It's all in the Name: Entity Typing using Multilingual Language Models

Russa Biswas[1,2]✉, Yiyi Chen[1,2], Heiko Paulheim[3],
Harald Sack[1,2], and Mehwish Alam[1,2]

[1] FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany
[2] Karlsruhe Institute of Technology, Institute AIFB, Germany
firstname.lastname@fiz-karlsruhe.de,
[3] University of Mannheim, Germany
firstname@informatik.uni-mannheim.de

Abstract. The entity type information in Knowledge Graphs (KGs) of different languages plays an important role in a wide range of Natural Language Processing applications. However, the entity types in KGs are often incomplete. Multilingual entity typing is a non-trivial task if enough information is not available for the entities in a KG. In this work, multilingual neural language models are exploited to predict the type of an entity from only the name of the entity. The model has been successfully evaluated on multilingual datasets extracted from different language chapters in DBpedia namely German, French, Spanish, and Dutch.

Keywords: Entity Type Prediction · Knowledge Graph Completion · Multilingual Language Models · Classification

1 Introduction

Entity typing is the task of assigning entities in a Knowledge Graph (KG) with similar characteristic features to the same semantic type. The type information of entities plays a fundamental role in KG completion. In KGs such as DBpedia, YAGO, Wikidata, etc., the entity types are extracted automatically from structured data such as Wikipedia Infoboxes, generated using heuristics, or are human-curated. Therefore, the entity type information in KGs is often incomplete. Recent research focuses on automated entity type prediction models exploiting the triples in a KG using heuristics (Paulheim and Bizer, 2013) and neural network-based models (Biswas et al, 2020; Jin et al, 2019; Biswas et al, 2021b). The multi-level representations of entities are learned in MuLR (Yaghoobzadeh and Schütze, 2017) by using character embeddings, word embeddings, and entity embeddings using the Structured SKIP-gram (SSKIP) model followed by a multi-label classification approach to predict the entity types. The pre-trained RDF2Vec entity embeddings in (Sofronova et al, 2020) coupled with a supervised approach using a neural network based classifier and a vector similarity based unsupervised approach are used to predict the types of the entities in DBpedia.

Nevertheless, it is still a challenging problem to predict the type information for non-popular entities or new entities that are added to the KG, i.e., entities with less or no triples associated with them. The meaningfulness of the entity names in the Semantic Web has been studied in (de Rooij et al, 2016). To this end, entity names have been leveraged to predict the types of the entities using Neural Language Models (NLMs) in (Biswas et al, 2021a) for the English DBpedia chapter. However, to be able to predict the types of the entities just by their names, one has to understand multiple languages. Therefore, this originates the necessity of an automated multilingual entity type prediction framework for different chapters in DBpedia. For example, *Is it possible to predict the types of the entities dbr: Lachse, dbr: Saumon, dbr: Salmo, and dbr: Zalm from their names?* These are the names of *Salmon fish* in German, French, Spanish, and Dutch respectively. Therefore, this paper focuses on predicting the types of entities just by their names for different language chapters of DBpedia, namely German (DE), French (FR), Spanish (ES), and Dutch (NL).

This paper focuses on tackling two main challenges: (*i*) predict the types of the entities for which significantly less or no triples are available in the KGs, and (*ii*) predict the types of the entities in different languages. This lack of available information is compensated by exploiting the Multilingual Neural Language Models (Multilingual-NLMs), namely Wikipedia2Vec, and m-BERT. They are trained on a huge amount of textual data in multiple languages, and they provide implicit contextual information about the entities in their corresponding language-agnostic vector representations. The main contributions are:

- A multi-class classification framework is proposed to predict the missing entity types in multilingual DBpedia chapters exploiting the NLMs.
- A benchmark dataset for multilingual entity typing consisting of entities from German (DE), French (FR), Spanish (ES), and Dutch (NL) DBpedia chapters are published for re-usability purposes for future research.

