Review: Inductive Entity Representations from Text via Link Prediction

Seminar Paper

by

Vinzenz Zinecker

Degree Course: Industrial Engineering and Management M.Sc.
Matriculation Number: 2067805

Institute of Applied Informatics and Formal Description Methods (AIFB)

KIT Department of Economics and Management

Advisor:	Prof. Dr. Harald Sack
Supervisor:	M.Sc. Genet Asefa Gesese
Submitted:	March 18, 2022

Abstract

Knowledge Graphs are commonly used to represent structured knowledge in the form of entities and relations between these. As knowledge graphs are often incomplete, there are many approaches to automatically generate missing relations. This Link Prediction task is challenging, especially in the inductive scenario, where new, unseen, entities are considered. Daza et al. [4] present Bert for Link Prediction (BLP) to tackle inductive link prediction, leveraging information contained in the textual descriptions of entities using a BERT-based encoder architecture. This seminar paper reviews their approach after giving an introduction and related literature on the topic. Then the BLP approach is expanded upon with the goal of reducing computational complexity by using DistilBERT [17] instead of BERT. Further, it was analysed how much of the performance can be attributed to the fine-tuning of the BERT-Layer. Results indicate that computation times can be significantly reduced without sacrificing the good performance. Code, implementation details and logs can be found at `https://github.com/vinzenzzinecker98/blp_reproduction`.

Contents

1 Introduction

Knowledge Graphs (KG) are being used more and more for representing structural knowledge in the form of entities and relations between the entities. However, they are often incomplete, meaning there are relations missing from them, especially when new entities are added to the graph.

Therefore, many methods have been developed to tackle the automated completion of missing links, called Link Prediction (LP). There are multiple scenarios for Link Prediction. In the transductive scenario, new links in an existing knowledge graph are predicted, with all entities seen at train time. In the semi-inductive scenario, the head or tail entity is new (unseen), while in the inductive scenario the relation for two unseen entities is predicted. To predict new links, relations can be modeled as operations on a vector space, which makes this process machine-understandable. This is enabled by generating embeddings for entities and relations, as relations can then be modeled as operations. These generated embeddings can also be useful for many other tasks, as they enable many operations on the knowledge graph, for example distance calculation between entities etc. Many existing KGs provide descriptive data, such as textual descriptions, for each entity. Using Natural Language Processing (NLP) methods, these descriptions can aide with the generation of useful embeddings. This is especially helpful when new entities are considered, as for those no existing relations can be exploited to find new links, however the textual description should be available for new entities by default.

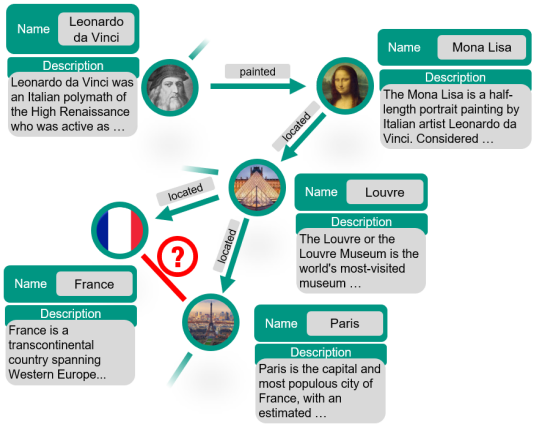

Figure 1: Knowledge Graph with textual descriptions and missing relation

Daza et al. [4] propose a system which uses the textual descriptions of entities to train an encoder with an inductive link prediction objective. It is also evaluated, how the embeddings produced by the encoder can be transferred to other tasks, showing good generalizability of the embeddings.

2 Related Work

This section will give an overview on scientific literature related to the topic.

2.1 Link Prediction

For Link Prediction based on Vector Representations multiple relational models are to be considered. A Knowledge Graph with Entities E and Relations R can be viewed as a Tensor with Dimension $D = |E| \times |E| \times |R|$. Using Tensor Factorization [14] or other methods, the missing links can be predicted using only the existing relations. Relevant for modeling relations is the so-called relational model, where different approaches exist.

Translational Models such as TransE [2] project both relations and entities in the same vector space and, given a valid triple with head h, relation r, and tail t, assume $\vec{h} + \vec{r} \approx \vec{t}$.

One weakness of the translational approach is the modeling of symmetric relation, for example "married to". There are however different ways to model relations, such as DistMult [25], where a multiplicative metric is used instead of the translational distance. DistMult itself is a special form of the RESCAL model [14]. To model asymmetric relations better, some approaches distinguish for the embeddings of entities whether they are the tail or the head of the relation, to this end, CompleX [21] is using complex numbers, while SimpleE [8] uses two separate embeddings, depending whether the entity is at the head or the tail of the relation.

