An Attentive Fine-Grained Entity Typing Model with Latent Type Representation

Ying Lin and **Heng Ji**
Computer Science Department
University of Illinois at Urbana-Champaign, Urbana, IL, USA 61802
{yinglin8,hengji}@illinois.edu

Abstract

We propose a fine-grained entity typing model with a novel attention mechanism and a hybrid type classifier. We advance existing methods in two aspects: feature extraction and type prediction. To capture richer contextual information, we adopt contextualized word representations instead of fixed word embeddings used in previous work. In addition, we propose a two-step mention-aware attention mechanism to enable the model to focus on important words in mentions and contexts. We also present a hybrid classification method beyond binary relevance to exploit type interdependency with latent type representation. Instead of independently predicting each type, we predict a low-dimensional vector that encodes latent type features and reconstruct the type vector from this latent representation. Experiment results on multiple data sets show that our model significantly advances the state-of-the-art on fine-grained entity typing, obtaining up to 6.6% and 5.5% absolute gains in macro averaged F-score and micro averaged F-score respectively. [1]

1 Introduction

Fine-grained entity typing aims to assign one or more types to each entity mention given a certain context. For example, in the following sentence, "*If **Rogers** is in the game, the Huskies will be much better equipped to match the Cougars in that aspect*", the mention "Rogers" should be labeled as `athlete` in addition to `person` according to the context (e.g., game, Huskies). These fine-grained entity types are proven to be effective in supporting a wide range of downstream applications such as relation extraction (Yao et al., 2010), question answering (Lin et al., 2012), and coreference resolution (Recasens et al., 2013).

Fine-grained entity typing is usually formulated as a multi-label classification problem. Previous approaches (Ling and Weld, 2012; Choi et al., 2018; Xin et al., 2018) typically address it with binary relevance that decomposes the problem into isolated binary classification subproblems and independently predicts each type. However, this method is commonly criticized for its label independence assumption, which is not valid for fine-grained entity typing. For example, if the model is confident at predicting the type `artist`, it should promote its parent type `person` but discourage `organization` and its descendant types. In order to capture inter-dependencies between types, we propose a hybrid model that incorporates latent type representation in addition to binary relevance. Specifically, the model learns to predict a low-dimensional vector that encodes latent type features obtained through Principle Label Space Transformation (Tai and Lin, 2012) and reconstruct the sparse and high-dimensional type vector from this latent representation.

Another major challenge in fine-grained entity typing is to differentiate similar types, such as `director` and `actor`, which requires the model to capture slightly different nuances in texts. Previous neural models (Shimaoka et al., 2016; Xin et al., 2018; Choi et al., 2018; Xu and Barbosa, 2018) generally extract features from pre-trained word embeddings. Instead, we adopt contextualized word representations (Peters et al., 2018), which can capture context-aware word semantics and better represent out-of-vocabulary words. We further propose a two-step attention mechanism to actively extract the most relevant information from the sentence. Particularly, we calculate the attention for context words in a mention-aware manner, allowing the model to focus on different parts of the sentence for different mentions. For example, in the following sentence, "*In 2005 two fed-*

[1] Code for this paper is available at: https://github.com/limteng-rpi/fet.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 6197–6202,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

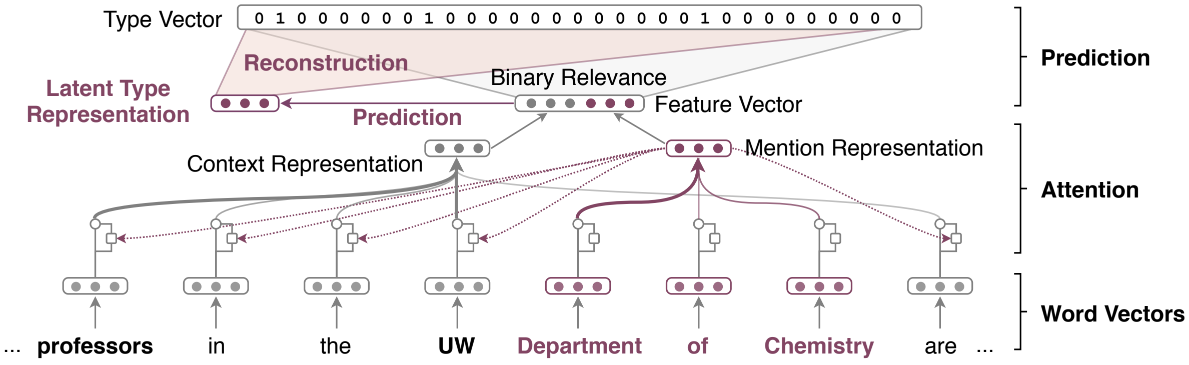


Figure 1: An illustration of our fine-grained entity typing framework.