2 Entity Typing using Language Models

This section discusses the Multilingual-NLMs and the classification model used for entity typing only from the names of the entities.
m-BERT. **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (Devlin et al, 2019) is a contextual embedding approach in which pretraining on bidirectional representations from the unlabeled text by using the left and the right context in all the layers is performed. Multilingual-BERT (m-BERT) supports 104 languages trained on text from the Wikipedia content with a shared vocabulary across all the languages. However, the size of Wikipedia varies greatly for different languages. The low-resource languages are underrepresented in the neural network model compared to the popular languages. The training on the low-resource languages of Wikipedia for a large number of epochs results in overfitting of the model. To combat the content imbalance of Wikipedia, less popular languages are over-sampled, whereas the popular languages are under-sampled. An exponential smoothing weighting of the data during the pre-training data

creation is used. For tokenization, a 110k shared WordPiece vocabulary is used. The word counts are weighted following the same method for the pre-training data creation. Therefore, the low-resource languages are up-weighted by some factors. Given an entity name $E_i = (w_1, w_2, ..., w_n)$, the input sequence to the m-BERT model is given by $([CLS], w_1, w_2, ..., w_n, [SEP])$, where E_i is the i^{th} entity and $w_1, w_2, .., w_n$ are the n words in the entity name. $[CLS]$ and $[SEP]$ are special tokens that mark the beginning and the end of the input sequence.

Wikipedia2vec. (Yamada et al, 2020) The model jointly learns word and entity embeddings from Wikipedia, where similar words and entities are close to one another in the vector space. It uses three submodels to learn the representation: Wikipedia Link Graph Model, Word-based skip-gram model, and Anchor context model. The skip-gram model forms the basis of these three submodels with a training objective to find embeddings useful for predicting context words or entities given a target word or entity. A Wikipedia Link Graph is generated in which the nodes are the entities in Wikipedia, and the edges are the links between them. There exists an edge between two nodes if the Wikipedia page of one entity has a link to that of the Wikipedia page of the other entity or if both the pages are linked to each other. Entity embeddings are learned from this Wikipedia Link Graph by predicting the neighboring entities following the skip-gram model. The second submodel is the Word-based skip-gram model, which learns word embeddings by predicting neighboring words given each word in a text contained on a Wikipedia page. Lastly, the Anchor context model learns the embeddings by predicting the neighboring words for each entity. This submodel focuses on putting similar words and the entities closer to each other in the vector space which helps in a deeper understanding of the interactions between the embeddings of the entities and the words in Wikipedia. In this work, pre-trained Wikipedia2vec models[4] for each of the languages, i.e., DE, FR, ES, and NL are used.

Embeddings of the Entity Names. In this work, the m-BERT base model has been used, in which each position outputs a vector of dimension equal to that of its hidden layer and its corresponding dimension is 768 for the base model. Each entity name is considered as a sentence for the input to m-BERT. The average of the last four hidden layers is taken to represent the entities. For Wikipedia2vec, the average of all word vectors in each entity name is taken as the vector representation of the entity.

Classification. Entity typing is considered a classification task with the types of entities as classes. A two-layered Fully Connected Neural Network (FCNN) model consisting of two dense layers with ReLU as an activation function has been used on the top of the entity vectors. This work considers non-overlapping classes. Therefore it is a multi-class classification problem. A softmax function used in the last layer calculates the probabilities of the entities in different classes.

[4] https://wikipedia2vec.github.io/wikipedia2vec/pretrained/

Table 1: Dataset Statistics

DBpedia chapters	Train	Test	Valid	Total Entities	#coarse-grained class	#fine-grained class
German	38500	23100	15400	77000	38	77
French	57999	34799	23199	115997	51	116
Spanish	42000	25200	16800	84000	45	84
Dutch	44000	26400	17600	88000	42	88

Table 2: Entity Typing Results on DE, FR, ES, and NL DBpedia Chapters

DBpedia chapters	#classes	m-BERT			Wikipedia2vec			m-BERT + Wikipedia2vec		
		Accuracy	Ma-F1	Mi-F1	Accuracy	Ma-F1	Mi-F1	Accuracy	Ma-F1	Mi-F1
German	38	0.818	0.760	0.818	0.870	0.817	0.870	**0.918**	**0.884**	**0.918**
	77	0.674	0.676	0.674	0.763	0.762	0.763	0.831	0.829	0.831
French	51	0.794	0.689	0.794	0.833	0.718	0.833	**0.867**	**0.780**	**0.867**
	116	0.544	0.542	0.544	0.611	0.612	0.611	0.678	0.680	0.678
Spanish	45	0.782	0.694	0.782	0.843	0.764	0.843	**0.894**	**0.846**	**0.894**
	84	0.629	0.627	0.629	0.681	0.682	0.681	0.788	0.788	0.788
Dutch	42	0.885	0.825	0.885	0.812	0.735	0.812	**0.908**	**0.859**	**0.908**
	88	0.664	0.665	0.664	0.753	0.757	0.753	0.825	0.825	0.825

