Fine-grained Entity Typing through Increased Discourse Context and Adaptive Classification Thresholds

Sheng Zhang
Johns Hopkins University
zsheng2@jhu.edu

Kevin Duh
Johns Hopkins University
kevinduh@cs.jhu.edu

Benjamin Van Durme
Johns Hopkins University
vandurme@cs.jhu.edu

Abstract

Fine-grained entity typing is the task of assigning fine-grained semantic types to entity mentions. We propose a neural architecture which learns a distributional semantic representation that leverages a greater amount of semantic context – both document and sentence level information – than prior work. We find that additional context improves performance, with further improvements gained by utilizing adaptive classification thresholds. Experiments show that our approach without reliance on hand-crafted features achieves the state-of-the-art results on three benchmark datasets.

1 Introduction

Named entity typing is the task of detecting the type (e.g., *person*, *location*, or *organization*) of a named entity in natural language text. Entity type information has shown to be useful in natural language tasks such as question answering (Lee et al., 2006), knowledge-base population (Carlson et al., 2010; Mitchell et al., 2015), and co-reference resolution (Recasens et al., 2013). Motivated by its application to downstream tasks, recent work on entity typing has moved beyond standard coarse types towards finer-grained semantic types with richer ontologies (Lee et al., 2006; Ling and Weld, 2012; Yosef et al., 2012; Gillick et al., 2014; Del Corro et al., 2015). Rather than assuming an entity can be uniquely categorized into a single type, the task has been approached as a multi-label classification problem: e.g., in "... *became a top seller ... Monopoly is played in 114 countries.* ..." (Figure 1), "*Monopoly*" is considered both a *game* as well as a *product*.

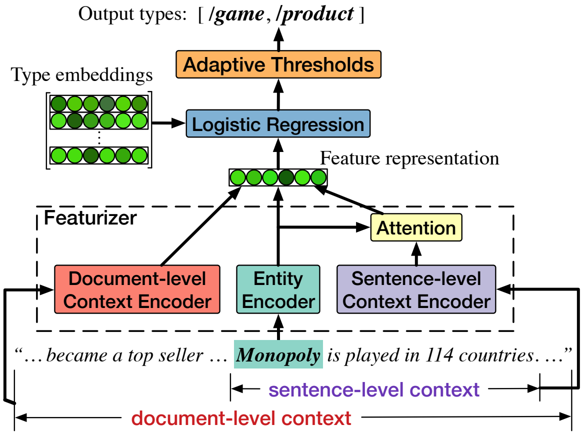

Figure 1: Neural architecture for predicting the types of entity mention "*Monopoly*" in the text "*... became a top seller ... Monopoly is played in 114 countries. ...*". Part of document-level context is omitted.

The state-of-the-art approach (Shimaoka et al., 2017) for fine-grained entity typing employs an attentive neural architecture to learn representations of the entity mention as well as its context. These representations are then combined with hand-crafted features (e.g., lexical and syntactic features), and fed into a linear classifier with a fixed threshold. While this approach outperforms previous approaches which only use sparse binary features (Ling and Weld, 2012; Gillick et al., 2014) or distributed representations (Yogatama et al., 2015), it has a few drawbacks: (1) the representations of left and right contexts are learnt independently, ignoring their mutual connection; (2) the attention on context is computed solely upon the context, considering no alignment to the entity; (3) document-level contexts which could be useful in classification are not exploited; and (4) hand-crafted features heavily rely on system or human annotations.

To overcome these drawbacks, we propose a neural architecture (Figure 1) which learns more context-aware representations by using a better attention mechanism and taking advantage of semantic discourse information available in both the document as well as sentence-level contexts. Fur-

*Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 173–179
New Orleans, June 5-6, 2018.

ther, we find that adaptive classification thresholds leads to further improvements. Experiments demonstrate that our approach, without any reliance on hand-crafted features, outperforms prior work on three benchmark datasets.

