Fine-Grained Named Entity Recognition using ELMo and Wikidata

Cihan Dogan, Aimore Dutra, Adam Gara, Alfredo Gemma,
Lei Shi, Michael Sigamani, Ella Walters
Constellation AI
7 Carlisle Street
London, W1D 3BW, United Kingdom
michaelsigamani@constellation.ai

Abstract

Fine-grained Named Entity Recognition is a task whereby we detect and classify entity mentions to a large set of types. These types can span diverse domains such as finance, healthcare, and politics. We observe that when the type set spans several domains the accuracy of the entity detection becomes a limitation for supervised learning models. The primary reason being the lack of datasets where entity boundaries are properly annotated, whilst covering a large spectrum of entity types. Furthermore, many named entity systems suffer when considering the categorization of fine grained entity types. Our work attempts to address these issues, in part, by combining state-of-the-art deep learning models (ELMo) with an expansive knowledge base (Wikidata). Using our framework, we cross-validate our model on the 112 fine-grained entity types based on the hierarchy given from the Wiki(GOLD) dataset.

1 Introduction

Named entity recognition (NER) (Collins and Singer, 1999; Tjong Kim Sang and De Meulder, 2003; Ratinov and Roth, 2009; Manning et al., 2014) is the process by which we identify text spans which mention named entities, and to classify them into predefined categories such as *person*, *location*, *organization* etc. NER serves as the basis for a variety of natural language processing (NLP) applications such as relation extraction (Mintz et al., 2009), machine translation (Koehn et al., 2007), question answering (Lin et al., 2012) and knowledge base construction (Dong et al., 2014). Although early NER systems have been successful in producing adequate recognition accuracy, they often require significant human effort in carefully designing rules or features.

In recent years, deep learning methods been employed in NER systems, yielding state-of-the-art performance. However, the number of types detected are still not sufficient for certain domain-specific applications. For relation extraction, identifying fine-grained types has been shown to significantly increase the performance of the extractor (Ling and Weld, 2012; Koch et al., 2014) since this helps in filtering out candidate relation types which do not follow this type constraint. Furthermore, for question answering fine-grained Named Entity Recognition (FgNER) can provide additional information helping to match questions to its potential answers thus improving performance (Dong et al., 2015). For example, Li and Roth (Li and Roth, 2002) rank questions based on their expected answer types (i.e. will the answer be *food*, *vehicle* or *disease*).

Typically, FgNER systems use over a hundred labels, arranged in a hierarchical structure. We find that available training data for FgNER typically contain noisy labels, and creating manually annotated training data for FgNER is a time-consuming process. Furthermore, human annotators will have to assign a subset of correct labels from hundreds of possible labels making this a somewhat arduous task. Currently, FgNER systems use distant supervision (Craven and Kumlien, 1999) to automatically generate training data. Distant supervision is a technique which maps each entity in the corpus to knowledge bases such as Freebase (Bollacker et al., 2008), DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007) and helps with the generation of labeled data. This method will assign the same set of labels to all mentions of a particular entity in the corpus. For

example, "Barack Obama" is a person, politician, lawyer, and author. If a knowledge base has these four matching labels, the distant supervision technique will assign all of them to every mention of "Barack Obama". Therefore, the training data will also fail to distinguish between mentions of "Barack Obama" in all subsequent utterances.

Ling et al. (2012) proposed the first system for FgNER, where they used 112 overlapping labels with a linear classifier perceptron for multi-label classification. Yosef et al. (2012) used multiple binary SVM classifiers to assign entities to a set of 505 types. Gillick et al. (2014) introduced context dependent FgNER and proposed a set of heuristics for pruning labels that might not be relevant given the local context of the entity. Yogatama et al. (2015) proposed an embedding based model where user-defined features and labels were embedded into a low dimensional feature space to facilitate information sharing among labels.

Shimaoka et al. (2016) proposed an attentive neural network model which used long short-term memory (LSTMs) to encode the context of the entity, then used an attention mechanism to allow the model to focus on relevant expressions in the entity mention's context. To learn entity representations, we propose a scheme which is potentially more generalizable.