2.2 Inductive Link Prediction

All approaches listed above perform good in the transductive scenario, where all entities have been observed at training time. For the inductive scenario however, these do not work. To generate embeddings for unseen entities, an encoder f_θ with parameters θ can be trained, which can then encode even unseen entities. There are many approaches applying advanced ML Methods on the entity attributes [3, 18, 7, 9], however they operate on a fixed domain, so that the encoder could not be transferred to another knowledge graph. Better generalizability is observed from approaches that rely on the textual descriptions: The Open World KG-Completion from Shi et al. [19] or KG-Bert [27] show good results, however they also use relations as input for the entity embedding, which makes the transfer to another knowledge graph without adjustments impossible. Further, new entities without existing relations pose a challenge. There is also the fully inductive approach KEPLER [22], where a BERT encoder is optimized for a joint objective: Link prediction and language modeling. This shows good results, but also high computation times due to the language modeling side-objective. In DKRL [23], a Convolutional Neural Network (CNN) is employed to generate embeddings from the textual descriptions. Daza et al. [4] used KEPLER and DKRL as baseline models to compare the performance of the proposed approach "BERT for Link Prediction" (BLP).

2.3 Text Embeddings

The generation of vector representations for text in general is an active field of research. Having a vector representation of text can be extremely useful for tasks such as text classification, sentiment analysis, search engines etc. Some approaches simply compare word counts or word frequencies (e.g., TF-IDF) of documents. By now there are more advanced methods for generating general-domain word embeddings, and the high computational effort to generate meaningful embeddings from large text corpora has led to the adoption of a transfer learning approach, where pretrained embeddings or models are retrained or reused for specific tasks. There are many pre-trained models for word embeddings, mostly using Bag of Words (word order is not considered) or Skip-gram (learning co-occurrences), some examples are GloVe [15] or Word2Vec [12]. A promising approach that, at the time, outperformed earlier models is Google's BERT [6], where word-piece embeddings were combined with a large transformer-based model and words and context were learned simultaneously. Since then, many models building upon BERT have been presented, such as Facebook's RoBERTa [11], XLNet [26], DistilBERT [17], which focuses on saving computing resources, or SentenceBERT [16], which is optimized for embedding whole sentences.

3 BERT for Link Prediction

3.1 Architecture

To achieve inductive Link Prediction, Daza et al. [4] propose the model "BERT for Link Prediction (BLP)", which uses textual descriptions of entities to generate meaningful embeddings in a link prediction scenario. To train the model, the link prediction task is formulated as a classification task, tuned to discriminate between true triples, that are part of the knowledge graph, and false triples (generated by replacing head or tail). This is achieved by calculating a score for such pairs of triples and calculating the loss based on the difference of scores. The model is thereby optimized for assigning higher scores to valid triples (where the triple is in the graph) than to the generated false triples. The calculation of scores is done according to figure 2: For the head and tail entity, the encoder f_θ with parameters θ encodes the description of the entities to generate the embeddings e_h, e_t. The embeddings for the relations have the same dimension as those and are randomly initialized at the start and looked up at train time.

The final score for this triple is then calculated according to the chosen scoring function, for which four variants relating to different relational models are considered (see table 1).

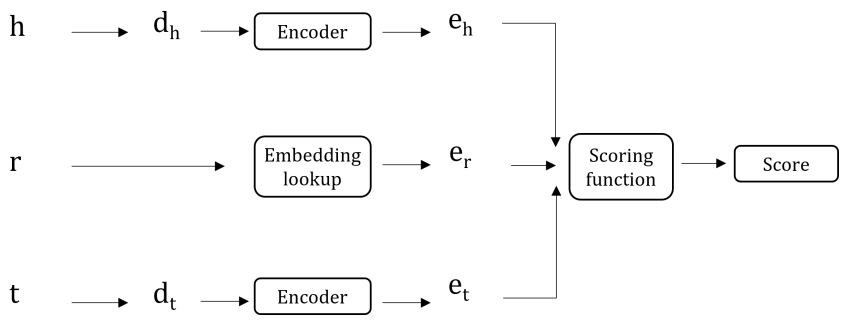

Figure 2: Score calculation for Link Prediction

Name	Scoring Function	Principle	Origin
TransE	$\|e_h + e_r - e_t\|_p$	Translational Distance	[2]
DistMult	$\langle e_h, e_r, e_t \rangle$	Multiplicative Distance	[25]
ComplEx	$Re(\langle e_h, e_r, \bar{e}_t \rangle)$	Complex Embeddings	[21]
SimplE	$\frac{1}{2}(\langle e_{h1}, e_{r1}, \bar{e}_{t2} \rangle + \langle e_{h1}, e_{r2}, \bar{e}_{t3} \rangle)$	Separate Embeddings when Head/Tail	[8]

Table 1: Scoring functions considered

The crucial part in this architecture is the encoder f_θ (see figure 3) used to encode the textual descriptions into embedding vectors. To this end, a pre-trained BERT model is trained on the link-prediction task to generate contextualized embeddings for each token. Before, extra tokens at the start ($[CLS]$) and end ($[SEP]$) are added. For the representation of the whole description, only the embedding $h_{[CLS]} \in \mathbb{R}^h$ (with $h =$ hidden size of BERT model) of the $[CLS]$ token is considered. Next, a fully connected linear layer reduces the dimension of the output. The weights of this layer, $W \in \mathbb{R}^{d \times h}$, is also a parameter to be trained. Here, d is the output size of the embeddings, which is set to 128, while BERTs hidden size h is 768.

The BERT encoding of the $[CLS]$ token is usually to be used on a sentence-level context representation (the training task is the prediction of the next sentence) [6]. As it is here used to encode the whole description, this discrepancy may explain why using more than the first 32 tokens brought no performance increases.

