End-to-End Trainable Attentive Decoder for Hierarchical Entity Classification

Sanjeev Kumar Karn[1,2], **Ulli Waltinger**[2] **and Hinrich Schütze**[1]

[1]LMU Munich
[2]Siemens Corporate Technology Munich
[1]sanjeev.karn@campus.lmu.de
[2]{sanjeev.kumar_karn,ulli.waltinger}@siemens.com

Abstract

We address fine-grained entity classification and propose a novel attention-based recurrent neural network (RNN) encoder-decoder that generates paths in the type hierarchy and can be trained end-to-end. We show that our model performs better on fine-grained entity classification than prior work that relies on flat or local classifiers that do not directly model hierarchical structure.

1 Introduction

Many tasks in natural language processing involve hierarchical classification, e.g., fine-grained morphological and part-of-speech tags form a hierarchy (Mueller et al., 2013) as do many large topic sets (Lewis et al., 2004). The task definition can either specify that a single path is correct, corresponding to a single-label classification problem at the lowest level of the hierarchy, e.g., in fine-grained morphological tagging; or that multiple paths can be correct, corresponding to a multilabel classification problem at the lowest level of the hierarchy, e.g., in topic classification.

In this paper, we address fine-grained entity mention classification, another problem with a hierarchical class structure. In this task, each mention can have several fine-grained types, e.g., Obama is both a politician and an author in a context in which his election is related to his prior success as a best-selling author; thus, the problem is multilabel at the lowest level of the hierarchy.

Two standard approaches to hierarchical classification are flat and local classification. In flat classification (e.g., FIGER (Ling and Weld, 2012), Attentive Encoder (Shimaoka et al., 2016; Shimaoka et al., 2017)), the task is formalized as a flat multiclass multilabel problem. In local classification (Gillick et al., 2014; Yosef et al., 2012; Yogatama et al., 2015), a separate local classifier is learned for each node of the hierarchy. In both approaches, some form of postprocessing is necessary to make the decisions consistent, e.g., an entity can only be a celebrity if they are also a person.

In this paper, we propose an attentive RNN encoder-decoder for hierarchical classification. The encoder-decoder performs classification by generating paths in the hierarchy from top node to leaf nodes. Thus, we model the structure of the hierarchy more directly than prior work. On each step of the path, part of the input to the encoder-decoder is an attention-weighted sum of the states of a bidirectional Gated Recurrent Unit (GRU) (Cho et al., 2014) run over the context of the mention to be classified. Unlike prior work on hierarchical entity classification, our architecture can be trained end-to-end. We show that our model performs better than prior work on the FIGER dataset (Ling and Weld, 2012).

This paper is structured as follows. In Section 2, we provide a detailed description of our model PthDCode. In Section 3, we describe and analyze our experiments. In Section 4, we discuss related work. Section 5 concludes.

2 Model

Figure 1 displays our model PthDCode.

We use lowercase italics for variables, uppercase italics for sequences, lowercase bold for vectors and uppercase bold for matrices. Sentence $S = \langle \mathbf{x}_1, \ldots, \mathbf{x}_{|S|} \rangle$ is a sequence of words, represented as embeddings $\mathbf{x}_i$, each of dimension d. The classes of an entity are represented as $\mathbf{y}$, a vector of l binary indicators, each indicating whether the corresponding class is correct. Hidden states of forward and backward encoders and of the decoder have dimensionality p.

PthDCode extracts mention $\langle \mathbf{x}_b, \ldots, \mathbf{x}_r \rangle$, right context $R_c = \langle \mathbf{x}_{r+1}, \ldots, \mathbf{x}_{r+w} \rangle$ and left context $L_c = \langle \mathbf{x}_{b-1}, \ldots, \mathbf{x}_{b-w} \rangle$ where w is a parameter.

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 752–758,
Valencia, Spain, April 3-7, 2017.

