Entity Typing with Triples using Language Models

Aniqa Riaz[1], Sara Abdollahi[2][0000−0001−7752−146X], and Simon Gottschalk[2][0000−0003−2576−4640]

[1] Universität Bonn, Germany
s6anriaz@uni-bonn.de
[2] L3S Research Center, Leibniz Universität Hannover, Germany
{abdollahi,gottschalk}@L3S.de

Abstract. Entity Typing is the task of assigning a type to an entity in a knowledge graph. In this paper, we propose ETwT (Entity Typing with Triples), which leverages the triples of an entity, namely its label, description and the property labels used on it. We analyse which language models and classifiers are best suited to this input and compare ETwT's performance on coarse-grained and fine-grained entity typing. Our evaluation demonstrates that ETwT is able to predict coarse-grained entity types with an F_1 score of 0.994, outperforming three baselines.

1 Introduction

The availability of entity types in a knowledge graph (e.g., *Microsoft* is a *Company*)[3] is important for a series of tasks including question answering and named entity linking. However, type information is often not complete. For example, in the well-established cross-domain knowledge graph DBpedia [1], $2,447,977$ out of $6,266,949$ entities do not have a type in the DBpedia ontology, including persons like Leonard E. Barrett and buildings like Deel Castle[4]. Therefore, entity typing is an essential sub-task of knowledge graph completion, aiming at full coverage of entity types in a knowledge graph.

Triples in a knowledge graph provide rich information describing an entity which can be used to detect the entity's type. This information includes both textual information, namely the label and description of an entity, as well as relationships to other entities (e.g., `dbr:Berlin dbo:country dbo:Germany`). Following the intuition behind [7], we expect that the properties used in an entity's triples can hint at the entity type, as well as its textual information [2,4]. In this paper, we propose ETwT (Entity Typing with Triples) which exploits both types of information simultaneously to perform highly precise entity typing.

[3] This entity type assignment can be expressed as a triple in the DBpedia knowledge graph: `dbr:Microsoft rdf:type dbo:Company` .

[4] In the DBpedia dumps of December 2022, considering all entities that have a Wikipedia page ID but do not redirect or disambiguate.

To train a model that best deals with the given entity information, we perform an analysis of how different language models (BERT [6], XLNet [10] and GPT [8]) and classifiers (Fully Connected, Convolutional and Recurrent Neural Networks) perform on entity typing.

We compare ETwT's best configuration on the DBpedia630k [9] dataset. The results demonstrate that ETwT outperforms three state-of-the-art baselines for coarse-grained entity typing, reaching an F_1 score of 0.994 on average. Thus, we highlight the effectiveness of leveraging entity properties along with entity labels and descriptions as input to a language model.

2 Related Work

Biswas et al. [2] combine a language model with a character embedding model to encode the entity label and detect its type. Cat2Type [3] instead utilises a Wikipedia category graph as an input to a language model. GRAND [4] employs BERT and RDF2Vec-based graph walking strategies. KLMo [5] uses translational embeddings of whole triples as an input to an attention layer. In contrast, ETwT focuses on an entity's properties including its label and description.

3 Approach

Fig. 1 gives an overview of ETwT where, given an entity e (here, a node representing *Berlin*), a ranking of entity types is generated of which the top-ranked entity type is chosen (here, *Place*). To do so, we (i) first extract triples describing the entity, (ii) embed them using a language model, and (iii) train a classifier.

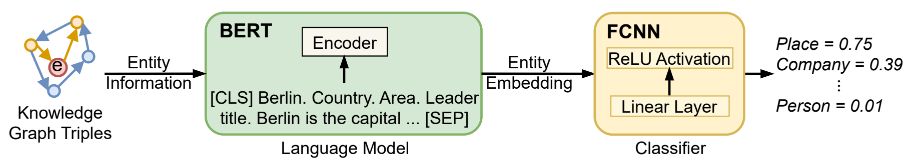

Fig. 1. The ETwT approach at the example of using BERT as language model and a fully connected neural network (FCNN) as classifier.

- **Knowledge Graph Triples**: We extract the property labels in the triples used on entity e plus its label and description as its entity information.
- **Language Model**: We fine-tune a pre-trained language model on the entity information. Fig. 1 exemplifies ETwT using BERT as the language model, where we use BERT's [CLS] token as input to the subsequent classifier.
- **Classification**: A multi-class classifier is trained to predict the type of an entity. Fig. 1 exemplifies ETwT using a fully connected neural network (FCNN) as classifier, where we use a ReLU activation for the final prediction.

4 Evaluation

4.1 Data

We use three splits (DB1, DB2 and DB3) of DBPedia630k [9][5] for evaluating our model. Each split is divided into a train, test, and validation set with a ratio of 50:30:20 [3]. We consider coarse-grained (14 entity types) and fine-grained (37 entity types[6]) entity typing.

4.2 Comparison with Baselines

We compare ETwT (using BERT and FCNN) against **JBN** (Judge an Entity by its Name) [2], **C2T** (Cat2Type) [3] and **GRAND** [4], as described in Section 2.

Table 1 shows that ETwT outperforms all baselines for coarse-grained entity typing based on Micro F_1 (MiF_1) and Macro F_1 (MaF_1) scores.[7] With F_1 scores of 0.994 on average, ETwT types entities nearly without a miss. For 37 fine-grained types, ETwT is outperformed by **GRAND** on DB1 but still performs best on DB3 with a Micro F_1 score of 0.947.