1.1 Datasets

We evaluate our model on two publicly available datasets. The statistics for both are shown in Table 1. The details of these datasets are as follows:

OntoNotes: OntoNotes 5.0 (Weischedel et al., 2013) includes texts from five different text genres: broadcast conversation (200k), broadcast news (200k), magazine (120k), newswire (625k), and web data (300k). This dataset is annotated with 18 categories.

Wiki(GOLD): The training data consists of Wikipedia sentences and was automatically generated using a distant supervision method, mapping hyperlinks in Wikipedia articles to Freebase, which we do not use in this study. The test data, mainly consisting of sentences from news reports, was manually annotated as described in (Ling and Weld, 2012). The class hierarchy is shown in Figure 1. This dataset is annotated with 7 main categories (bold text in Figure 1), which maps di-

Datasets	OntoNotes	Wiki(GOLD)
# types	18	112
# training labels	239,617	NA
# evaluation labels	23,325	5,943

Table 1: Statistics of the datasets used in this work.

rectly to OntoNotes. The miscellaneous category in Figure 1 does not have direct mappings, so future work may include redefining these categories so the mappings are more meaningful.

Figure 1: The 112 tags used in Wiki(GOLD). The tags in bold are extracted in the step described in Section 2.1. The finer grained tags are extracted as a final step described in Section 2.2.

1.2 Evaluation Metrics

NER involves identifying both entity boundaries and entity types. With "exact-match evaluation", a named entity is considered correctly recognized only if both the boundaries and type match the ground truth (Ling and Weld, 2012; Yogatama et al., 2015; Shimaoka et al., 2016). Precision, Recall, and F-1 scores are computed on the number of true positives (TP), false positives (FP), and false negatives (FN). Their formal definitions are as follows:

- True Positive (TP): entities that are recognized by NER and match the ground truth.

- False Positive (FP): entities that are recognized by NER but do not match the ground truth.

- False Negative (FN): entities annotated in

the ground which that are not recognized by NER.

Precision measures the ability of a NER system to present only correct entities, and Recall measures the ability of a NER system to recognize all entities in a corpus.

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

The F-1 score is the harmonic mean of precision and recall, and the balanced F-1 score is the variant which is most commonly used. This is defined as:

$$\text{F-1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Since most NER systems involve multiple entity types, it is often required to assess the performance across all entity classes. Two measures are commonly used for this purpose: the macro-averaged F-1 score and the micro-averaged F-1 score. The macro-averaged F-1 score computes the F-1 score independently for each entity type, then takes the average (hence treating all entity types equally). The micro-averaged F-1 score aggregates the contributions of entities from all classes to compute the average (treating all entities equally). We use the micro-averaged F-1 in our study since this accounts for label imbalances in the evaluation data and therefore a more meaningful statistic.

2 Method

Over the few past years, the emergence of deep neural networks has fundamentally changed the design of entity detection systems. Consequently, recurrent neural networks (RNN) have found popularity in the field since they are able to learn long term dependencies of sequential data. The recent success of neural network based architectures principally comes from its deep structure. Training a deep neural network, however, is a difficult problem due to vanishing or exploding gradients. In order to solve this, LSTMs were proposed. An LSTM is an internal memory cell controlled by forget gate and input gate networks. A forget gate in an LSTM layer which determines how much prior memory should be passed into the next time increment. Similarly, an input gate scales new input to memory cells. Depending on the states of both gates, LSTM is able to capture long-term or short-term dependencies for sequential data. This is an ideal property for many NLP tasks.

2.1 NER using ELMo

Recently, Peters et al. (Peters et al., 2018) proposed ELMo word representations. ELMo extends a traditional word embedding model with features produced bidirectionally with character convolutions. It has been shown that the utilization of ELMo for different NLP tasks result in improved performance compared to other types of word embedding models such as Word2Vec (Mikolov et al., 2013), GloVe (Ma et al., 2013), and fastText (Wang et al., 2013).

The architecture of our proposed model is shown in Figure 2. The input is a list of tokens and the output are the predicted entity types. The ELMo embeddings are then used with a residual LSTM to learn informative morphological representations from the character sequence of each token. We then pass this to a softmax layer as a tag decoder to predict the entity types.