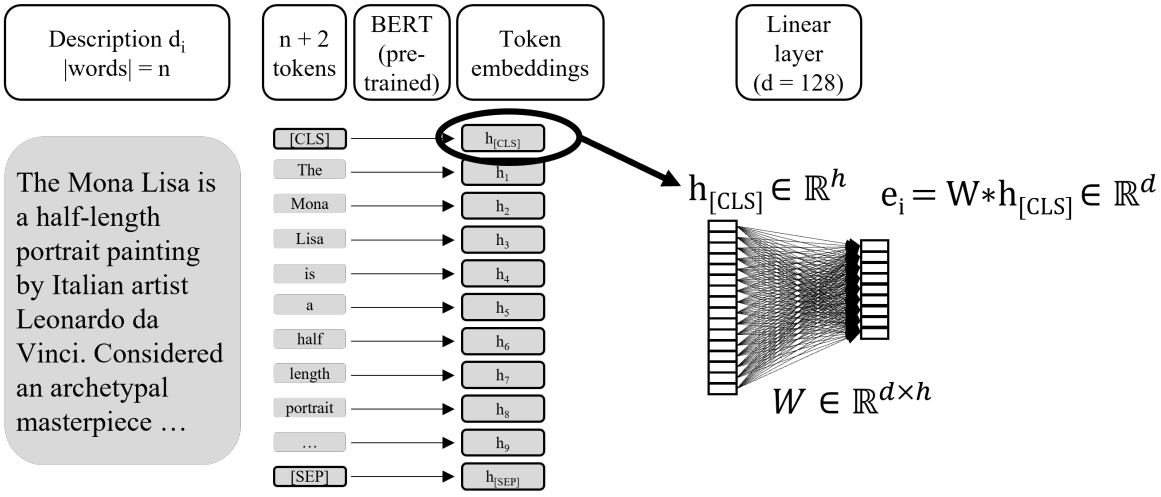

Figure 3: Encoder architecture: BERT for Link Prediction

3.2 Evaluation

3.2.1 Datasets

The datasets used for validating the model are FB15k-237 [20], WN18RR [5], and Wikidata5M [22]. FB15k-237, which is derived from the discontinued Freebase knowledge graph, is mainly composed of facts about sports, films, personalities. WN18RR is based on WordNet [1] and contains words and their relations such as hyponym, hypernym, synonym etc. Finally, the Wikidata5M KG, which was also used by KEPLER [22], is a subset of Wikidata.

	WN18RR	FB15k-237	Wikidata5M
Relations	11	237	822
Descriptions	Word definitions	Wikipedia introduction	Wikipedia introduction
		Training	
Entities	32,755	11,633	4,579,609
Triples	69,585	215,082	20,496,514
		Validation	
Entities	4,094	1,454	7,374
Triples	11,381	42,164	6,699
		Testing	
Entities	4,094	1,454	7,475
Triples	12,087	52,870	6,894

Table 2: Dataset statistics and datasplits used for evaluation of BLP

[1] `https://wordnet.princeton.edu/`

3.2.2 Link Prediction

For the inductive link prediction, two distinct evaluation scenarios are considered:

- **Transfer Evaluation**
  In this case, at test time, head and tail entity are not seen in the training, and incorrect candidates are drawn from a pool disjoint from the training entities. This scenario is used for the Wikidata5M dataset.

- **Dynamic evaluation**
  With dynamic evaluation, either only the head or tail, or both entities are new, and incorrect entities are drawn from all entities (train and test). So while some entities are known from the training, the pool of incorrect candidates is larger at test time. This is applied for WN18RR and FB15k-237.

Reported metrics are Mean Reciprocal Rank (MRR) and Hits@k for k = {1, 5, 10}. The "filtered" metrics are considered, meaning the randomly generated pool of incorrect triples is filtered to exclude triples that actually are in the knowledge graph. For the link prediction task, the baselines used are KEPLER [22], DKRL [23] with GloVe embeddings, DKRL [23] with BERT embeddings, GloVe-BOW (baseline from DKRL paper [23]), and BERT-BOW (Baseline from DKRL paper [23]). For the datasets FB15k-237, WN18RR and Wikidata5M, the proposed model managed to outperform the baselines on all metrics, the authors' detailed results can be seen in the appendix in tables 8 and 9. The results indicate a larger gap to the baseline performance for the WN18RR dataset, which the authors attribute to more "subtle" semantics in the word definition texts, while keywords are sufficient for the Freebase description which were obtained from Wikipedia introduction texts. It is also relevant to consider that the descriptions in WN18RR are shorter than the Wikipedia introductions used for the other datasets, indicating that the model works better with shorter texts. Looking at the relational models 1, TransE outperforms the more complex models most of the time, which is unexpected as complex relation types, such as asymmetry, cannot be modeled. The authors contribute this to better data-efficiency of the simpler TransE model, as for the much larger Wikidata5M dataset the complex relational models performed better. Another explanation could be a potential overfitting of the complex models. For the transfer evaluation performed on Wikidata5M, results are generally better compared to the other datasets. This can be attributed to the way larger size of the KG, as seen in table 2. Further, this results from the different evaluation setting, as the dynamic evaluation seems to be more challenging when compared to the transfer evaluation. A direct comparison of the evaluation modes on the same dataset would have been interesting in order to evaluate the effect of the dataset size versus the effect of the evaluation setting, which is not possible from the results presented. The best performing baseline, KEPLER, has results only for Wikidata5M, as this

is the only model that is also based on a BERT encoder (apart from the BoW model), it would have been interesting to see results on the other datasets. The comparatively worse performance of DKRL also shows how the BERT embeddings manage to outperform the CNN-based approach, which also does remove stopwords, which could lead to the loss of certain semantics (especially on the WN18RR dataset, where the difference is more pronounced).