Table 1. Evaluation of coarse-grained (14 types) and fine-grained (37 types) entity typing. For ETwT, we use FCNN as classifier and BERT as language model.

	Coarse-grained						**Fine-grained**			
	DB1		**DB2**		**DB3**		**DB1**		**DB3**	
	MaF_1	**MiF_1**	**MaF_1**	**MiF_1**	**MaF_1**	**MiF_1**	**MaF_1**	**MiF_1**	**MaF_1**	**MiF_1**
JBN	0.714	0.720	0.606	0.657	0.446	0.511	0.231	0.521	0.318	0.531
C2T	0.983	0.984	0.983	0.983	0.985	0.985	0.402	0.732	0.847	0.915
GRAND	0.911	0.911	0.990	0.990	0.989	0.989	**0.745**	**0.870**	0.880	0.931
ETwT	**0.996**	**0.996**	**0.994**	**0.994**	**0.993**	**0.993**	0.404	0.765	**0.885**	**0.947**

4.3 Analysis of Language Models and Classifiers & Ablation Study

To identify which language models and classifiers are best to be used with ETwT, we evaluated all combinations of three language models (BERT [6], XLNet [10] and GPT [8]) and three classifiers: Fully Connected Neural Network (FCNN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). Table 2 shows the results of selected configurations and reveals that BERT with an FCNN performs best in most cases. Only for fine-grained entity types on DB3, XLNet performs best regarding Macro F_1.

[5] https://github.com/russabiswas/GRAND-Entity-Typing-in-KGs

[6] We omit DB2 for fine-grained entity typing as its selection of classes deviates from the other splits.

[7] Before train/test/validation split, the coarse-grained types are distributed evenly over the datasets. Therefore, MaF_1 and MiF_1 are similar.

Table 2. Analysis of classifiers (Class.) and language models (LM).

Class.	LM	Coarse-grained DB1		Coarse-grained DB2		Coarse-grained DB3		Fine-grained DB1		Fine-grained DB3	
		MaF$_1$	**MiF$_1$**	**MaF$_1$**	**MiF$_1$**	**MaF$_1$**	**MiF$_1$**	**MaF$_1$**	**MiF$_1$**	**MaF$_1$**	**MiF$_1$**
FCNN	XLNet	0.995	0.995	0.993	0.993	0.992	0.992	0.365	0.760	**0.896**	0.944
FCNN	GPT	0.989	0.989	0.988	0.988	0.987	0.987	0.368	0.753	0.868	0.937
FCNN	BERT	**0.996**	**0.996**	**0.994**	**0.994**	**0.993**	**0.993**	**0.404**	**0.765**	0.885	**0.947**
RNN	BERT	0.988	0.988	0.990	0.990	0.989	0.989	0.349	0.757	0.700	0.902
CNN	BERT	0.991	0.991	0.988	0.988	0.972	0.972	0.295	0.705	0.567	0.859

We further analyse the contribution of the inputs into the language model in an ablation study shown in Table 3. In the case of coarse-grained entity typing, the F_1 score drops from 0.994 to 0.959 (averaged over all splits) when removing the entity label and description and to 0.989 when removing the property labels. This indicates that both inputs are best used in combination.

Table 3. Ablation study for coarse-grained and fine-grained types.

Model	Coarse-grained DB1	Coarse-grained DB2	Coarse-grained DB3	Fine-grained DB1		Fine-grained DB3	
	MaF$_1$	**MaF$_1$**	**MaF$_1$**	**MaF$_1$**	**MiF$_1$**	**MaF$_1$**	**MiF$_1$**
ETwT	**0.996**	**0.994**	**0.993**	**0.404**	**0.765**	**0.885**	**0.947**
without description	0.972	0.953	0.951	0.358	0.758	0.817	0.898
without property	0.990	0.989	0.988	0.341	0.737	0.839	0.930

5 Conclusion

We introduced ETwT, an approach for predicting entity types in a knowledge graph using the entity label, description and property labels as input to a language model. ETwT outperforms state-of-the-art baselines and reaches average $F1$ scores of 0.994 for predicting coarse-grained entity types in DBpedia.

Acknowledgements This work was partially funded the Federal Ministry for Economic Affairs and Climate Action (BMWK), Germany ("ATTENTION!", 01MJ22012D).

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings. pp. 722–735. Springer (2007)
2. Biswas, R., et al.: Do Judge an Entity by its Name! Entity Typing using Language Models. In: Proceedings of the Extended Semantic Web Conference (2021)
3. Biswas, R., et al.: Wikipedia Category Embeddings for Entity Typing in Knowledge Graphs. In: Proceedings of the Knowledge Capture Conference (2021)
4. Biswas, R., et al.: Entity Type Prediction Leveraging Graph Walks and Entity Descriptions. In: Proceedings of the International Semantic Web Conference (2022)
5. He, L., et al.: KLMo: Knowledge Graph Enhanced Pretrained Language Model with Fine-Grained Relationships. In: Findings of EMNLP (2021)
6. Jacob, D., et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT (2019)
7. Paulheim, H., Bizer, C.: Type Inference on Noisy RDF Data. In: Proceedings of the International Semantic Web Conference (2013)
8. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners. OpenAI blog (2019)
9. Zhang, X., et al.: Character-level Convolutional Networks for Text Classification. Advances in Neural Information Processing Systems **28** (2015)
10. Zhilin, Y., et al.: XLNet: Generalized Autoregressive Pretraining for Language Understanding. Advances in Neural Information Processing Systems **32** (2019)