Hyperparameter settings: The hidden-layer size of each LSTM within the model is set 512. We use a dropout with the probability of 0.2 on the output of the LSTM encoders. The embedding dimension from ELMo is 1024. The optimization method we use is Adam (Kingma and Ba, 2014). We train with a batch size of 32 for 30 epochs. The model was implemented using the TensorFlow[1] framework.

2.2 Entity Linking using Wikidata

Entity linking (EL) (Shen et al., 2018), also known as named entity disambiguation or normalization, is the task to determine the identity of entities mentioned in a piece of text with reference to a knowledge base. There are a number of knowledge bases that provide a background repository for entity classification of this type. For this study, we use Wikidata, which can be seen diagrammatically in Figure 2. Systems such as Deep-Type (Raiman et al., 2018) integrate symbolic information into the reasoning process of a neural network with a type system and show state-of-the-art performances for EL. They do not, however, quote results on Wiki(GOLD) so a direct comparison is difficult.

While these knowledge bases provide semantically rich and fine-granular classes and relationship types, the task of entity classification often requires associating coarse-grained classes with discovered surface forms of entities. Most existing
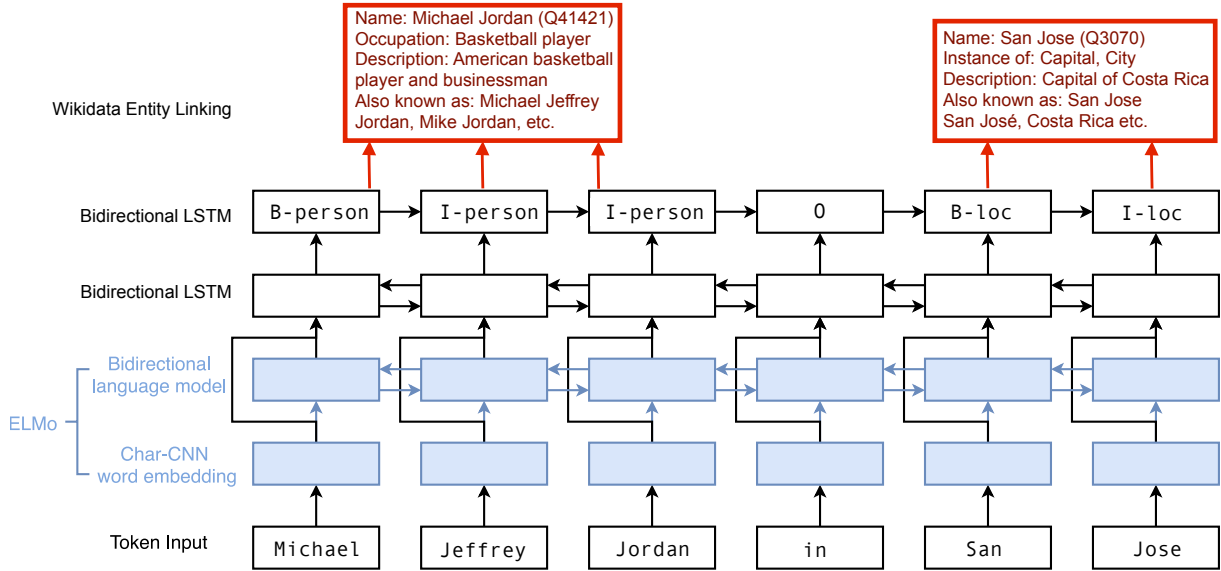
---

[1] http://tensorflow.org/

Figure 2: The full model pipeline. The first level involves token embeddings from ELMo which are fed into a residual LSTM module. The final layer involves passing the detected entities into a knowledge base, which in our case is Wikidata.

studies consider NER and entity linking as two separate tasks, whereas we try to combine the two. It has been shown that one can significantly increase the semantic information carried by a NER system when we successfully linking entities from a deep learning method to the related entities from a knowledge base (Ji et al., 2018; Phan et al., 2018).