eral agencies, the ***US Geological Survey*** *and the* ***Fish and Wildlife Service****, began to identify fish in the* ***Potomac*** *and tributaries* ...", the model should use "federal agencies" to help classify "US Geological Survey" and "Fish and Wildlife Service" as `government_agency`, but focus on "fish" and "tributaries" to determine that "Potomac" should be a `body_of_water` (river) instead of a `city`.

2 Methodology

Figure 1 illustrates our fine-grained entity typing framework. We represent the input sentence using pre-trained contextualized word representations. Next, we apply a two-step mention-aware attention mechanism to extract the most relevant features from the sentence to form the feature vector. On top of the model, we employ a hybrid classifier to predict the types of each mention.

2.1 Sentence Encoder

Contextual information plays a key role as we often need to determine the types especially subtypes according to the context. Hence, unlike previous neural models that generally use fixed word embeddings, we employ contextualized word representations (ELMo, Peters et al. 2018) that can capture word semantics in different contexts. Furthermore, because ELMo takes as input characters instead of words, it can better represent out-of-vocabulary words that are prevalent in entity mentions by leveraging sub-word information. Given a sentence of S words, the encoder generates a sequence of word vectors $\{\boldsymbol{r}_1, ..., \boldsymbol{r}_S\}$, where $\boldsymbol{r}_i \in \mathbb{R}^{d_r}$ is the representation of the i-th word.

2.2 Mention Representation

Previous attentive models (Shimaoka et al., 2017; Xu and Barbosa, 2018; Xin et al., 2018; Choi et al., 2018) only apply attention mechanisms to the context. However, some words in an entity mention may provide more useful information for typing, such as "Department" in Figure 1. To allow the model to focus on more informative words, we represent a mention m consisting of M words as a weighted sum of its contextualized word representations with an attention mechanism (Bahdanau et al., 2015) as

$$\boldsymbol{m} = \textstyle\sum_i^M a_i^m \boldsymbol{r}_i,$$

where the attention score a_i^m is computed as

$$a_i^m = \mathrm{Softmax}(e_i^m) = \frac{\exp\left(e_i^m\right)}{\sum_k^M \exp\left(e_k^m\right)},$$

$$e_i^m = \boldsymbol{v}^{m\top} \tanh\left(\boldsymbol{W}^m \boldsymbol{r}_i\right),$$

where parameters $\boldsymbol{W}^m \in \mathbb{R}^{d_a \times d_r}$ and $\boldsymbol{v}^m \in \mathbb{R}^{d_a}$ are learned during training, and the hidden attention dimension d_a is set to d_r in our experiments.

2.3 Context Representation

Given the context of mention m, we form its representation from involved contextualized word vectors with a mention-aware attention mechanism

$$\boldsymbol{c} = \textstyle\sum_i^C a_i^c \boldsymbol{r}_i,$$

where C is the number of contextual words, and a_i^c is defined as $a_i^c = \mathrm{Softmax}(e_i^c)$, where

$$e_i^c = \boldsymbol{v}^{c\top} \tanh(\boldsymbol{W}^c(\boldsymbol{r}_i \oplus \boldsymbol{m} \oplus p_i)),$$

where $\oplus$ represents concatenation, and $\boldsymbol{v}^c \in \mathbb{R}^{d_a}$ and $\boldsymbol{W}^c \in \mathbb{R}^{d_a \times (2d_r+1)}$ are trainable parameters. We introduce a relative position term p_i to indicate the distance from the i-th word to the mention as

$$p_i = \Big(1 - \mu\Big(\min(|i-a|, |i-b|) - 1\Big)\Big)^+,$$

where a and b are indices of the first and last words of the mention, and μ is set to 0.1.

