/20190801/

[4]https://www.wikidata.org/wiki/Special:EntityData

for treating literals. Consider the case where literals are ignored: when evaluating whether the model correctly predicted *EddingtonMedal* as a completion for the fact (*StephenHawking*, *AwardReceived*, [MASK], (*PointInTime*, *1975*)), the entity *CopleyMedal* would be filtered out of the ranking if the fact (*StephenHawking*, *AwardReceived*, *CopleyMedal*, (*PointInTime*, *2006*)) also exists in the dataset. This occurs because the *PointInTime* qualifier is ignored, so that the subject and relation of the facts are identical (and both medals are equally valid completions). When literal-containing qualifiers are not ignored, the facts are distinct, with only one correct answer for each.

The primary WikiPeople dataset used here was adapted by Rosso et al. (2020) from the original WikiPeople (Guan et al., 2019). To investigate whether this literal data can be leveraged by our model, we generated a new dataset from a subset of WikiPeople that we call *WikiPeople Literal*. Unlike in Rosso et al. (2020) and Galkin et al. (2020), where literal qualifier terms are ignored when filtering the rankings for evaluation, we include the literal terms during filtering in *WikiPeople Literal*. Additionally, we evaluate only on facts that include at least one literal. This focus enables us to probe the model's ability to interpret literal qualifiers.

Following Rosso et al. (2020), we drop all statements that contain literals in the main triple.

5.3 Entity Linked Corpus

For pre-training we create an entity linked corpus based on the 2019/08/01 English Wikipedia[5] dump used in BLINK (Ledell Wu, 2020). We process the XML with Gensim[6], which we adapt to leave article hyperlinks in the text.

For simplicity, we use a regex to find occurrences of the entities of interest in the large hyperlinked Wikipedia corpus. For each article we extract the title entity and all of the hyperlinked entities, along with their surface forms in the text and their title name in the hyperlink. We find the wikidata IDs for each of these entities[7] and we retain those entities that are in our downstream n-ary dataset. We then split the article into sentences and run a case insensitive regex over each sentence to find the spans of these entities and link them to their Wikidata IDs, using the ID to surface form/title name dictionaries. Given this collection of entity linked sentences, we create the pre-training examples as described in Section 4.1.

6 Experiments

Throughout this section we compare to the following external baselines developed for n-ary link prediction: (i) NaLP-Fix (Rosso et al., 2020), (ii) HINGE (Rosso et al., 2020), (iii) StarE (Galkin et al., 2020), (iv) Hy-Transformer (Yu and Yang, 2021), and (v) GRAN (Wang et al., 2021b). NaLP-Fix is an improved version of the original NaLP model (Guan et al., 2019). None of these methods make predictions over natural language and none of them encode literals.

The metrics that we use are based on predicting only the subject and object of the hyper-relational facts. We follow the *filtered* setting introduced by Bordes et al. (2013) as discussed in Section 5.2 to ensure that corrupted facts are not valid facts from the rest of the dataset. For each test example, we filter from the model's predicted ranking all of the entities that appear in the same position in otherwise identical examples in either the training, validation or test set (except the test entity of interest). We consider mean reciprocal rank (MRR) and hits at 1 and 10 (H@1 and H@10 respectively).

6.1 Link Prediction with Literals

In order to showcase the expressive power of natural language as a representation, we employ an experiment that involves making predictions with non-named entity qualifier terms (i.e. literals). We use an evaluation dataset (described in Section 5.2) that contains only the examples in the WikiPeople dataset that have at least one literal qualifier. Additionally, we consider these qualifiers when filtering the ranking at evaluation time, unlike the typical WikiPeople evaluation.

To the best of our knowledge, no existing works leverage literals in qualifiers, so no strong baselines exist. We therefore use two baselines that cannot leverage literals as comparison points. The first, **Hyper-ELC [UNK]**, is an ablated version of our model that replaces any literal entity with the [UNK] token. We also used the publicly-available StarE repository [8] to reproduce StarE performance

[5] http://dl.fbaipublicfiles.com/BLINK/enwiki-pages-articles.xml.bz2

[6] https://github.com/RaRe-Technologies/gensim/blob/develop/gensim/corpora/wikicorpus.py

[7] https://dumps.wikimedia.org/wikidatawiki/latest/wikidatawiki-latest-wb_items_per_site.sql.gz

[8] Hy-Transformer did not have a published codebase, and we were unable to successfully run the published GRAN code.