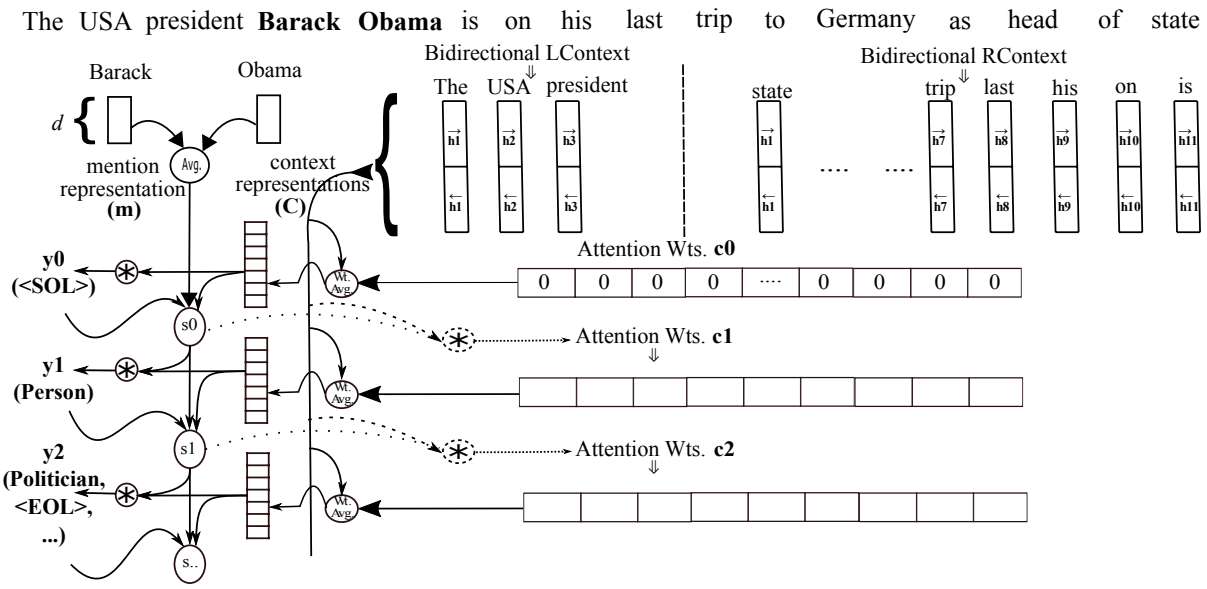

Figure 1: PthDCode, the attentive encoder-decoder for hierarchical entity classification

The representation $\mathbf{m}$ of the mention is computed as the average of its $r - b + 1$ vectors. The context is represented by $\mathbf{C}$, a matrix of size $2w \times 2p$; each column of $\mathbf{C}$ consists of two hidden state vectors $\mathbf{h}$ (each of dimension $2p$), corresponding to forward and backward GRUs run on L_c and R_c.

The initial state $\mathbf{s}_0$ of PthDCode's decoder RNN is computed using the mention representation $\mathbf{m}$ compressed to p dimensions by an extra hidden layer (not shown in the figure). Initial output $\mathbf{y}_0$ is a dummy symbol SOL (Start Of Label), and initial attention weights $\mathbf{c}_0$ are set to zero. At each path generation step i, attention weights α_{ij} are computed following Bahdanau et al. (2014):

$$\alpha_{ij} = \frac{\exp(\mathbf{e}_{ij})}{\sum_{j=1}^{2w} \exp(\mathbf{e}_{ij})} \quad (1)$$

$$\mathbf{e}_{ij} = \text{att}(\mathbf{s}_{i-1}, \mathbf{C}_{.j}) \quad (2)$$

where att is a feedforward network with softmax output layer and $\mathbf{C}_{.j}$ is the j^{th} column of $\mathbf{C}$. The final context representation for the decoder is then computed as $\mathbf{c}_i = \sum_{j=1}^{2w} \alpha_{ij} \mathbf{C}_{.j}$. In Figure 1, dashed objects are used for indicating involvement in calculating attention weights.