2 Model

Fine-grained entity typing is considered a multi-label classification problem: Each entity e in the text x is assigned a set of types T^* drawn from the fine-grained type set $\mathcal{T}$. The goal of this task is to predict, given entity e and its context x, the assignment of types to the entity. This assignment can be represented by a binary vector $y \in \{1, 0\}^{|\mathcal{T}|}$ where $|\mathcal{T}|$ is the size of $\mathcal{T}$. $y_t = 1$ iff the entity is assigned type $t \in \mathcal{T}$.

2.1 General Model

Given a type embedding vector w_t and a featurizer φ that takes entity e and its context x, we employ the logistic regression (as shown in Figure 1) to model the probability of e assigned t (i.e., $y_t = 1$)

$$P(y_t = 1) = \frac{1}{1 + \exp\left(-w_t^\intercal \varphi(e, x)\right)}, \tag{1}$$

and we seek to learn a type embedding matrix $W = [w_1, \ldots, w_{|\mathcal{T}|}]$ and a featurizer φ such that

$$T^* = \underset{T}{\operatorname{argmax}} \prod_{t \in T} P(y_t = 1) \cdot \prod_{t \notin T} P(y_t = 0). \tag{2}$$

At inference, the predicted type set $\hat{T}$ assigned to entity e is carried out by

$$\hat{T} = \left\{ t \in \mathcal{T} : P(y_t = 1) \geq r_t \right\}, \tag{3}$$

with r_t the threshold for predicting e has type t.

2.2 Featurizer

As shown in Figure 1, featurizer φ in our model contains three encoders which encode entity e and its context x into feature vectors, and we consider both *sentence-level* context x_s and *document-level* context x_d in contrast to prior work which only takes *sentence-level* context (Gillick et al., 2014; Shimaoka et al., 2017).[1]

The output of featurizer φ is the concatenation of these feature vectors:

$$\varphi(e, x) = \begin{bmatrix} f(e) \\ g_s(x_s, e) \\ g_d(x_d) \end{bmatrix}. \tag{4}$$

We define the computation of these feature vectors in the followings.

Entity Encoder: The entity encoder f computes the average of all the embeddings of tokens in entity e.

Sentence-level Context Encoder: The encoder g_s for sentence-level context x_s employs a single bi-directional RNN to encode x_s. Formally, let the tokens in x_s be $x_s^1, \ldots, x_s^n$. The hidden state h_i for token x_s^i is a concatenation of a left-to-right hidden state $\overrightarrow{h_i}$ and a right-to-left hidden state $\overleftarrow{h_i}$,

$$h_i = \begin{bmatrix} \overrightarrow{h}_i \\ \overleftarrow{h}_i \end{bmatrix} = \begin{bmatrix} \overrightarrow{f}(x_s^i, \overrightarrow{h}_{i-1}) \\ \overleftarrow{f}(x_s^i, \overleftarrow{h}_{i+1}) \end{bmatrix}, \tag{5}$$

where $\overrightarrow{f}$ and $\overleftarrow{f}$ are L-layer stacked LSTMs units (Hochreiter and Schmidhuber, 1997). This is different from Shimaoka et al. (2017) who use two separate bi-directional RNNs for context on each side of the entity mention.

Attention: The feature representation for x_s is a weighted sum of the hidden states: $g_s(x_s, e) = \sum_{i=1}^{n} a_i h_i$, where a_i is the attention to hidden state h_i. We employ the dot-product attention (Luong et al., 2015). It computes attention based on the alignment between the entity and its context:

$$a_i = \frac{\exp\left(h_i^\intercal W_a f(e)\right)}{\sum_{j=1}^{n} \exp\left(h_j^\intercal W_a f(e)\right)}, \tag{6}$$

where W_a is the weight matrix. The dot-product attention differs from the self attention (Shimaoka et al., 2017) which only considers the context.

Document-level Context Encoder: The encoder g_d for document-level context x_d is a multi-layer perceptron:

$$g_d(x_d) = \mathrm{relu}(W_{d_1} \tanh(W_{d_2} \mathrm{DM}(x_d))), \tag{7}$$

where DM is a pretrained distributed memory model (Le and Mikolov, 2014) which converts the document-level context into a distributed representation. W_{d_1} and W_{d_2} are weight matrices.

---

[1] Document-level context has also been exploited in Yaghoobzadeh and Schütze (2015); Yang et al. (2016); Karn et al. (2017); Gupta et al. (2017).