3.2.3 Transfer tasks

To test generalizability, the tasks "Entity Classification" and "Information Retrieval" were evaluated using the embeddings trained for Link Prediction without further modification (no fine-tuning of the model). For the classification tasks, entity type was used as target variable, and the embeddings generated by the BLP Model were used as inputs to train a logistic regression classifier on this task. The promising results show that the entities type is encoded in the latent representation generated by the encoder. Furthermore, an information retrieval (IR) task is devised: A very common IR algorithm (BM25F-CA) is employed to generate a list of responses to a natural language query, assigning scores z_{IR} to entities of the knowledge graph. Then the encoder, which is only trained for the link prediction objective, is used to compute the similarity, using the simple dot product: $z_{BLP} = f_\theta(q)^\intercal f_\theta(d_e)$ between the query and the entity texts. This similarity score is then used to re-rank the results, using a linear combination: $z = \alpha z_{BLP} + (\alpha - 1) z_{IR}$. Optimization showed that the optimal α is between 0.1 and 0.7, depending on the query type. Here, results using only z_{BLP} would have been interesting. Also, using other distance metrics, such as cosine similarity could be an idea to be expanded upon.

4 Experiments

4.1 Models

The presented methodology will be expanded by looking at some variations of the used model. First, the more efficient DistilBERT [17] will be used to replace BERT in the encoder architecture, which significantly reduces training times (Model 1: DistilBERT). Further, Model 2 (Linear-SBERT) is devised in which the encoder is using SentenceBERT embeddings, which are pretrained for sentence classification. The embeddings are then passed through one linear layer, as before. However, here, only the linear layer is fine tuned with the link prediction objective, while the embeddings are not fine-tuned. To examine the influence of the type of pre-trained word embeddings, Wiki2Vec embeddings are applied in a BOW-setting (Model 3: "Wiki2vec-BOW").

No.	Model	Description
1	DistiBERT	BERT replaced by DistilBERT
2	Linear-SBERT	Only linear layer, and pretrained SBert embeddings
3	Wiki2vec-BOW	Wikipedia2Vec Embeddings with BOW-Model

Table 3: Modified Models: Overview

4.1.1 DistilBERT

DistilBERT was first presented in 2019 by Sanh et al. [17], claiming DistilBERT is significantly smaller and faster compared to BERT, while retaining most of BERT's performance. In order to achieve faster computations for the task at hand, the author's architecture was used, with DistilBERT replacing BERT. Concretely, the model "distilbert-base-uncased"[2] from the transformers library was employed.

4.1.2 Linear-SBERT

For this model, the embeddings of the entity descriptions were obtained from Sentence-BERT [16] and then passed through one linear layer to reduce dimensionality. The SentenceBERT architecture was however not fine-tuned, instead, only the linear layer was trained. This allows for faster training while still using the SBERT embeddings. Concretely, the model "bert-base-nli-mean-tokens" from the sentence-transformers library [3] was used. As here, only the linear layer is trained, hyperparameters were tuned using a grid-search, varying loss function margin, negative log-likelihood and learning rate 1e-5, 2e-5, 5e-5. As margin outperformed the nll-loss, and the biggest learning rate performed best, learning rates $\in$ 1e-4, 2e-4, 5e-4, 1e-3, 1e-2 were explored, giving (margin-based loss, learning rate = 1e-3) as best hyperparameters. Results for the grid-search can be inspected in appendix A.4.

4.1.3 Wikipedia2Vec

Using the same BOW-architecture as the paper did using the GloVe embeddings, entities were encoded using the average of the word embeddings in the description. Unknown tokens - [UNK] - were encoded using the average embedding over all tokens in the vocabulary. This approach is expected to produce worse results compared to DistilBERT or Linear-SBERT, as word order is not considered and the Wikipedia2Vec embeddings are expected to be less informative when compared to the BERT or SBERT models, as these use Word Piece embeddings, that consider the context of words as opposed to the

[2]DistilBERT model: `https://huggingface.co/distilbert-base-uncased`
[3]sentence-transformers: `https://huggingface.co/sentence-transformers`

Bag-of-Word approach here. This model can however be compared to the GloVe-Model, which uses the same architecture, but different embeddings.

As the vocabulary of the Wikipedia2Vec Embeddings is exceptionally large ($>$ 4 Million), only the top 400.000 words were considered to avoid memory issues. This is also the size of the vocabulary of the GloVe Embeddings used by the authors. To test the influence of this cut-off the results were compared to the results using a vocabulary of 800.000, showing that increasing the word count increases performance only by a small amount (See appendix A.3).

