SlotGAN: Detecting Mentions in Text via Adversarial Distant Learning

Daza, D.; Cochez, M.; Groth, P.

DOI
10.18653/v1/2022.spnlp-1.4

Publication date
2022

Document Version
Final published version

Published in
Sixth Workshop on Structured Prediction for NLP

SlotGAN: Detecting Mentions in Text via Adversarial Distant Learning

Daniel Daza[1,2,3], **Michael Cochez**[1,3], **Paul Groth**[2,3]
[1]Vrije Universiteit Amsterdam
[2]University of Amsterdam
[3]Discovery Lab, Elsevier, The Netherlands
d.dazacruz@vu.nl, m.cochez@vu.nl, p.groth@uva.nl

Abstract

We present SlotGAN, a framework for training a mention detection model that only requires unlabeled text and a gazetteer. It consists of a generator trained to extract spans from an input sentence, and a discriminator trained to determine whether a span comes from the generator, or from the gazetteer. We evaluate the method on English newswire data and compare it against supervised, weakly-supervised, and unsupervised methods. We find that the performance of the method is lower than these baselines, because it tends to generate more and longer spans, and in some cases it relies only on capitalization. In other cases, it generates spans that are valid but differ from the benchmark. When evaluated with metrics based on overlap, we find that SlotGAN performs within 95% of the precision of a supervised method, and 84% of its recall. Our results suggest that the model can generate spans that overlap well, but an additional filtering mechanism is required.

1 Introduction

Detecting mentions of entities in text is an important step towards the extraction of structured information from natural language sources. Mention Detection (MD) components can be found frequently in systems for Named Entity Recognition (NER) (Straková et al., 2019; Wang et al., 2021), entity linking (Wu et al., 2020; Cao et al., 2021), relationship extraction (Katiyar and Cardie, 2017; Zhong and Chen, 2021), and coreference resolution (Joshi et al., 2019; Xu and Choi, 2020; Kirstain et al., 2021), where accurately modeling mentions is crucial for downstream performance.

The MD task is often subsumed under NER, where most effective approaches employ supervised learning with exhaustively annotated datasets. These methods become less feasible in cases where we need to rapidly build MD systems, for example, when moving to a domain with incompatible NER classes; or when there are not enough resources to create a labeled dataset. In contrast, we assume that we have access to an unlabeled corpus, and a list of known entity names (i.e. a *gazetteer*). We propose SlotGAN– a framework for detecting mentions that uses a generator to extract spans conditioned on some input text, and a discriminator that determines whether a span comes from the generator, or from the gazetteer (see Fig. 1). In contrast with distant supervision methods that require training with false negatives (Ratner et al., 2016; Giannakopoulos et al., 2017; Shang et al., 2018), SlotGAN relies on the discriminator to learn patterns that are *not* likely to be names of entities (such as verb phrases, or very long spans, which rarely occur in a gazzetteer), thereby improving the generator's ability to detect valid mentions.

We evaluate the method in a MD task using the CoNLL 2003 English dataset (Tjong Kim Sang and De Meulder, 2003). We observe that the absence of strong supervision in SlotGAN results in different, yet valid notions of what constitutes an entity. For instance, while in the sentence *"On the road to Tripoli airport..."* the word *Tripoli* is selected as a gold mention, SlotGAN selects *Tripoli airport*. In this case, exact match metrics for NER underestimate performance, assigning zero precision and recall. To account for this, we introduce overlap-based metrics into the evaluation.

When using exact boundary match metrics, SlotGAN exhibits lower performance compared to different baselines. When evaluating overlap, precision (how much of the predicted span overlaps with the gold span) is within 95% of the performance of the supervised baseline, while recall (how much of the gold span is actually predicted) is within 84%. We observe that SlotGAN tends to generate more and longer spans than those in the benchmark, and in some cases it relies only on capitalization.

Our contributions are the following: 1) A framework towards distantly-supervised MD that avoids explicit training with false negatives, and an imple-

Proceedings of the Sixth Workshop on Structured Prediction for NLP, pages 32 - 39