The attention-weighted sum $\mathbf{c}_i$ and the current state $\mathbf{s}_{i-1}$ are used to predict the distribution $\mathbf{y}_i$ over entity classes (non-dashed *-nodes in Figure 1):

$$\mathbf{y}_i = \text{g}\left(\mathbf{s}_{i-1}, \mathbf{c}_i\right) \quad (3)$$

where g is a feedforward network with element-wise sigmoid. Finally, PthDCode uses prediction $\mathbf{y}_i$, weighted average $\mathbf{c}_i$ and previous state $\mathbf{s}_{i-1}$ to compute the next state:

$$\mathbf{s}_i = \text{f}(\mathbf{s}_{i-1}, \mathbf{y}_i, \mathbf{c}_i) \quad (4)$$

The loss function at each step or level is binary cross-entropy:

$$\frac{1}{l} \sum_{k=1}^{l} -t_{ik} \log(y_{ik}) - (1 - t_{ik}) \log(1 - y_{ik}) \quad (5)$$

where $\mathbf{y}_i$ and $\mathbf{t}_i$ are prediction and truth and l the number of classes. The objective is to minimize the total loss, i.e., the sum of the losses at each level. During inference, we compute the Cartesian product of predicted types at each level and filter out those paths that do not occur in train.

3 Experiments and results

Dataset. We use the Wiki dataset (Ling and Weld, 2012) published by Ren et al. (2016).[1] It consists of 2.69 million mentions obtained from 1.5 million sentences sampled from Wikipedia articles. These mentions are tagged with 113 types with a maximum of two levels of hierarchy. Ling and Weld (2012) also created a test set of 434 sentences that contain 562 gold entity mentions. Similar to prior work (Ling and Weld, 2012; Ren et al., 2016; Yogatama et al., 2015; Shimaoka et al., 2017), we randomly sample a training set of 2 million and a disjoint dev set of size 500.

[1] https://drive.google.com/file/d/0B2ke42d0kYFfVC1fazdKYnVhYWs

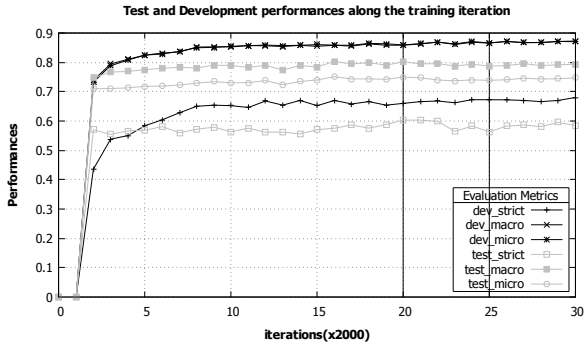

Figure 2: Learning curve

Evaluation. Like prior work, we use three F_1 metrics, strict, loose macro and loose micro, that differ in the definition of precision P and recall R. Let n be the number of mentions, T_i the true set of tags of mention i and Y_i the predicted set. Then, we define $P = R = 1/n \sum_{i=1}^{n} \delta(Y_i = T_i)$ for strict; $P = 1/n \sum_{i=1}^{n} (|Y_i \cap T_i|)/(|Y_i|)$ and $R = 1/n \sum_{i=1}^{n} (|Y_i \cap T_i|)/(|T_i|)$ for loose macro; and $P = (\sum_{i=1}^{n} |Y_i \cap T_i|)/(\sum_{i=1}^{n} |Y_i|)$ and $R = (\sum_{i=1}^{n} |Y_i \cap T_i|)/(\sum_{i=1}^{n} |T_i|))$ for loose micro.

Parameter Settings. We use pre-trained word embeddings of size 300 provided by (Pennington et al., 2014). OOV vectors are randomly initialized. Similar to (Shimaoka et al., 2017), all hidden states **h** of the encoder-decoder were set to 100 dimension and mention lengths m to 5. Window size is $w = 15$. We bracket left and right contexts with special start and end symbols. For short left / right contexts, we bracket with additional different start / end symbols that are masked out for calculation of loss and attention weights. Another special symbol EOL (End Of Label) is appended to short paths, so that all hierarchical paths have the same length. We use ADAM (Kingma and Ba, 2014) with learning rate .001 and batch size 500. Following (Srivastava et al., 2014), we regularize our learning by dropout of states used in computing prediction as in Eq. 3 with probability of .5. Similarly, we also drop out feedback connections used in computing next states as in Eq. 4 with probability of .2. We also add Gaussian noise with a probability of .1 to feedforward weights. The weights of feedforward units are initialized with an isotropic Gaussian distribution having mean 0 and standard deviation .02 while weights of recurrent units are initialized with random orthogonal matrix.