2.3 Adaptive Thresholds

In prior work, a fixed threshold ($r_t = 0.5$) is used for classification of all types (Ling and Weld, 2012; Shimaoka et al., 2017). We instead assign a different threshold to each type that is optimized to maximize the overall strict F_1 on the dev set. We show the definition of strict F_1 in Section 3.1.

3 Experiments

We conduct experiments on three publicly available datasets.[2] Table 1 shows the statistics of these datasets.

OntoNotes: Gillick et al. (2014) sampled sentences from OntoNotes (Weischedel et al., 2011) and annotated entities in these sentences using 89 types. We use the same train/dev/test splits in Shimaoka et al. (2017). Document-level contexts are retrieved from the original OntoNotes corpus.

BBN: Weischedel and Brunstein (2005) annotated entities in Wall Street Journal using 93 types. We use the train/test splits in Ren et al. (2016b) and randomly hold out 2,000 pairs for dev. Document contexts are retrieved from the original corpus.

FIGER: Ling and Weld (2012) sampled sentences from 780k Wikipedia articles and 434 news reports to form the train and test data respectively, and annotated entities using 113 types. The splits we use are the same in Shimaoka et al. (2017).

	Train	Dev	Test	Types
OntoNotes	251,039	2,202	8,963	89
BBN	84,078	2,000	13,766	93
FIGER	2,000,000	10,000	563	113

Table 1: Statistics of the datasets.

3.1 Metrics

We adopt the metrics used in Ling and Weld (2012) where results are evaluated via strict, loose macro, loose micro F_1 scores. For the i-th instance, let the predicted type set be $\hat{T}_i$, and the reference type set T_i. The precision (P) and recall (R) for each metric are computed as follow.

Strict:

$$P = R = \frac{1}{N}\sum_{i=1}^{N}\delta(\hat{T}_i = T_i)$$

Loose Macro:

$$P = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{T}_i \cap T_i|}{|\hat{T}_i|}$$

$$R = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{T}_i \cap T_i|}{|T_i|}$$

Loose Micro:

$$P = \frac{\sum_{i=1}^{N}|\hat{T}_i \cap T_i|}{\sum_{i=1}^{N}|\hat{T}_i|}$$

$$R = \frac{\sum_{i=1}^{N}|\hat{T}_i \cap T_i|}{\sum_{i=1}^{N}|T_i|}$$

3.2 Hyperparameters

We use open-source GloVe vectors (Pennington et al., 2014) trained on Common Crawl 840B with 300 dimensions to initialize word embeddings used in all encoders. All weight parameters are sampled from $\mathcal{U}(-0.01, 0.01)$. The encoder for sentence-level context is a 2-layer bidirectional RNN with 200 hidden units. The DM output size is 50. Sizes of W_a, W_{d_1} and W_{d_2} are 200×300, 70×50, and 50×70 respectively. Adam optimizer (Kingma and Ba, 2014) and mini-batch gradient is used for optimization. Batch size is 200. Dropout (rate=0.5) is applied to three feature functions. To avoid overfitting, we choose models which yield the best strict F_1 on dev sets.

3.3 Results

We compare experimental results of our approach with previous approaches[3], and study contribution of our base model architecture, document-level contexts and adaptive thresholds via ablation. To ensure our findings are reliable, we run each experiment twice and report the average performance.

Overall, our approach significantly increases the state-of-the-art macro F_1 on both OntoNotes and BBN datasets.

On OntoNotes (Table 3), our approach improves the state of the art across all three metrics. Note that (1) without adaptive thresholds or document-level contexts, our approach still outperforms other approaches on macro F_1 and micro F_1; (2) adding hand-crafted features (Shimaoka et al., 2017) does not improve the performance.

---

[2] We made the source code and data publicly available at `https://github.com/sheng-z/figet`.

[3] For PLE (Ren et al., 2016b), we were unable to replicate the performance benefits reported in their work, so we report the results after running their codebase.