Method	WikiPeople Literal			WikiPeople		
	MRR	H@1	H@10	MRR	H@1	H@10
NaLP-Fix	—	—	—	0.420	0.343	0.556
HINGE	—	—	—	0.476	0.415	0.585
StarE	0.246	0.161	0.424	0.491	0.398	**0.648**
Hy-Transformer	—	—	—	**0.501**	0.426	0.634
GRAN	—	—	—	**0.503**	**0.438**	0.620
Hyper-ELC [UNK]	0.211	0.141	0.347	0.415	0.325	0.566
Hyper-ELC	**0.322**	**0.226**	**0.519**	0.440	0.348	0.592

Table 2: Performance comparison on the two WikiPeople-derived datasets. WikiPeople Literal evaluates only on examples with literal qualifiers (about 10.9% of the full test set) and filters ranking for evaluation with literals included. Methods above the line can encode literal terms, while methods below can't.

on literal-containing qualifiers after adding them back into the dataset (note that StarE achieves state-of-the-art on the full dataset on Hits@10).

On the WikiPeople Literal dataset, Hyper-ELC significantly outperformed both StarE and Hyper-ELC [UNK] (Table 2, first three columns). In particular, the performance boost over the [UNK] ablation illustrates that our model specifically makes use of the information represented in literal qualifiers.

Hyper-ELC also performed reasonably well on the standard WikiPeople dataset (Table 2, last three columns), outperforming NaLP-Fix, but with lower overall performance than the most recent baselines (StarE, Hy-Transformer and GRAN).

To investigate the differences between Hyper-ELC and the other state-of-the-art baselines on WikiPeople, we examined the MRR performance ratio of StarE compared to Hyper-ELC for the relationship-entity position (i.e. head or tail) pairs that occur more than 500 times in the evaluation set (see Appendix, Table 6 in the appendix). Notably, Hyper-ELC displayed the most pronounced performance deficit compared to StarE on inferring correct entities in one-to-many relationships with many possible answers. In Section 7, we discuss potential reasons for this deficit and possible future improvements.

6.2 Link Prediction with Named Entities Only

Next, we evaluated Hyper-ELC on the WD50K datasets (Table 3), which do not contain any literal entities. WD50K (100) has been created by filtering WD50K to have 100% higher order relationships.

In order to understand the value of the pre-training and finetuning steps, we consider multiple ablation models:

Hyper-ELC (only P): a pre-trained version of Hyper-ELC without any exposure to the templated finetuning data (the train set).

Hyper-ELC (only F): a randomly initialised (i.e. only finetuned) version of Hyper-ELC.

BERT (only F): a BERT model (base uncased) with its own initialisation followed by a randomly initialised classification layer, finetuned.

On the full WD50K dataset, Hyper-ELC achieved an MRR of 0.354, nearly identical to the state-of-the-art Hy-Transformer with 0.356. While Hy-Transformer achieved the best performance on Hits@1, Hyper-ELC achieved state-of-the-art on Hits@10.

On the smaller, purely hyper-relational WD50K (100) dataset, Hyper-ELC performed comparably to StarE but was outperformed by Hy-Transformer (see discussion in Section 7).

6.3 Link Prediction without a Training Graph

Finally, we focus specifically on hyper-relational link prediction with the ablated version of Hyper-ELC exposed only to the pre-training data (Table 3, last row, and Table 4). Hyper-ELC (only P) has some ability to perform inference, without any access to the training knowledge graph; it achieves an MRR of 0.087 and 0.207 on WD50K and WD50K (100) respectively, compared to 0.0003 and 0.0006 for the random model and 0.356 and 0.699 for the state-of-the-art Hy-Transformer. This approach could be very powerful in domains where expensive and time consuming curation of hyper-relational knowledge graphs is not feasible.