Results. As shown in Figure 2, we evaluate our model on dev and test sets after every 2k iterations and report the performances of the models that are stable in all form of metrics on dev set. The reason for evaluating on range of models is nature of collection of dev and test data. We use $c_v = \sigma/\mu$, the coefficient of variation (Brown, 1998), to select and combine models in application. After an initial training stage, we compute c_v for each of the three metrics for windows of 10,000 iterations, startpoints have the form $4000 + 6000s$. For a given window starting at iteration $2000t$, we compute c_v of the three metrics based on the six iterations $2000(t + i), 0 \leq i \leq 5$. We select the range with the lowest average c_v; this was the interval $[40000, 50000]$; cf. Figure 2. Since train and test data are collected from different sources, the sensitive strict measure varies with a larger standard deviation compared to other metrics.

Table 1 shows performance of PthDCode on test, based on the interval $[40000, 50000]$; average and standard deviation are computed for $2000(20 + i), 0 \leq i \leq 5$, as described above. PthDCode achieves clearly better results than other baseline methods – FIGER (Ling and Weld, 2012), (Yogatama et al., 2015) and (Shimaoka et al., 2017) – when trained on raw (i.e., not denoised) datasets of a similar size. Attentive encoder (Shimaoka et al., 2017) is a neural baseline for PthDCode, to which comparison in Table 1 suggests decoding of path hierarchy rather than flat classification significantly improves the performance. Ren et al. (2016) implementation of FIGER (Ling and Weld, 2012) trained on the denoised corpus performs better on strict and loose micro metrics, but as the training data are different, results are not directly comparable. An important observation in Table 1 is that most of the improved systems (Ren et al., 2016; Yogatama et al., 2015) consider entity classification in a hierarchical setup either through denoising or classification. One can also observe that our model achieves relatively high increase in terms of loose macro. The reason for this is mostly because of the macro F_1 direct dependence on average precision and average recall, which in our case is relatively high because of large improvement in the recall.

Table 2 shows that for level-wise comparisons on loose micro F_1, PthDCode improves recall compared to Yogatama et al. (2015)'s precision oriented system. We attribute this increase in recall and F_1 to the fact that PthDCode at each step collects feedback from the preceding level and is

	strict	macro F_1	micro F_1
FIGER, L&W	.532	.699	.693
Yogatama et al.	–	–	.723
Shimaoka et al.	.545	.748	.716
PthDCode	**.586**±.016	**.793**±.005	**.742**±.005
HYENA, Ren et al.	.543	.695	.681
FIGER, Ren et al.	.589	.763	.749

Table 1: Entity classification evaluation on original data (top four rows). For comparison, we also provide results by Ren et al. (2016) on denoised data (bottom two rows).

	Level 1			Level 2		
	P	R	F_1	P	R	F_1
Yogatama et al.	**.828**	.704	.761	**.682**	.471	.557
PthDCode	.788	**.830**	**.808**	.534	**.641**	**.583**

Table 2: Per-level evaluation

trained end-to-end.

Table 3 shows, for some examples, which five words received the highest attention on level 1 (L1) and on level 2 (L2). The words are ordered from highest to lowest attention. We see that PthDCode attends to "from" for the location "Glasgow", but not for the organization "University of Glasgow". We also see that some words appear only on one of the two levels, e.g., for the mention "Glasgow", the context word "Glasgow" only appears on level 2. This indicates the benefit of level-wise attention. The last row shows an example of two types, */PEOP, /PEOP/Ethnc*, that are correct, but are not part of the gold standard, so we count them as errors.