Finally, the feature vector of mention m is formed by concatenating its mention representation $\boldsymbol{m}$ and context representation $\boldsymbol{c}$.

2.4 Hybrid Classification Model

We propose a hybrid type classification model consisting of two classifiers as Figure 1 shows. We first learn a matrix $\boldsymbol{W}^b \in \mathbb{R}^{d_t \times 2d_r}$ to predict type scores by

$$\tilde{\boldsymbol{y}}^b = \boldsymbol{W}^b(\boldsymbol{m} \oplus \boldsymbol{c}),$$

where $\tilde{y}_i^b$ is the score for the i-th type and d_t is the number of types. However, this method independently predicts each type and does not consider their inter-dependencies. To tackle this issue, we introduce an additional classifier inspired by Principle Label Space Transformation (Tai and Lin, 2012). Under the hypercube sparsity assumption that the number of training examples is much smaller than 2^{d_t}, Tai and Lin (2012) project high-dimensional type vectors into a low-dimensional space to find underlying type correlations behind the first order co-occurrence through Singular Value Decomposition (SVD)

$$\boldsymbol{Y} \approx \tilde{\boldsymbol{Y}} = \boldsymbol{U\Sigma L}^\top,$$

where $\boldsymbol{U} \in \mathbb{R}^{d_t \times d_l}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{d_l \times d_l}$, $\boldsymbol{L} \in \mathbb{R}^{N \times d_l}$, and $d_l \ll d_t$. This low-dimensional space is similar to the hidden concept space in Latent Semantic Analysis (Deerwester et al., 1990). The i-th row of $\boldsymbol{L}$ is the latent representation of the i-th type vector. After that, we learn to predict the latent type representation from the feature vector using

$$\boldsymbol{l} = \boldsymbol{V}^l(\boldsymbol{m} \oplus \boldsymbol{c}),$$

where $\boldsymbol{V}^l \in \mathbb{R}^{2d_r \times d_l}$ is trainable. We then reconstruct the type vector from $\boldsymbol{l}$ using a linear projection $\tilde{\boldsymbol{y}}^l = \boldsymbol{W}^l\boldsymbol{l} = \boldsymbol{U\Sigma l}$. Next, by combining scores from both classifiers, we have

$$\tilde{\boldsymbol{y}} = \sigma(\boldsymbol{W}^b(\boldsymbol{m} \oplus \boldsymbol{c}) + \gamma\boldsymbol{W}^l\boldsymbol{l}),$$

where γ is a scalar initialized to 0.1 and updated during training. Finally, our training objective is to minimize the following cross-entropy-based loss function

$$J(\theta) = -\frac{1}{N}\sum_i^N \boldsymbol{y}_i \log \tilde{\boldsymbol{y}}_i + (1 - \boldsymbol{y}_i)\log(1 - \tilde{\boldsymbol{y}}_i).$$

In the test phase, we predict each type with a probability $\tilde{y}_i > 0.5$ or $\arg\max \tilde{y}_i$ if all probabilities are lower than 0.5.

3 Experiments

3.1 Data Sets

In our experiments, we evaluate the proposed model on the following data sets.

OntoNotes fine-grained entity typing data set is derived from the OntoNotes corpus (Weischedel et al., 2013) and annotated by Gillick et al. (2014) using a three-layer set of 87 types. We use the augmented data set created by Choi et al. (2018).

FIGER (Ling and Weld, 2012) contains 2.7 million automatically labeled training instances from Wikipedia and 434 manually annotated sentences from news reports. We use ground truth mentions in our experiments and sample 0.1M training instances as the development set.

KNET (Xin et al., 2018) is another data set derived from Wikipedia. It consists of an automatically annotated subset (WIKI-AUTO) and a manually annotated (WIKI-MAN) test set.

BBN Pronoun Coreference and Entity Type Corpus (BBN, Weischedel and Brunstein, 2005) annotates 2,311 Wall Street Journal articles in Treebank-2 (LDC95T7) with fine-grained entity types. We use the version processed by Ren et al. (2016a).