The significant performance difference between Hyper-ELC and Hyper-ELC (only P) can likely be partially attributed to the distributional shift in the language from pre-training to the templated format used in finetuning and evaluation on the "Ba-

Method	WD50K			WD50K (100)		
	MRR	H@1	H@10	MRR	H@1	H@10
NaLP-Fix	0.177	0.131	0.264	0.458	0.398	0.563
HINGE	0.243	0.176	0.377	0.492	0.417	0.636
StarE	0.349	0.271	0.496	0.654	0.588	0.777
Hy-Transformer	**0.356**	**0.281**	0.498	**0.699**	**0.637**	**0.812**
Hyper-ELC	**0.354**	0.273	**0.508**	0.642	0.564	0.789
Hyper-ELC (only F)	0.283	0.214	0.415	0.549	0.475	0.688
BERT (only F)	0.29	0.22	0.43	0.609	0.536	0.748
Random	< 0.001	< 0.0001	< 0.001	< 0.001	0.00	< 0.001
Hyper-ELC (only P)	0.087	0.051	0.157	0.207	0.129	0.360

Table 3: Performance comparison on the WD50K datasets. We train and test on the dataset indicated following the approach used by the baselines. Model names "only P" and "only F" indicate that only pre-training or finetuning was performed respectively. Methods above the line use the n-ary training graph, while those below do not.

Method	Dataset	WD50K (100)
		MRR
Random	—	< 0.001
Hyper-ELC (only P)	Basic	0.207
Hyper-ELC (only P)	Cleaned	**0.232**
Hyper-ELC	Basic	0.642
Hyper-ELC	Cleaned	0.645

Table 4: With some minor adjustments to the wording of some of the most frequent relationships/roles, to move from the "Basic" to the "Clean" dataset, we can boost performance for the model that doesn't have access to graph based training data. Here, "only P" indicates only pre-training, without finetuning.

sic" dataset, where the templates are often stilted and ungrammatical. To test the hypothesis that improved templates could drive improved performance, we considered 37 of the roles/relationships that occur most frequently in the WD50K (100) training dataset and altered some to make the templates for the "Clean" dataset to be more similar to the natural language occurring in the Wikipedia pre-training corpus; for instance, we improved the grammar with stop words like "the". Table 7 in the appendix shows the 37 roles/relationships that we considered and the changes that we made. In Table 4 we can see a performance increase from 0.207 MRR to 0.232 for Hyper-ELC (only P) with these simple template changes. However, we saw only a minimal improvement when finetuning was introduced, from 0.642 MRR to 0.645, suggesting that the model adapts effectively to the templated linguistic style with finetuning.

7 Discussion and Future Work

Here, we presented Hyper-ELC, the first purely natural language-based approach to n-ary link prediction and the first model to leverage literals in n-ary qualifiers. The natural language-based approach allows us to take advantage of pre-training on massive entity-linked corpora and easily leverage the detail present in hyper-relational facts.

Hyper-ELC matched state-of-the-art performance on WD50K and established state-of-the-art on a version of WikiPeople containing only literal qualifiers. However, it did not reach the performance of existing KG models on the full WikiPeople dataset. As shown in Table 6, Hyper-ELC tends to perform significantly worse than StarE on one-to-many relationships; e.g. ([MASK], *SexOrGender*, *Male*). One hypothesis for this result is that the softmax loss function used in training the model assumes a single correct answer out of all entities for a given masked template; for each unique training example, all competing entities (including valid ones) are treated as false. The objective function and negative sampling approach are therefore potential areas for investigation in future work.

In addition, we expect performance improvements by increasing coverage of relevant information for the entities of interest in the pre-training dataset. The WD50K and WikiPeople pre-training datasets only have 88.2% and 85.3% coverage of the WD50K and WikiPeoople entities, respectively. This could be achieved by improving the quality of the entity linking methodology used. Simple improvements could be made to our regex method, such as including the WikiData surface forms in the regex dictionaries. Even greater improvements could likely be made with feature based or neural

entity linking methodologies.