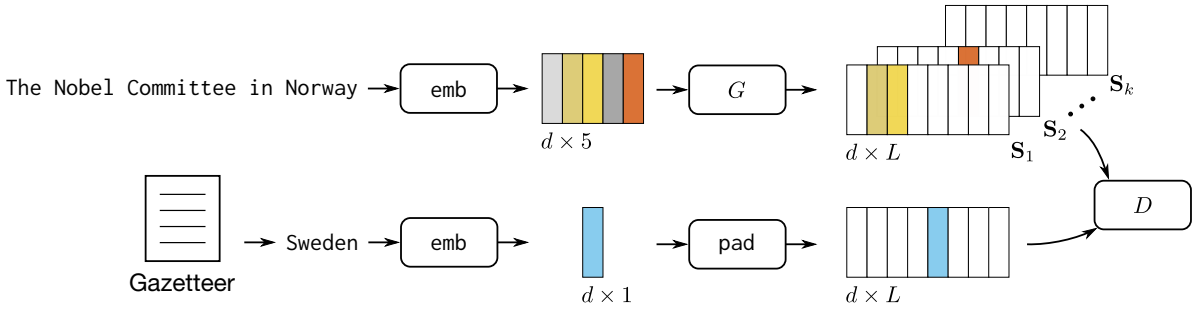

Figure 1: SlotGAN consists of a generator G trained to extract spans from an input sentence. We represent spans as matrices containing embeddings of words in a span, padded with zeros to a fixed length L. True spans are generated from a gazetteer. A discriminator D is trained to determine if a span was generated from G or from the gazetteer.

mentation via an end-to-end differentiable architecture for extracting distinct spans; 2) Evidence for the use of overlap-based metrics into the evaluation of MD methods to account for ambiguous cases in gold annotations; 3) An analysis of the performance of SlotGAN, identifying its failure modes and outlining directions of improvement.

2 SlotGAN

In the MD task, we are given a sentence from a corpus as a sequence of words $(w_1, w_2, ..., w_n)$. The output of the system is a set of spans that contain a mention, and each span is a tuple (i_s, i_e) where i_s is an integer indicating the position where the span starts, and i_e the position where it ends. Additionally, we have access to a gazetteer $E = (e_1, e_2, ..., e_N)$ containing names of entities relevant to a particular domain.

SlotGAN is a method for MD based on Generative Adversarial Networks (Goodfellow et al., 2014; Mirza and Osindero, 2014). It consists of a generator trained to extract spans from a sentence, and a discriminator that determines whether a span comes from the generator or from the gazetteer.

We define the embedding of a sentence $w = (w_1, ..., w_n)$ as a matrix $\mathsf{emb}(w) \in \mathbb{R}^{d \times n}$, where emb is a function that maps words to d-dimensional pretrained embeddings (for example, from the input embedding layer of BERT (Devlin et al., 2019)).

We represent each mention span in a sequence as a matrix in a space $\mathcal{S} = \mathbb{R}^{d \times L}$, where L is the length of the sequence. For a span (i_s, i_e), the matrix contains the embeddings of the words within the span, from column i_s to column i_e, and is zero in the remaining columns.

The generator takes as input the embedding matrix $\mathsf{emb}(w)$ of a sentence, and assigns each of its columns to one of k *slots*. The output is a sequence of k span representations $(\mathbf{S}_i)_{i=1}^k$ with $\mathbf{S}_i \in \mathcal{S}$, such that the j-th column of $\mathbf{S}_i$ contains the j-th column of the input matrix, if it was assigned to slot i. Unused columns of $\mathbf{S}_i$ are filled with zeros.

When sampling a name e of an entity in the gazetteer, we embed it as $\mathsf{emb}(e)$ and then add zero padding via a pad function until reaching a maximum length L, to obtain a span representation in $\mathcal{S}$. The amount of padding is added randomly to both sides of an entity name, with the purpose of emulating how in a sentence, a mention can appear at an arbitrary position. The discriminator takes as input span representations in $\mathcal{S}$, and outputs a score that should be high for samples from the gazetteer, and low for samples from the generator.

Denoting as p_w the distribution used to sample sentences from the corpus, and as p_e the distribution for sampling names from the gazetteer, the generator and discriminator are trained via gradient descent using the W-GAN (Arjovsky et al., 2017) minimax optimization objective:

$$\min_G \max_D \mathbb{E}_{e \sim p_e}[D(\mathsf{pad}(\mathsf{emb}(e)))] - \mathbb{E}_{w \sim p_w}\left[\sum_{i=1}^{k} D(G(\mathsf{emb}(w))_i)\right], \quad (1)$$

where we have denoted as $G(\mathsf{emb}(w))_i$ the i-th span representation produced by the generator.

To allow also *not* extracting any mentions when not required, we randomly introduce empty spans in the gazetteer, and we reformulate the generator objective with an equality constraint. Following Bastings et al. (2019), we define the constraint in terms of a differentiable function C such that $C(G(\mathsf{emb}(w)_i)$ counts the number of transitions from zero to non-zero, and vice versa, in a span representation. For valid spans, this should be equal to 2. We solve the problem introducing a Lagrange

multiplier λ, and the term in Eq. 1 that depends on the generator becomes

$$\min_{\lambda,G} \mathbb{E}_{w\sim p_w} \left[\sum_{i=1}^{k} -D(\mathbf{S}_i(w)) - \lambda(2 - C(\mathbf{S}_i(w)) \right], \tag{2}$$

where $\mathbf{S}_i(w)$ is a shorthand for $G(\texttt{emb}(w))_i$. This constraint prevents the generator from producing only empty spans.