ID	Sentence	Gold	Prediction
A	*... Canada's declining crude output, combined with ... will help intensify U.S. reliance on* **oil** *from overseas. ...*	/other	/other /other/health /other/health/treatment
B	*Bozell joins Backer Spielvogel Bates and* **Ogilvy Group** *as U.S. agencies with interests in Korean agencies.*	/organization /organization/company	/organization /organization/company

Table 2: Examples showing the improvement brought by document-level contexts and dot-product attention. Entities are shown in the green box. The gray boxes visualize attention weights (darkness) on context tokens.

Approach	Strict	Macro	Micro
Binary(Gillick et al., 2014)	N/A	N/A	70.01
Kwsabie(Yogatama et al., 2015)	N/A	N/A	72.98
PLE(Ren et al., 2016b)	51.61	67.39	62.38
Ma et al. (2016)	49.30	68.23	61.27
AFET(Ren et al., 2016a)	55.10	71.10	64.70
Fnet(Abhishek et al., 2017)	52.20	68.50	63.30
Neural(Shimaoka et al., 2017)	51.74	70.98	64.91
w/o Hand-crafted features	47.15	65.53	58.25
Our Approach	**55.52**	**73.33**	**67.61**
w/o Adaptive thresholds	53.49	73.11	66.78
w/o Document-level contexts	53.17	72.14	66.51
w/ Hand-crafted features	54.40	73.13	66.89

Table 3: Results on the OntoNotes dataset.

Approach	Strict	Macro	Micro
Kwsabie(Yogatama et al., 2015)	N/A	N/A	72.25
Attentive(Shimaoka et al., 2016)	58.97	77.96	74.94
Fnet(Abhishek et al., 2017)	65.80	81.20	77.40
Ling and Weld (2012)	52.30	69.90	69.30
PLE(Ren et al., 2016b)	49.44	68.75	64.54
Ma et al. (2016)	53.54	68.06	66.53
AFET(Ren et al., 2016a)	53.30	69.30	66.40
Neural(Shimaoka et al., 2017)	59.68	**78.97**	75.36
w/o Hand-crafted features	54.53	74.76	71.58
Our Approach	**60.23**	78.67	**75.52**
w/o Adaptive thresholds	60.05	78.50	75.39
w/ Hand-crafted features	60.11	78.54	75.33

Table 5: Results on the FIGER dataset.

This indicates the benefits of our proposed model architecture for learning fine-grained entity typing, which is discussed in detail in Section 3.4; and (3) Binary and Kwasibie were trained on a different dataset, so their results are not directly comparable.

Approach	Strict	Macro	Micro
PLE(Ren et al., 2016b)	49.44	68.75	64.54
Ma et al. (2016)	**70.43**	75.78	76.50
AFET(Ren et al., 2016a)	67.00	72.70	73.50
Fnet(Abhishek et al., 2017)	60.40	74.10	75.70
Our Approach	60.87	**77.75**	**76.94**
w/o Adaptive thresholds	58.47	75.84	75.03
w/o Document-level contexts	58.12	75.65	75.11

Table 4: Results on the BBN dataset.

On BBN (Table 4), while Ma et al. (2016)'s label embedding algorithm holds the best strict F_1, our approach notably improves both macro F_1 and micro F_1.[4] The performance drops to a competitive level with other approaches if adaptive thresholds or document-level contexts are removed.

On FIGER (Table 5) where no document-level context is currently available, our proposed approach still achieves the state-of-the-art strict and micro F_1. If compared with the ablation variant of the Neural approach, i.e., w/o hand-crafted features, our approach gains significant improvement. We notice that removing adaptive thresholds only causes a small performance drop; this is likely because the train and test splits of FIGER are from different sources, and adaptive thresholds are not generalized well enough to the test data. Kwasibie, Attentive and Fnet were trained on a different dataset, so their results are not directly comparable.