Table 3: Comparison of our approach with SDType common entities

DBpedia Chapters	#Common Test Entities	SDType Accuracy	Wikipedia2vec	m-BERT
German	1103	43%	66.2%	61.4%
French	9223	75.4%	79.1%	75.3%
Spanish	3486	84.57%	85.3%	84%
Dutch	7977	73.09%	77.34%	73.13%

3 Evaluation

Datasets. The work focuses on predicting the types of the entities in different DBpedia chapters, namely, DE, FR, ES, and NL. The entities are extracted from the language versions of DBpedia-version 2016-10[5]. The most popular classes from each DBpedia chapter are chosen with 1000 entities per class. The coarse-grained classes are the parent classes of the fine-grained classes in the hierarchy tree. In this dataset, no entity belongs to two different classes in different hierarchy branches. Further details about the dataset are provided in Table 1 and are made available via Github[6].

Results. It is observed from the results as depicted in Table 2 that the static NLM Wikipedia2Vec trained on different languages of Wikipedia performs better than the m-BERT model for all the DBpedia chapters. BERT is a contextual embedding model that generates better latent representations where the context

[5] http://downloads.dbpedia.org/wiki-archive/downloads-2016-10.html

[6] https://github.com/russabiswas/MultilingualET_with_EntityNames

is available in the input sequence. The entity names are considered input sentences to the m-BERT model that do not provide any contextual information. On the other hand, the Wikipedia2Vec models trained on different Wikipedia languages perform better as they provide the fixed dense representation of the words or entities in the pre-trained models. It is noticeable that the concatenated vectors from m-BERT and Wikipedia2vec yield the best result as both the features are combined. Furthermore, Table 2 shows that the model performs better for coarse-grained classes compared to the fine-grained because it is often not possible to identify if a certain entity is of the type *Scientist* or an *Actor* from its name. However, it is possible to identify if the entity is of type *Person*. Also, the proposed model is compared with SDType in Table 3. For this, the publicly available results of the SDType method[7] are used. However, only a small fraction of the entities are common between the available results and DBpedia test datasets. The accuracy provided in Table 3 is calculated based on the number of common entities, and the proposed model with Wikipedia2Vec outperforms the SDType model.

4 Conclusion and Outlook

This paper analyzes multilingual NLMs for entity typing in a KG using entity names. In the future, fine-grained type prediction using other textual entity descriptions from the KG using the multilingual NLMs will be explored.

Bibliography

Biswas R, Sofronova R, Alam M, Sack H (2020) Entity Type Prediction in Knowledge Graphs using Embeddings. DL4KG @ ESWC2020

Biswas R, Sofronova R, Alam M, Heist N, Paulheim H, Sack H (2021a) Do Judge an Entity by Its Name! Entity Typing Using Language Models. ESWC P&D

Biswas R, Sofronova R, Sack H, Alam M (2021b) Cat2Type: Wikipedia Category Embeddings for Entity Typing in Knowledge Graphs. In: K-CAP

Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL-HLT)

Jin H, Hou L, Li J, Dong T (2019) Fine-Grained Entity Typing via Hierarchical Multi Graph Convolutional Networks. In: EMNLP-IJCNLP

Paulheim H, Bizer C (2013) Type Inference on Noisy RDF Data. In: ISWC

de Rooij S, Beek W, Bloem P, van Harmelen F, Schlobach S (2016) Are names meaningful? Quantifying social meaning on the semantic web. In: ISWC

Sofronova R, Biswas R, Alam M, Sack H (2020) Entity Typing Based on RDF2Vec Using Supervised and Unsupervised Methods. ESWC P&D

Yaghoobzadeh Y, Schütze H (2017) Multi-level Representations for Fine-Grained Typing of Knowledge Base Entities. In: EACL

Yamada I, Asai A, Sakuma J, Shindo H, Takeda H, Takefuji Y, Matsumoto Y (2020) Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia. In: EMNLP

[7] https://bit.ly/3eggWP0