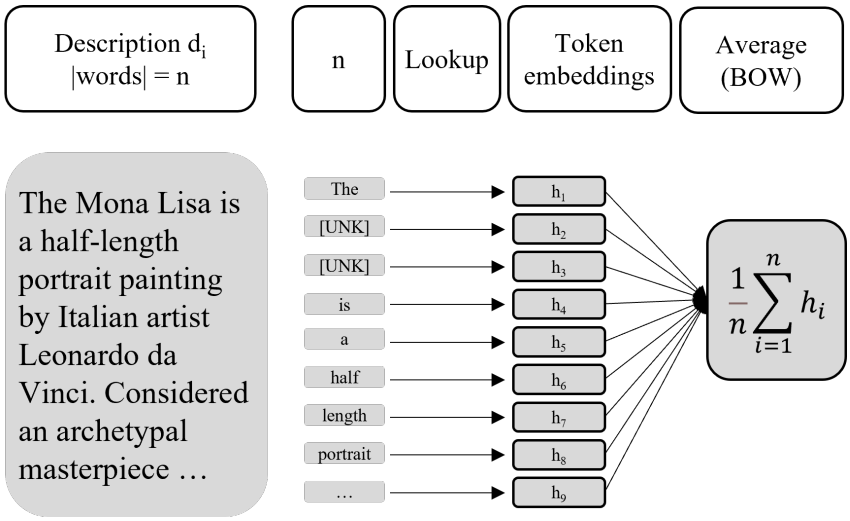

Figure 4: Entity encoding with BOW model

4.2 Results

In this section, results for the link prediction task using the architectures described in 4.1 will be discussed.

4.2.1 DistilBERT

For DistilBERT, the results were very similar to the original BLP results. Table 4 reports the results relative to the respective results of the original paper by calculating $\frac{MRR_{DistilBERT}}{MRR_{BLP}} * 100\%$. For some metrics, the DistilBERT model even outperformed the original implementation for some configurations. This could be explained by an overfitting of the complex BLP model, as this higher performance is observed especially for the more complex relational models. These results show the efficiency of the DistilBERT models, as the results differ only slightly, while computation time was significantly reduced by almost 50% compared to the original BERT-based model (see chapter 4.4). All results can be seen in the appendix in table 7.

	WN18RR	FB15k-237	Wikidata5M
TransE	97.47 %	99.85 %	100.40 %
DistMult	105.44 %	113.08 %	
ComplEx	99.31 %	112.91 %	
SimplE	94.06 %	113.4 %	

Table 4: Filtered MRR results for the DistilBERT model, relative to the respective results reported by Daza et al. [4] for BLP

4.2.2 Linear-SBERT

Results for the Linear-SBERT Model, seen in table 13, show worse results when compared to DistilBERT: for the FB15k-237 dataset, the filtered MRR is in average 28% worse, for WN18RR a decline of around 59% can be observed. This is expected, as less parameters are trained. Interestingly, the even worse performance on the WN18RR dataset seems to confirm the hypothesis that the WordNet entity descriptions contain more subtle semantics, which is better captured by fine-tuning a BERT-based encoder, while the FreeBase descriptions (Wikipedia introductions) are more easily modeled, even without fine-tuning the BERT-layer. There is also a difference regarding the relational models, as the simpler TransE performs better on the FB15k-237 dataset, while the more complex scoring functions seem to be better suited to the WN18RR dataset. This was observed with the other models, too. However, here this difference is more pronounced. This is also an indication of the nature of the relations in the datasets, as it is expected that complex, asymmetric or hierarchical relations are captured better by complex relational models.

4.2.3 Wikipedia2Vec

For the evaluation of Wikipedia2vec, all four different relational models were tested. Although a translational model makes more sense intuitively, implying the difference between the averaged word vectors of two descriptions can be modeled as their relation. This holds true, as the TransE model outperforms the other across all metrics, as can be seen in table 12. Comparing the results to the GloVe embeddings, only very small differences can be observed, which makes sense, as the GloVe embeddings used by Daza et al. [4, 15] as well as the Wikipedia2vec embeddings [24] were trained on the Wikipedia corpus.

4.3 Transfer task

To further test the embeddings' viability, the node classification task was performed on both datasets - the embeddings trained for Link Prediction (without further fine-tuning) were used as inputs for a logistic regression classifier, predicting entity types. For

WN18RR, Part of Speech (PoS) is classified, while for FB15k-237, the 50 most common entity types are predicted. The results can be seen in table 5. For balanced accuracy, the classes are weighted with their inverse prevalence, assigning more importance to minority classes. As these results are very similar to the results reported by Daza et al. [4], it can be assumed, that the embeddings generated with DistilBERT are comparable in quality to the ones produced using BERT.

	WN18RR		FB15k-237	
	Acc.	Bal. Acc.	Acc.	Bal. Acc.
TransE	98.9	74.9	85.4	44.4
DistMult	99.1	72.2	84.2	45.7
ComplEx	99.2	72.8	85.2	41.4
SimplE	99.2	82.2	69.2	44.6

Table 5: Results for the entity classification task, using the embeddings from the DistilBERT models. Reporting accuracy in % (Acc.) and balanced accuracy in % (Bal. Acc.).