3.4 Analysis

Table 2 shows examples illustrating the benefits brought by our proposed approach. Example A illustrates that sentence-level context sometimes is not informative enough, and attention, though already placed on the head verbs, can be misleading. Including document-level context (i.e., "*Canada's declining crude output*" in this case) helps preclude wrong predictions (i.e., /other/health and /other/health/treatment). Example B shows that the semantic patterns learnt by our attention mechanism help make the correct prediction. As we observe in Table 3 and Table 5, adding hand-crafted features to our approach does not im-

---

[4] Integrating label embedding into our proposed approach is an avenue for future work.

prove the results. One possible explanation is that hand-crafted features are mostly about syntactic-head or topic information, and such information are already covered by our attention mechanism and document-level contexts as shown in Table 2. Compared to hand-crafted features that heavily rely on system or human annotations, attention mechanism requires significantly less supervision, and document-level or paragraph-level contexts are much easier to get.

Through experiments, we observe no improvement by encoding type hierarchical information (Shimaoka et al., 2017).[5] To explain this, we compute cosine similarity between each pair of fine-grained types based on the type embeddings learned by our model, i.e., w_t in Eq. (1). Table 6 shows several types and their closest types: these types do not always share coarse-grained types with their closest types, but they often co-occur in the same context.

Type	Closest Types
/other/event/accident	/location/transit/railway /location/transit/bridge
/person/artist/music	/organization/music /person/artist/director
/other/product/mobile_phone	/location/transit/railway /other/product/computer
/other/event/sports_event	/location/transit/railway /other/event
/other/product/car	/organization/transit /other/product

Table 6: Type similarity.

4 Conclusion

We propose a new approach for fine-grained entity typing. The contributions are: (1) we propose a neural architecture which learns a distributional semantic representation that leverage both document and sentence level information, (2) we find that context increased with document-level information improves performance, and (3) we utilize adaptive classification thresholds to further boost the performance. Experiments show our approach achieves new state-of-the-art results on three benchmarks.

Acknowledgments

This work was supported in part by the JHU Human Language Technology Center of Excellence (HLTCOE), and DARPA LORELEI. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA or the U.S. Government.

References

Abhishek Abhishek, Ashish Anand, and Amit Awekar. 2017. Fine-grained entity type classification by jointly learning representations and label embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 797–807, Valencia, Spain. Association for Computational Linguistics.

Andrew Carlson, Justin Betteridge, Richard C Wang, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 101–110. ACM.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. Finet: Context-aware fine-grained named entity typing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 868–878, Lisbon, Portugal. Association for Computational Linguistics.

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

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690, Copenhagen, Denmark. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Sanjeev Karn, Ulli Waltinger, and Hinrich Schütze. 2017. End-to-end trainable attentive decoder for hierarchical entity classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 752–758, Valencia, Spain. Association for Computational Linguistics.

---

[5] The type embedding matrix W for the logistic regression is replaced by the product of a learnt weight matrix V and the constant sparse binary matrix S which encodes type hierarchical information.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196.

Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. 2006. *Fine-Grained Named Entity Recognition Using Conditional Random Fields for Question Answering*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 94–100. AAAI Press.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Yukun Ma, Erik Cambria, and SA GAO. 2016. Label embedding for zero-shot fine-grained named entity typing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 171–180. The COLING 2016 Organizing Committee.

Tom M. Mitchell, William W. Cohen, Estevam R. Hruschka Jr., Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard C. Wang, Derry Tanti Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2302–2310.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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*, pages 627–633, Atlanta, Georgia. Association for Computational Linguistics.

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*, pages 1369–1378, Austin, Texas. Association for Computational Linguistics.

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*, KDD '16, pages 1825–1834, New York, NY, USA. ACM.

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*, pages 69–74, San Diego, CA. Association for Computational Linguistics.

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: Volume 1, Long Papers*, pages 1271–1280, Valencia, Spain. Association for Computational Linguistics.

Ralph Weischedel and Ada Brunstein. 2005. Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*, 112.

Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. Ontonotes: A large training corpus for enhanced processing. *Handbook of Natural Language Processing and Machine Translation. Springer*.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2015. Corpus-level fine-grained entity typing using contextual information. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 715–725, Lisbon, Portugal. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

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 (Volume 2:*

Short Papers), pages 291–296, Beijing, China. Association for Computational Linguistics.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. HYENA: Hierarchical type classification for entity names. In *Proceedings of COLING 2012: Posters*, pages 1361–1370, Mumbai, India. The COLING 2012 Organizing Committee.