Data set	Train	Dev	Test	Label	Depth
OntoNotes	3.4M	2,202	8,963	87	3
FIGER	2.7M	-	563	113	2
KNET (Wiki-Auto)	1M	0.1M	0.1M	74	2
KNET (Wiki-Man)	-	-	100	74	2
BBN	32,739	-	6,430	56	2

Table 1: Data set statistics: Numbers of train/dev/test instances, label set size, max hierarchy depth.

3.2 Experimental Setup

We use the pre-trained `original-5.5b` ELMo model [2] and freeze its weights during training. We use an Adam optimizer with learning rate of 5e-5, L2 weight decay of 0.01, warmup rate of 0.1, and linear learning rate decay. We use a mini-batch size of 200. To reduce overfitting, we apply dropout (Srivastava et al., 2014) to the word representation, relative position term, and the final feature vector with probability 0.5, 0.2, and 0.2.

We evaluate the performance by strict accuracy (Acc), macro-average F-score (Macro F1), and micro-average F-score (Micro F1) (Ling and Weld, 2012).

[2] https://allennlp.org/elmo

3.3 Evaluation Results

We compare the performance of our model with state-of-the-art methods on OntoNotes, FIGER, and KNET in Table 2, 3, and 4.

Model	Acc	Macro F1	Micro F1
Shimaoka et al. (2016)	51.7	70.9	64.9
Ren et al. (2016b)	57.2	71.5	66.1
Choi et al. (2018)	59.5	76.8	71.8
Our Model	**63.8**	**82.9**	**77.3**

Table 2: Results on the OntoNotes test set. The first three methods use only KB-based supervision.

Model	Acc	Macro F1	Micro F1
Ling and Weld (2012)	53.2	69.9	69.3
Yogatama et al. (2015)	–	–	72.3
Shimaoka et al. (2017)	54.5	74.8	71.6
+ hand-crafted	59.7	79.0	75.4
Our Model	**62.9**	**83.0**	**79.8**

Table 3: Results on the FIGER (Gold) test set.

	Wiki-Auto		
Model	**Acc**	**Macro F1**	**Micro F1**
Shimaoka et al. (2016)	42.8	72.4	74.9
KNET-MA	41.6	72.7	75.7
KNET-KA*	45.5	73.6	76.2
KNET-KAD*	**47.2**	74.9	77.9
Our Model	45.8	**77.4**	**78.4**
	Wiki-Man		
Model	**Acc**	**Macro F1**	**Micro F1**
Shimaoka et al. (2016)	18.0	69.4	70.1
KNET-MA	26.0	71.2	72.1
KNET-KA*	23.0	71.1	71.7
KNET-KAD*	**34.0**	74.9	75.3
Our Model	29.0	**77.6**	**75.3**

Table 4: Results on KNET test sets. KNET-KA and KNET-KAD use additional entity embeddings.

Model	Acc	Macro F1	Micro F1
Ling and Weld (2012)	46.7	67.2	61.2
Yosef et al. (2012)	52.3	57.6	58.7
Ren et al. (2016a)	**67.0**	72.7	73.5
Our Model	55.9	**79.3**	**78.1**
Our Model*	55.4	76.1	75.7

Table 5: Results on the BBN test set. Our Model* is a variant of our model that uses Bert contextualized word representations.

We compare the outputs of both classifiers ($\tilde{\boldsymbol{y}}^b$ and $\tilde{\boldsymbol{y}}^l$). The reconstructed type vector $\tilde{\boldsymbol{y}}^l$ alone doesn't predict entity types accurately, while adding this classifier substantially improves the performance of the model.

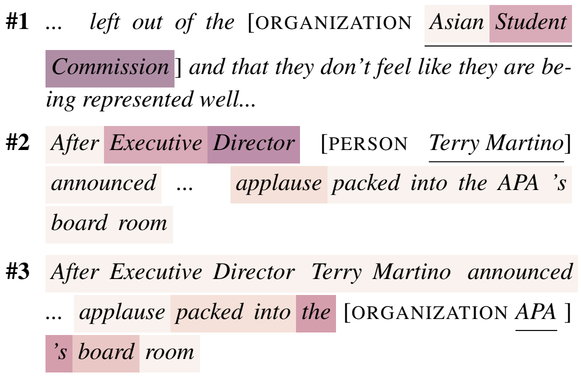
We show results on the BBN dataset in Table 5. Our Model* is a variant where we replace ELMo embeddings with Bert (large cased model) contextualized word representations.