Redirection: For the Wikidata linking element, we recognize that the lookup will be constrained by the most common lookup name for each entity. Consider the utterance (referring to the NBA basketball player) from Figure 2 "Michael Jeffrey Jordan in San Jose" as an example. The lookup for this entity in Wikidata is "Michael Jordan" and consequently will not be picked up if we were to use an exact string match. A simple method to circumvent such a problem is the usage of a redirection list. Such a list is provided on an entity by entity basis in the "Also known as" section in Wikidata. Using this redirection list, when we do not find an exact string match improves the recall of our model by 5-10%. Moreover, with the example of Michael Jordan (*person*), using our current framework, we will always refer to the retired basketball player (Q41421). We will never, for instance, pick up Michael Jordan (Q27069141) the American football cornerback. Or in fact any other Michael Jordan, famous or otherwise. One possible method to overcome this is to add a disam-

biguation layer, which seeks to use context from earlier parts of the text. This is, however, work for future improvement and we only consider the most common version of that entity.

Clustering: The Wikidata taxonomy provides thousands of possible *instance of*, and *subclass of* types for our entities. Consequently, in order to perform a meaningful validation of our model, we must find a way to cluster these onto the 112 types provided by Wiki(GOLD). Our clustering is performed as follows:

1. If the entity type is either *person*, *location*, *organization* we use the NECKAr (Geiß et al., 2018) tool to narrow down our list of searchable entities.

2. We then look at either the *occupation* for *person*, or *instance of* for *location/organization* categories to map to the available subtypes.

3. If the entity type is not *person*, *location*, or *organization* we search all of Wikidata.

4. The clustering we perform in part 1 or 2 is from a cosine similarity of the entity description to the list of possible subtypes for that entity. For this we use Word2Vec word embeddings trained on Wikipedia. We set the minimum threshold of the average cosine similarity to be 0.1.

4

As an example, consider the test sentence: "The device will be available on sale on 20th April 2011 on amazon uk Apple's iPad" from Figure 3. First, we tag iPad as *product* using the context encoder described in Section 2.1. We then search Wikidata and return the most common variant for that entity in this case Q2796 (the most referenced variant is the one with the lowest Q-id). We then calculate a cosine similarity of the description, in this case "line of tablet computers", with the possible subtypes of *product*. The possible subtypes, in this case, are *engine, airplane, car, ship, spacecraft, train, camera, mobile phone, computer, software, game, instrument, ship, weapon*. We return the highest result above 0.1, which in this case is *computer* (0.54).

3 Results

The results for each class type are shown in Table 2, with some specific examples shown in Figure 3. For the Wiki(GOLD) we quote the micro-averaged F-1 scores for the entire top level entity category. The total F-1 score on the OntoNotes dataset is 88%, and the total F-1 cross-validation score on the 112 class Wiki(GOLD) dataset is 53%. It is worth noting that one could improve Wiki(GOLD) results by training directly using this dataset. However, the aim is not to tune our model specifically on this class hierarchy. We instead aim to present a framework which can be modified easily to any domain hierarchy and has acceptable out-of-the-box performances to any fine-grained dataset. The results in Table 2 (OntoNotes) only show the main 7 categories in OntoNotes which map to Wiki(GOLD) for clarity. The other categories (*date, time, norp, language, ordinal, cardinal, quantity, percent, money, law*) have F-1 scores between 80-90%, with the exception of *time* (65%)

4 Conclusion and Future Work

In this paper, we present a deep neural network model for the task of fine-grained named entity classification using ELMo embeddings and Wikidata. The proposed model learns representations for entity mentions based on its context and incorporates the rich structure of Wikidata to augment these labels into finer-grained subtypes. We can see comparisons of our model made on Wiki(GOLD) in Table 3. We note that the model performs similarly to existing systems without be-

Figure 3: Some example outputs from the full model pipeline on the Wiki(GOLD) evaluation set.

Label	OntoNotes		Wiki(GOLD)			
	%	F-1	%	Prec.	Rec.	F-1
person	14	90	23	79	59	66
location	14	93	37	62	47	54
organization	24	85	26	45	16	23
event	1	70	2	81	17	28
product	1	56	2	44	4	8
building	1	65	4	81	17	11
art	2	54	0	0	0	0

Table 2: Performance of our model from the NER classifier evaluated on OntoNotes, and the 112 subclass Wikidata linking step evaluated on Wiki(GOLD). The first column denotes the percentage breakdown per class type. The precision, recall, and F-1 scores are shown for Wiki(GOLD). For OntoNotes the precision and recall are identical for each category, therefore we only quote F-1. All values are quoted as a percentage and rounded to the nearest whole number. Since the table only shows 7 categories, the percentages will not sum to 100.

ing trained or tuned on that particular dataset. Future work may include refining the clustering method described in Section 2.2 to extend to types other than *person*, *location*, *organization*, and also to include disambiguation of entity types.