4 Related work

Named entity recognition (NER) is the joint problem of entity mention segmentation and entity mention classification (Finkel et al., 2005; McCallum and Li, 2003). Most work on NER uses a small set of coarse-grained labels like *person* and *location*, e.g., MUC-7 (Chinchor and Robinson, 1998). Most work on the fine-grained FIGER (Ling and Weld, 2012) and HYENA (Yosef et al., 2012) taxonomies has cast NER as a two-step process (Elsner et al., 2009; Ritter et al., 2011; Collins and Singer, 1999) of entity mention segmentation followed by entity mention classification. The reason for two-step is the high complexity of joint models for fine-grained entity recognition. A joint model like CRF (Lafferty et al., 2001) has a state space corresponding to segmentation type times semantic types. Introducing a larger class set into joint models already increases the complexity of learning drastically, furthermore the multilabel nature of fine-grained entity mention classification explodes the state space of the exponential model further (Ling and Weld, 2012).

Utilizing fine-grained entity information enhances the performance for tasks like named entity disambiguation (Yosef et al., 2012), relation extraction (Ling and Weld, 2012) and question answering (Lin et al., 2012; Lee et al., 2006). A major challenge with fine grained entity mention classification is the scarcity of human annotated datasets. Currently, most of the datasets are collected through distant supervision, utilizing Wikipedia texts with anchor links to obtain entity mentions and using knowledge bases like Freebase and YAGO to obtain candidate types for the mention. This introduces noise and complexities like unrelated labels, redundant labels and large sizes of candidate label sets. To address these challenges, Ling and Weld (2012) mapped Freebase types to their own tag set with 113 types, Yosef et al. (2012) derived a 505-subtype fine-grained taxonomy using YAGO knowledge base, Gillick et al. (2014) devised heuristics to filter candidate types and, most recently, Ren et al. (2016) proposed a heterogeneous partial-label embedding framework to denoise candidate types by jointly embedding entity mentions, context features and entity type hierarchy.

We address fine-grained entity mention classification in this paper. A related problem is fine-grained entity typing: the problem of predicting the complete set of types of the entity that a mention refers to (Yaghoobzadeh and Schütze, 2017). For the sentences "Obama was elected president" and "Obama graduated from Harvard in 1991", fine-grained entity mention classification should predict "politician" for the first and "lawyer" for the second. In contrast, given a corpus containing these two sentences, fine-grained entity typing should predict the types {"politician", "lawyer"} for "Obama".

A common approach for solving hierarchical problems has been flat classification, i.e., not making direct use of the hierarchy. But exploiting the hierarchical organization of the classes reduces complexity, makes better use of training data in learning and enhances performance. Gillick et al. (2014) showed that addressing the entity classification problem with a hierarchical approach

mention	predict types	left context	right context	L1 attention	L2 attention
Lexar	/ORG, /ORG/Comp	According to PhotographyBlog , SanDisk and	have no immediate plans to produce XQD or WiFi SD cards .	to cards According SanDisk .	According . SanDisk cards and
University of Glasgow	/ORG, /ORG/ED-INST	The study is from the College of Medical , Veterinary & Life Sciences ,	, Glasgow , UK .	The . Sciences Glasgow ,	The . Veterinary Sciences study
Glasgow	/LOC, /LOC/city	from the College of Medical , Veterinary & Life Sciences , University of Glasgow ,	, UK .	from . University the UK	from . Glasgow College Veterinary
South Asian	/LOC, /PEOP, /PEOP/Ethnc	" The	student groups and cultures are very different than the East Asian student groups and cultures	cultures " student cultures The	cultures " student The cultures

Table 3: Top 5 Attention per level (L1/L2). ORG = organization, Comp = company ED-INST = educational_institution, LOC = Location, PEOP = People, Ethnc = ethnicity

through local classifiers for each label in the hierarchy and enforcing their outputs to follow a single path in it improved performance. Similarly, Yosef et al. (2012) used a set of support vector machine classifiers corresponding to each node in the hierarchy and then postprocessed them during inference through a metaclassifier. Yogatama et al. (2015), using a kernel enhanced WSABIE embedding method (Weston et al., 2011), learned an embedding for each type in the hierarchy and during inference filtered out predicted types that exceeded a threshold limit and did not fit into a path. Ren et al. (2016) showed that mapping a set of correlations, more specifically correlations of the types in the hierarchy, into an embedding space generates embeddings for mentions and types. These embeddings were then used for filtering the noisy candidate types and for denoising the train corpus. Ren et al. (2016) also showed that using the denoised corpus with baseline methods of (Ling and Weld, 2012; Yosef et al., 2012) enhanced the performance of those baseline methods significantly.