4.4 Computational Resources

All experiments were conducted on bwUniCluster [4], using NVIDIA Tesla V100 GPUs. Table 6 compares runtimes of the models presented in this paper, compared to the original BERT for Link Prediction model, which was executed without changes on the same infrastructure. This especially shows the significant reduction of computing time achieved by using DistilBERT, while retaining performance (see chapter 4.2).

Model	Runtime [h]
BLP	09:39:51
DistilBERT	04:41:12
Wiki2Vec	01:02:18
LinearSBERT	02:56:59

Table 6: Average runtimes of training for FB15k-237 dataset

5 Future Work

These results should be expanded upon, especially regarding the understanding of underlying mechanics. Further experiments could for example integrate unseen relations into the model, as for now this is not possible (relations are initialized randomly at the start and cannot be expanded upon). One idea could be an encoder for relation descriptions, however, most current datasets do not provide descriptions for relations. One way

[4]Homepage: `https://wiki.bwhpc.de/e/Category:BwUniCluster_2.0`

of leveraging additional information about relations are hyper-relational facts: This approach focuses on triple-based knowledge graphs. There are however knowledge graphs that incorporate additional information for relations, so-called qualifiers, represented by key-value pairs. For example, the triple (Leonardo da Vinci, painted, Mona Lisa) could be enriched using (medium, oil paint), with "medium" being a characteristic of "painted". There are already approaches that use this additional knowledge in hyper-relational knowledge graphs in an inductive LP scenario [1], which shows some potential. What the results also show is how different relational models perform differently across datasets, implying there are inherent differences in how relations are to be modeled for each knowledge graph (domain). It could be of interest to devise methods expanding on this, e.g., to find the best fitting relational models for each or across knowledge graphs. Another idea could be testing the incorporation of more and different relational models, such as HolE [13] or TransR [10].

6 Conclusion

BERT for Link Prediction (BLP) is a strong tool for inductive link prediction on knowledge graphs. Considering multiple ways to model relations (scoring functions) and multiple datasets, it has been shown to be able to outperform legacy methods. To prove that the learned embeddings contain semantic information of the entities, transfer tasks were designed transferring the embeddings without further fine-tuning. This paper gives a review on the BLP architecture and then expands on it by showing how computation times can be significantly reduced by using DistilBERT, while retaining performance, also regarding the transfer tasks. Further it was illuminated how freezing the BERT layer and only training one linear layer still produces acceptable results, quantifying the amount of accuracy attributed to the fine-tuning of the BERT-based layer.

References

[1] Mehdi Ali, Max Berrendorf, Mikhail Galkin, Veronika Thost, Tengfei Ma, Volker Tresp, and Jens Lehmann. Improving Inductive Link Prediction Using Hyper-Relational Facts. 7 2021. URL: http://arxiv.org/abs/2107.04894.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. *Advances in Neural Information Processing Systems*, 26, 2013. URL: https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf.

[3] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep Neural Networks for Learning Graph Representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), 2 2016. URL: https://ojs.aaai.org/index.php/AAAI/article/view/10179.

[4] Daniel Daza, Michael Cochez, and Paul Groth. Inductive entity representations from text via link prediction. *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021*, pages 798–808, 4 2021. doi:10.1145/3442381.3450141.

[5] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2D Knowledge Graph Embeddings. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 1811–1818, 7 2017. URL: https://arxiv.org/abs/1707.01476v6, doi:10.48550/arxiv.1707.01476.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova Google, and A I Language. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Human Language Technologies, Volume 1*, pages 4171–4186. Association for Computational Linguistics, 6 2019. URL: http://dx.doi.org/10.18653/v1/N19-1423, doi:10.18653/v1/N19-1423.

[7] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. *Advances in Neural Information Processing Systems*, 2017-December:1025–1035, 6 2017. URL: https://arxiv.org/abs/1706.02216v4.

[8] Seyed Mehran Kazemi and David Poole. SimplE Embedding for Link Prediction in Knowledge Graphs. *Advances in Neural Information Processing Systems*, 31, 2018. URL: https://arxiv.org/abs/1802.04868.

[9] Thomas N. Kipf and Max Welling. Variational Graph Auto-Encoders. 11 2016. URL: https://arxiv.org/abs/1611.07308v1.

[10] Yankai Lin, Zhiyuan Liu, Huanbo Luan, Maosong Sun, Siwei Rao, and Song Liu. Modeling Relation Paths for Representation Learning of Knowledge Bases. 2015. URL: https://arxiv.org/pdf/1506.00379.pdf.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, and Paul G Allen. RoBERTa: A Robustly Optimized BERT Pretraining Approach. 2019. URL: https://arxiv.org/abs/1907.11692.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, 1 2013. URL: https://arxiv.org/abs/1301.3781v3.

[13] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic Embeddings of Knowledge Graphs. 10 2015. URL: http://arxiv.org/abs/1510.04935.

[14] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A Three-Way Model for Collective Learning on Multi-Relational Data. 2011. URL: https://icml.cc/2011/papers/438_icmlpaper.pdf.

[15] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1532–1543, 2014. doi:10.3115/V1/D14-1162.