	Acc	Marco F1	Micro F1
Shimaoka et al. (2017)	54.5	74.8	71.6
+ ELMo	59.3	81.7	79.0
+ latent	57.7	77.5	75.2
+ two-step attention	59.7	78.8	76.3
+ ELMo	62.9	83.0	79.8

Table 6: Ablation study on the FIGER (Gold) test set.

To evaluate the influence of individual components of our model, we conduct an ablation study as shown in Table 6. We implement a baseline model similar to (Shimaoka et al., 2017) (no hand-crafted features), the state-of-the-art on this data set. We observe that each component added to the model improves its performance.

We visualize mention and context attention in Figure 2. The first example shows the impact of mention attention. The baseline model mistakenly classifies the mention as `location` probably because "Asian" generally appears in location mentions, while our model successfully predicts `organization` by assigning higher weights to "Student Commission". In Example #2 and #3, we compare context attention between "Terry Martino" and "APA". Although they occur in the same sentence, our model is able to focus on different context words for different mentions with the mention-aware attention mechanism.



#1 ... *left out of the* [ORGANIZATION *Asian Student Commission*] *and that they don't feel like they are being represented well...*

#2 *After Executive Director* [PERSON *Terry Martino*] *announced* ... *applause packed into the APA 's board room*

#3 *After Executive Director Terry Martino announced* ... *applause packed into the* [ORGANIZATION *APA*] *'s board room*

Figure 2: Mention and context attention visualization.

4 Related Work

As entity types are usually organized as a forest of hierarchies, several models are proposed to leverage this structure. In (Yosef et al., 2012), the authors build a set of classifiers based on the taxonomy of YAGO (Hoffart et al., 2013) and perform top-down hierarchical classification. Shimaoka et al. (2017) propose a hierarchical label encoding method to share parameters between types in the same hierarchy. Xu and Barbosa (2018) propose a hierarchy-aware loss function to reduce the penalty when predicted types are related. By contrast, our model automatically find type interdependency via matrix factorization and is able to capture inter-dependencies between types regardless of whether they are in the same hierarchy.

Attention mechanisms are widely used in neural fine-grained entity typing models (Shimaoka et al., 2016, 2017; Xu and Barbosa, 2018; Xin et al., 2018; Choi et al., 2018) to weight context words. We also apply it to mention words and introduce a position term to make attentions for context words more mention-aware.

5 Conclusions and Future Work

We propose an attentive architecture for fine-grained entity typing with latent type representation. Experiments on multiple data sets demonstrate that our model achieves state-of-the-art performance. In the future, we will further improve the performance of fine-grained types, which is still lower than that of general types due to less training instances and distant supervision noise. We also plan to utilize fine-grained entity typing results in more downstream applications, such as coreference resolution and event extraction.

Acknowledgement

This work was supported by the U.S. DARPA AIDA Program No. FA8750-18-2-0014, Air Force No. FA8650-17-C-7715, and U.S. ARL NS-CTA No. W911NF-09-2-0053. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio, and Roee Aharoni. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2015)*.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61.

Thomas Lin, Oren Etzioni, et al. 2012. No noun phrase left behind: detecting and typing unlinkable entities. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP 2012)*.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI 2012)*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2018)*.

Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. 2013. The life and death of discourse entities: Identifying singleton mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013)*.

Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. 2016a. AFET: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*.

Xiang Ren, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, and Jiawei Han. 2016b. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD 2016)*.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2016. An attentive neural architecture for fine-grained entity type classification. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*.

Farbound Tai and Hsuan-Tien Lin. 2012. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542.

Ralph Weischedel and Ada Brunstein. 2005. Bbn pronoun coreference and entity type corpus. *BBN Technologies*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 LDC2013T19. *Linguistic Data Consortium, Philadelphia, PA*.

Ji Xin, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2018. Improving neural fine-grained entity typing with knowledge attention. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*.

Peng Xu and Denilson Barbosa. 2018. Neural fine-grained entity type classification with hierarchy-aware loss. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2018)*.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*.

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL 2015)*.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. Hyena: Hierarchical type classification for entity names. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*.