Finally, we found that Hy-Transformer had the best performance on WD50K (100), though HyperELC performed similarly to or better than the other KG baselines. Yu and Yang (2021) propose that Hy-Transformer's auxiliary masked qualifier prediction task allows it to better leverage the train set, which could explain why Hy-Transformer performs well on the smaller train set in WD50K (100). A similar qualifier prediction task could also be investigated in the context of a language model, which we leave for future work.

Overall, our results show how a language model can leverage weakly relevant data (an entity-linked corpus) to reach strong performance on a complex link prediction task. In particular, we note that many practical relational inference problems do not exist in isolated domains where only a structured KG model is available; rather, they are loosely informed by massive, readily available unstructured natural language datasets. In these cases, the sheer quantity and variety of data available to language models, combined with their inherent flexibility in representing context, may swing the balance in their favour.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

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.

Louis Clouâtre, Philippe Trempe, Amal Zouaq, and A. P. Sarath Chandar. 2021. Mlmlm: Link prediction with mean likelihood masked language model. In *FINDINGS*.

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.

Allyson Ettinger. 2020. What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense? *CoRR*, abs/1908.02899.

Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. 2020. Message passing for hyper-relational knowledge graphs. *arXiv preprint arXiv:2009.10847*.

Alberto García-Durán and Mathias Niepert. 2017. Kblrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features. *arXiv preprint arXiv:1709.04676*.

Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2019. Link prediction on n-ary relational data. In *The World Wide Web Conference*, pages 583–593.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Bosung Kim, Taesuk Hong, Youngjoong Ko, and Jungyun Seo. 2020. Multi-task learning for knowledge graph completion with pre-trained language models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1737–1743, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Agustinus Kristiadi, Mohammad Asif Khan, Denis Lukovnikov, Jens Lehmann, and Asja Fischer. 2019. Incorporating literals into knowledge graph embeddings. In *International Semantic Web Conference*, pages 347–363. Springer.

Martin Josifoski Sebastian Riedel Luke Zettlemoyer Ledell Wu, Fabio Petroni. 2020. Zero-shot entity linking with dense entity retrieval. In *EMNLP*.

Yongjie Lin, Yi Chern Tan, and Roberta Frank. 2019. Open sesame: Getting inside bert's linguistic knowledge. *ArXiv*, abs/1906.01698.

Yu Liu, Quanming Yao, and Yong Li. 2020. Generalizing tensor decomposition for n-ary relational knowledge bases. In *Proceedings of The Web Conference 2020*, pages 1104–1114.

David N. Nicholson and Casey S. Greene. 2020. Constructing knowledge graphs and their biomedical applications. *Computational and Structural Biotechnology Journal*, 18:1414–1428.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *Proceedings of The Web Conference 2020*, pages 1885–1896.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Alexandre Tamborrino, Nicola Pellicanò, Baptiste Pannier, Pascal Voitot, and Louise Naudin. 2020. Pre-training is (almost) all you need: An application to commonsense reasoning. *ArXiv*, abs/2004.14074.

Yi Tay, Luu Anh Tuan, Minh C Phan, and Siu Cheung Hui. 2017. Multi-task neural network for non-discrete attribute prediction in knowledge graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1029–1038.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *ArXiv*, abs/1905.06316.

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.

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

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in 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 5307–5315, Hong Kong, China. Association for Computational Linguistics.

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.

Quan Wang, Haifeng Wang, Yajuan Lyu, and Yong Zhu. 2021b. Link prediction on n-ary relational facts: A graph-based approach. *arXiv preprint arXiv:2105.08476*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. On the representation and embedding of knowledge bases beyond binary relations. *arXiv preprint arXiv:1604.08642*.

Yanrong Wu and Zhichun Wang. 2018. Knowledge graph embedding with numeric attributes of entities. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 132–136.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Jiacheng Xu, Kan Chen, Xipeng Qiu, and Xuanjing Huang. 2016. Knowledge graph representation with jointly structural and textual encoding. *arXiv preprint arXiv:1611.08661*.

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

Donghan Yu and Yiming Yang. 2021. Improving hyper-relational knowledge graph completion. *arXiv preprint arXiv:2104.08167*.

Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. 2018. Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *Proceedings of the 2018 World Wide Web Conference*, pages 1185–1194.