[16] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992, 2020. doi:10.18653/V1/D19-1410.

[17] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 10 2019. URL: http://arxiv.org/abs/1910.01108.

[18] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling Relational Data with Graph Convolutional Networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10843 LNCS:593–607, 2018. doi:10.1007/978-3-319-93417-4{_}38.

[19] Baoxu Shi and Tim Weninger. Open-World Knowledge Graph Completion. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 1957–1964, 11 2017. URL: https://arxiv.org/abs/1711.03438v1.

[20] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. pages 57–66, 12 2015. doi:10.18653/V1/W15-4007.

[21] Théo Trouillon, Johannes Welbl, Sebastian Riedel, UKÉric Gaussier, and Guillaume Bouchard. Complex Embeddings for Simple Link Prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2071–2080, 2016. URL: https://proceedings.mlr.press/v48/trouillon16.html.

[22] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2 2021. URL: http://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00360/1923927/tacl_a_00360.pdf.

[23] Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation Learning of Knowledge Graphs with Entity Descriptions. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2659–2665. AAAI Press, 2016. URL: https://dl.acm.org/doi/10.5555/3016100.3016273, doi:10.5555/3016100.3016273.

[24] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia. pages 23–30, 12 2018. URL: https://arxiv.org/abs/1812.06280v4, doi:10.48550/arxiv.1812.06280.

[25] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 12 2014. URL: https://arxiv.org/abs/1412.6575v4.

[26] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Advances in Neural Information Processing Systems*, 32, 6 2019. URL: https://arxiv.org/abs/1906.08237v2, doi:10.48550/arxiv.1906.08237.

[27] Liang Yao, Chengsheng Mao, and Yuan Luo. KG-BERT: BERT for Knowledge Graph Completion. 2019. URL: https://arxiv.org/abs/1909.03193.

A Appendix

A.1 Results: DistilBERT compared to BLP

Model	Dataset	Rel. Model	MRR	hits@1	hits@3	hits@10
DistilBERT	FB15k-237	TransE	**0.1947**	**0.1134**	**0.2138**	0.3576
DistilBERT	FB15k-237	DistMult	0.1651	0.0931	0.1763	0.3111
DistilBERT	FB15k-237	ComplEx	0.1671	0.0977	0.1778	0.3089
DistilBERT	FB15k-237	SimplE	0.1633	0.0927	0.1740	0.3068
BLP	FB15k-237	TransE	**0.195**	**0.113**	0.213	**0.363**
BLP	FB15k-237	DistMult	0.146	0.076	0.156	0.286
BLP	FB15k-237	ComplEx	0.148	0.081	0.15	0.283
BLP	FB15k-237	SimplE	0.144	0.077	0.152	0.274
DistilBERT	WN18RR	TransE	0.2778	0.1220	0.3603	**0.5855**
DistilBERT	WN18RR	DistMult	0.2615	0.1424	0.3070	0.5017
DistilBERT	WN18RR	ComplEx	0.2592	0.1538	0.2921	0.4767
DistilBERT	WN18RR	SimplE	0.2248	0.1308	0.2476	0.4229
BLP	WN18RR	TransE	**0.285**	0.135	**0.361**	0.580
BLP	WN18RR	DistMult	0.248	0.135	0.288	0.481
BLP	WN18RR	ComplEx	0.261	**0.156**	0.297	0.472
BLP	WN18RR	SimplE	0.239	0.144	0.265	0.435
DistilBERT	Wikidata5M	TransE	0.4799	0.2454	0.6622	0.8748
BLP	Wikidata5M	TransE	0.478	0.241	0.66	0.871

Table 7: Full comparison of filtered metrics. Results for BLP as reported by Daza et al. [4, table 3]

A.2 Results: Bert for Link Prediction

	WN18RR				FB15k-237			
Method	MRR	H@l	H@3	H@IO	MRR	H@1	H@3	H@10
GloVe-BOW	0.170	0.055	0.215	0.405	0.172	0.099	0.188	0.316
BE-BOW	0.180	0.045	0.244	0.450	0.173	0.103	0.184	0.316
GloVe-DKRL	0.115	0.031	0.141	0.282	0.112	0.062	0.111	0.211
BE-DKRL	0.139	0.048	0.169	0.320	0.144	0.084	0.151	0.263
KEPLER	-	-	-	-	-	-	-	-
BLP-TransE	**0.285**	0.135	**0.361**	**0.580**	**0.195**	**0.113**	**0.213**	**0.363**
BLP-DistMu1t	0.248	0.135	0.288	0.481	0.146	0.076	0.156	0.286
BLP-ComplEx	0.261	**0.156**	0.297	0.472	0.148	0.081	0.15	0.283
BLP-SimplE	0.239	0.144	0.265	0.435	0.144	0.077	0.152	0.274

Table 8: Results by Daza et al. [4] on FB15k-237 and WN18RR

	Wikidata5M			
Method	MRR	H@1	H@3	H@10
GloVe-BOW	0.343	0.092	0.531	0.756
BE-BOW	0.362	0.082	0.586	0.798
GloVe-DKRL	0.282	0.077	0.403	0.660
BE-DKRL	0.322	0.097	0.474	0.720
KEPLER	0.402	0.222	0.514	0.730
BLP-TransE	0.478	0.241	0.660	0.871
BLP-DistMu1t	0.472	0.242	0.646	0.869
BLP-ComplEx	0.489	0.262	**0.664**	**0.877**
BLP-SimplE	**0.493**	**0.289**	0.639	0.866

Table 9: Results by Daza et al. [4] on Wikidata5M

A.3 Influence of vocabulary size of Wikipedia2Vec embeddings

Vocabulary was cut off at 400,000 and 800,000, respectively, and evaluated on both datasets (With TransE as relational model). Filtered MRR was 1.198% better for WN18RR and 1.208% for Fb15k-237 using the larger vocabulary.