References

[Collins and Singer1999] Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110.

[Tjong Kim Sang and De Meulder2003] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter

Datasets	OntoNotes	Wiki(GOLD)
Our model	88.7%	52.8%
Akbik et al. (2018)	89.7%	NA
Link et al. (2012)	NA	53.2%

Table 3: Comparison with existing models.

Daelemans and Miles Osborne, editors, *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

[Ratinov and Roth2009] Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado, June. Association for Computational Linguistics.

[Manning et al.2014] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, June. Association for Computational Linguistics.

[Mintz et al.2009] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, August. Association for Computational Linguistics.

[Koehn et al.2007] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

[Lin et al.2012] 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.

[Dong et al.2014] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 601–610, New York, NY, USA. ACM.

[Ling and Weld2012] 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.

[Koch et al.2014] Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S. Weld. 2014. Type-aware distantly supervised relation extraction with linked arguments. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1891–1901, Doha, Qatar, October. Association for Computational Linguistics.

[Mitchell et al.2015] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2302–2310. AAAI Press.

[Mitchell et al.2015] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2015. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2302–2310. AAAI Press.

[Dong et al.2015] Li Dong, Furu Wei, Hong Sun, Ming Zhou, and Ke Xu. 2015. A hybrid neural model for type classification of entity mentions. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1243–1249. AAAI Press.

[Li and Roth2002] Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Craven and Kumlien1999] Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86. AAAI Press.

[Bollacker et al.2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph

database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

[Auer et al.2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 722–735, Berlin, Heidelberg. Springer-Verlag.

[Suchanek et al.2007] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA. ACM.

[Yosef et al.2012] 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, December. The COLING 2012 Organizing Committee.

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

[Yogatama et al.2015] 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, July. Association for Computational Linguistics.

[Shimaoka et al.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*, pages 69–74, San Diego, CA, June. Association for Computational Linguistics.

[Weischedel et al.2013] 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*.

[Peters et al.2018] Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2018. Deep contextualized word representations. In *Proc. NAACL-HLT*, 2018, pp. 2227–2237.

[Mikolov et al.2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean 2013. Distributed representations of words and phrases and their compositionality In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, 2013.

[Ma et al.2013] X. Ma and E. Hovy 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf In *Proc. ACL, pp. 1064–1074*, 2016.

[Li et al.2013] P.-H. Li, R.-P. Dong, Y.-S. Wang, J.-C. Chou, and W.-Y. Ma 2017. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks In *Proc. EMNLP, pp. 2664–2669*, 2017.

[Wang et al.2013] C. Wang, K. Cho, and D. Kiela 2018. Code-switched named entity recognition with embedding attention In *Proc. the Third Workshop on Computational Approaches to Linguistic Code-Switching, pp. 154–158*, 2018.

[Kingma and Ba2014] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Shen et al.2018] W. Shen, J. Han, J. Wang, X. Yuan, and Z. Yang 2018. Shine+: A general framework for domain-specific entity linking with heterogeneous information networks In *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 2, pp. 353–366, 2018.

[Raiman et al.2018] Jonathan Raiman, Olivier Raiman 2018. DeepType: Multilingual Entity Linking by Neural Type System Evolution In *arXiv preprint arXiv:1802.01021*, 2018.

[Ji et al.2018] Z. Ji, A. Sun, G. Cong, and J. Han 2018. Joint recognition and linking of fine-grained locations from tweets In *Proc. WWW*, 2016, pp. 1271–1281.

[Phan et al.2018] M. C. Phan, A. Sun, Y. Tay, J. Han, and C. Li 2018. Pair-linking for collective entity disambiguation: Two could be better than all In *arXiv preprint arXiv:1802.01074*, 2018.

[Geiß et al.2018] Johanna Geiß, Andreas Spitz, Michael Gertz, Georg Rehm, Thierry Declerck 2018. NECKAr: A Named Entity Classifier for Wikidata In *Springer International Publishing 115–129*, 2018.