A Appendices

Hyperparameter	Hyper-ELC	BERT
lr	0.0001	0.00001
gradient clip	1	1
pre-FF dropout	0.2	0.2
max sentence length	100	100
batch size	512	128
ES patience	3	3
ES monitor quantity	val mrr	val mrr
max pretrain epochs	20	—
hidden size	256	768
intermediate size	512	3072
# attention heads	4	12
# hidden layers	4	12
# encoder parameters	10M	109M
# decoder parameters WD50K	12M	37M
# decoder parameters WikiPeople	9M	27M

Table 5: Hyperparameters used for pre-training and finetuning models. During pre-training the model was trained with early stopping and a maximum number of epochs, but for finetuning only early stopping was used. Only learning rate (lr) was tuned. [0.00001, 0.0001, 0.001] were experimented with and the maximum learning rate that led to convergence was used. FF indicates the feed-foward layer and ES indicates early stopping.

MRR Ratio (StarE/Hyper-ELC)	Count	Relationship (head/tail)
0.23	1205	given name (t)
0.69	691	nominated for (h)
0.93	1038	educated at (h)
0.96	1175	member of sports team (h)
0.98	586	described by source (t)
0.99	1982	sex or gender (t)
0.99	542	family (t)
0.99	691	nominated for (t)
1.01	1688	country of citizenship (t)
1.02	1075	languages spoken, written or signed (t)
1.05	1019	place of birth (t)
1.05	596	work location (t)
1.05	606	position held (t)
1.08	695	father (t)
1.08	883	place of death (t)
1.09	606	position held (h)
1.1	1205	given name (h)
1.11	6657	sibling (t)
1.15	3892	occupation (t)
1.17	1038	educated at (t)
1.17	1492	bmember of (t)
1.17	6657	sibling (h)
1.18	4018	award received (t)
1.19	875	child (t)
1.19	875	child (h)
1.19	695	father (h)
1.24	4018	award received (h)
1.44	542	family (h)
1.47	3892	occupation (h)
1.47	883	place of death (h)
1.47	1019	place of birth (h)
1.5	1492	member of (h)
1.58	586	described by source (h)
1.63	1175	member of sports team (t)
2.09	1075	languages spoken, written or signed (h)
2.18	1688	country of citizenship (h)
2.23	596	work location (h)
6.77	1982	sex or gender (h)

Table 6: MRR ratio between Hyper-Elc and StarE for relationship head/tail prediction combinations on WikiPeople. Limited to the relationship head/tail pairs that occur more than 500 times in the evaluation set.

ID	Train Count	Original Name	Clean Name
P805	9204	statement is the subject of	is the subject of
P1686	9204	for work	for their work on
P1411	5590	nominated for	was nominated for the
P1346	4867	winner	winner was
P530	3515	diplomatic relation	diplomatic relations with
P166	3432	award received	received the award of
P2453	3011	nominee	the nominee was
P3831	1856	object has role	had the role of
P459	1755	determination method	which was determined by
P518	999	applies to part	for the part of
P453	989	character role	played the character
P17	879	country	in the country of
P2293	859	genetic association	is genetically associated with
P6942	736	animator	(movie) animator
P3092	682	film crew member	(movie) film crew member
P161	537	cast member	(movie) cast member
P750	477	distributed by	is distributed by
P421	414	located in time zone	is located in the time zone
P725	409	voice actor	—
P1264	400	valid in period	during the period of
P366	297	use	used for
P2852	259	emergency telephone number	emergency telephone number is
P159	241	headquarters location	is located in
P1552	237	has quality	has the quality
P642	221	of	—
P131	204	located in the administrative territorial entity	in
P39	202	position held	held the position of
P69	200	educated at	was educated at
P812	199	academic major	with academic major
P156	187	followed by	is followed by
P5800	178	narrative role	had the narrative role of
P31	175	instance of	is an instance of
P1365	167	replaces	—
P674	166	characters	character
P155	166	follows	—
P1366	157	replaced by	was replaced by
P19	153	place of birth	place of birth is

Table 7: 37 of the roles/relationships that occur most frequently in the WD50K (100) train dataset were considered and some were altered to make templates more similar to natural language - for example improving grammar.