	MRR	hits@1	hits@3	hits@10
WN18RR |v| = 400k	0.1649	0.0413	0.2185	0.4089
WN18RR |v| = 800k	0.1669	0.0399	0.2228	0.4181
FB15k-237 |v| = 400k	0.1717	0.0991	0.1873	0.3162
FB15k-237 |v| = 800k	0.1738	0.1010	0.1892	0.3207

Table 10: Filtered metrics for different vocabulary sizes

A.4 Grid-Search for Linear-SentenceBERT

Grid search was used to determine best loss funtion and learning rate for the LinerSBERT model. Results can be seen in table 11. All results are for FB15k-237 dataset using TransE relational model.

learning rate	loss function	MRR	hits@1	hits@3	hits@10
1,00E-05	margin	0.1120	0.0607	0.1139	0.2088
2,00E-05	margin	0.1263	0.0705	0.1307	0.2319
5,00E-05	margin	0.1371	0.0787	0.1432	0.2478
1,00E-04	margin	0.1431	0.0853	0.1506	0.2525
2,00E-04	margin	0.1436	0.0839	0.1500	0.2575
5,00E-04	margin	0.1439	0.0850	0.1496	0.2584
1,00E-03	**margin**	**0.1455**	**0.0860**	**0.1523**	**0.2598**
1,00E-02	margin	0.1286	0.0722	0.1334	0.2345
1,00E-05	neg. log loss	0.0958	0.0633	0.0976	0.1521
2,00E-05	neg. log loss	0.0953	0.0622	0.0961	0.1532
5,00E-05	neg. log loss	0.0952	0.0627	0.0978	0.1513

Table 11: Filtered metrics of the grid-search, varying learning rate and loss function

A.5 Full results for Wiki2Vec model

Model	Dataset	Rel. Model	MRR	hits@1	hits@3	hits@10
Wiki2Vec-BOW	FB15k-237	TransE	0.1717	0.0991	0.1873	0.3162
Wiki2Vec-BOW	FB15k-237	SimplE	0.138	0.0786	0.1464	0.2515
Wiki2Vec-BOW	FB15k-237	DistMult	0.1386	0.0767	0.1482	0.2566
Wiki2Vec-BOW	FB15k-237	ComplEx	0.1444	0.0816	0.154	0.2687
Wiki2Vec-BOW	WN18RR	TransE	0.1649	0.0413	0.2185	0.4089
Wiki2Vec-BOW	WN18RR	DistMult	0.1424	0.0765	0.1529	0.2799
Wiki2Vec-BOW	WN18RR	ComplEx	0.1655	0.0421	0.2184	0.4125
Wiki2Vec-BOW	WN18RR	SimplE	0.1582	0.0624	0.1866	0.3582
GloVe-BOW	FB15k-237	TransE	0.172	0.099	0.188	0.316
GloVe-BOW	WN18RR	TransE	0.17	0.055	0.215	0.405

Table 12: Results for the wiki2vec-BOW model, compared to GloVe-BOW model. Results for GloVe-BOW as reported by Daza et al. [4, table 3]

A.6 Full results for Linear-SBERT model

Model	Dataset	Rel. Model	MRR	hits@1	hits@3	hits@10
Linear-SBERT	FB15k-237	TransE	0.1424	0.0827	0.1488	0.258
Linear-SBERT	FB15k-237	DistMult	0.1184	0.0623	0.1217	0.2264
Linear-SBERT	FB15k-237	ComplEx	0.1195	0.0631	0.1223	0.2296
Linear-SBERT	FB15k-237	SimplE	0.1162	0.0618	0.1175	0.2216
Linear-SBERT	WN18RR	TransE	0.0824	0.0061	0.1057	0.2247
Linear-SBERT	WN18RR	DistMult	0.109	0.0114	0.1448	0.2953
Linear-SBERT	WN18RR	ComplEx	0.1099	0.0217	0.1326	0.285
Linear-SBERT	WN18RR	SimplE	0.109	0.0334	0.1229	0.2605

Table 13: Results for the Linear-SBERT Model, filtered metrics.

Declaration of Authorship

I hereby declare that we have composed this paper by myself and without any assistance other than the sources given in the list of works cited. This paper has not been submitted in the past or is currently being submitted to any other examination institution. It has not been published. All direct quotes, as well as indirect quotes which in phrasing or original idea have been taken from a different text (written or otherwise), have been marked as such clearly and in every single instance under a precise specification of the source.

Karlsruhe, 18.03.2022 Vinzenz Zinecker