Recurrent neural networks (RNN) have been a successful model for sequence modeling tasks. Introduction of RNN based encoder-decoder architectures (Cho et al., 2014; Sutskever et al., 2014) addressed the end to end sequence to sequence learning problem that does not highly depend on lengths of sequences. Bahdanau et al. (2014) included attention mechanism to an encoder-decoder architecture and subsequently several other methods used them to improve performance on a range of tasks, e.g., machine translation (Bahdanau et al., 2014), image captioning (Xu et al., 2015), question answering (Kumar et al., 2016), morphological reinflection (Kann and Schütze, 2016). Recently, Shimaoka et al. (2016) and Shimaoka et al. (2017) included attention weighted contextual information into their logistic classification based entity classification model and showed improvement over traditional and non-attention based LSTM models.

In this paper, we describe the first decoder for hierarchical classification. It is trained end-to-end to predict paths from root to leaf nodes and also leverages attention-weighted sums of hidden state vectors of context when predicting classes at each level of the hierarchy.

5 Conclusion

We introduced an entity mention classification model that learns to predict types from an entity type hierarchy using an encoder-decoder with a level-wise contextual attention mechanism. A clear improvement in performance is observed at each level as well as in overall type hierarchy prediction compared to models trained in a comparable setting and performance close to models trained on datasets that have been denoised. We attribute this good performance to the fact that our method is the first neural network model for hierarchical classification that can be trained end-to-end while taking into account the tree structure of the entity classes through direct modeling of paths in the hierarchy.

Acknowledgments. We thank Stephan Baier, Siemens CT members and the anonymous reviewers for valuable feedback. This research was supported by Bundeswirtschaftsministerium (bmwi.de), grant 01MD15010A (Smart Data Web).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Charles E. Brown. 1998. Coefficient of variation. In *Applied multivariate statistics in geohydrology and related sciences*, pages 155–157. Springer.

Nancy Chinchor and Patricia Robinson. 1998. Appendix e: Muc-7 named entity task definition (version 3.5). In *Seventh Message Understanding Conference (MUC-7)*, Fairfax, Virginia. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110.

Micha Elsner, Eugene Charniak, and Mark Johnson. 2009. Structured generative models for unsupervised named-entity clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 164–172, Boulder, Colorado. Association for Computational Linguistics.

Rose Jenny Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, Ann Arbor, Michigan. Association for Computational Linguistics.

Daniel Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *CoRR*, abs/1412.1820.

Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinflection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 555–560, Berlin, Germany, August. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1378–1387, New York City, NY, USA, June. JMLR.org.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 282–289, Williams College, Williamstown, MA, USA, July. Morgan Kaufmann.

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. In Hwee Tou Ng, Mun-Kew Leong, Min-Yen Kan, and Dong-Hong Ji, editors, *Information Retrieval Technology, Third Asia Information Retrieval Symposium, AIRS 2006*, volume 4182 of *Lecture Notes in Computer Science*, pages 581–587, Singapore, October. Springer.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Thomas Lin, Mausam, and Oren Etzioni. 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*, pages 893–903, Jeju Island, Korea, July. Association for Computational Linguistics.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In Jörg Hoffmann and Bart Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 94–100, Toronto, Ontario, Canada, July. AAAI Press.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 188–191. Edmonton, Canada.

Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order crfs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Xiang Ren, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, and Jiawei Han. 2016. Label noise reduction in entity typing by heterogeneous partial-label embedding. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1825–1834, San Francisco, CA, USA, August. ACM.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK. Association for Computational Linguistics.

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, California, USA, June. 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*, Valencia, Spain, April. Association for Computational Linguistics. to appear.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112, Montreal, Quebec, Canada, December.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. WSABIE: scaling up to large vocabulary image annotation. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2764–2770, Barcelona, Catalonia, Spain, July. IJCAI/AAAI.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057, Lille, France, July. JMLR.org.

Yadollah Yaghoobzadeh and Hinrich Schütze. 2017. Multi-level representations for fine-grained typing of knowledge base entities. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain. Association for Computational Linguistics. to appear.

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.

Amir Mohamed 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.