At test time, we can use the spans produced by the generator as predictions for mentions. Alternatively, we can balance precision and recall by leveraging the discriminator, by only keeping spans with a score $D(\mathbf{S}_i(w)) > t$ where t is a threshold.

We implement the generator using BERT (Devlin et al., 2019), followed by a modified Slot Attention layer (Locatello et al., 2020) to model discrete selections of distinct spans. The discriminator is a temporal CNN. For more details on the architecture, we refer the reader to Appendix A.

3 Related Work

The task of MD has been addressed under NER effectively via supervised learning (Devlin et al., 2019; Straková et al., 2019; Peters et al., 2018; Yu et al., 2020; Wang et al., 2021). Some works address the lack of labeled data in a target domain by applying adaptation techniques from a source domain with labeled data (Zhou et al., 2019; Li et al., 2019; Zhang et al., 2021). In this work we focus on the case where annotations are not available.

Closer to our work are methods for weakly or distantly supervised learning, where heuristics and domain-specific rules are used to generate a noisy training set, often using external sources like gazetteers (Safranchik et al., 2020; Lison et al., 2020; Zhao et al., 2021; Ratner et al., 2016; Shang et al., 2018; Li et al., 2021a). These methods are limited by false negatives that reduce recall in MD. Furthermore, even though rules can be used to annotate a dataset at a large scale, the process of devising these rules in the first place can be tedious, and might require domain expert knowledge.

Luo et al. (2020) recently introduced a fully unsupervised method for NER that uses a pipeline of clustering over word embeddings, a generative model, and reinforcement learning to solve the NER task without any labels or external sources. These elements are optimized separately, whereas SlotGAN provides an end-to-end architecture.

4 Experiments

Datasets We evaluate MD performance using the CoNLL 2003 English dataset for NER (Tjong Kim Sang and De Meulder, 2003). For methods that require a dictionary of entity types or a gazetteer, we build it using the annotations in the training set. We also explore a pretraining strategy for SlotGAN, where we sample sentences from Wikipedia articles, and names of entities from Wikidata. Both are obtained from the July 2019 dumps.

Experimental setup We evaluate the performance of SlotGAN when trained with the CoNLL 2003 data only, and when pretraining with Wikipedia and Wikidata. We apply a threshold to all spans based on the discriminator score, selected using the validation set. Training and hyperparameter details can be found in Appendix B. Our implementation is available online[1].

Baselines We consider a string matching baseline where we label all spans present in the gazetteer, giving precedence to longer spans. We also compare with methods ranging from supervised, weakly supervised, to unsupervised. ACE (Wang et al., 2021) is a state-of-the-art method for supervised NER. AutoNER (Shang et al., 2018) is a weakly supervised method that requires a type dictionary. Lastly, we compare with the unsupervised method of Luo et al. (2020)[2].

Evaluation Recent works have highlighted the presence of unlabeled mentions in the CoNLL dataset, which has a negative effect when training and evaluating models based on exact match (Jie et al., 2019; Li et al., 2021b). Exact match metrics also penalize more strongly models that do not match boundaries exactly, than a model that does not predict a span at all (Manning, 2006; Esuli and Sebastiani, 2010). With this motivation, we also report overlap[3] by computing the intersection between gold and predicted spans. Precision is defined as the length of the intersection divided by the length of the predicted span, and recall is the length of the intersection divided by the length of the gold span. We denote these as OP and OR, respectively. Overlap F1 (OF1) is the harmonic mean of OP and OR. We report the average over all gold spans.

[1] https://github.com/dfdazac/slotgan

[2] Their implementation is not available. Results for P, R, and F1 from their paper.

[3] Partial matches have been considered by Segura-Bedmar et al. (2013), though not taking span lengths into account.

Method	Data	P	R	F1	OP	OR	OF1
String matching	Gazetteer	76.2	54.0	63.2	57.4	61.3	58.6
ACE (Wang et al., 2021)	Gold labels	96.0	97.1	96.5	98.3	98.1	98.1
AutoNER (Shang et al., 2018)	Type dictionary	88.4	94.2	91.2	97.4	97.2	96.9
Unsupervised (Luo et al., 2020)	Domain concepts	80.0	72.0	76.0	—	—	—
SlotGAN - no pretraining	Gazetteer	55.9	66.1	60.6	82.9	79.5	82.9
SlotGAN - pretrained		60.1	71.1	65.2	93.2	83.0	84.7

Table 1: Mention detection results evaluated via exact match precision (P), recall (R), and F1 score; and overlap metrics (preceded with O). The "Data" column indicates what is required to train the model in addition to a corpus.

Gold	on the road to **[Tripoli]** airport
Predicted	on the road to **[Tripoli airport]**
Gold	**[Belgian]** police said on Saturday
Predicted	**[Belgian police]** said on Saturday
Gold	**[JOHNSON]** WINS UNANIMOUS POINTS VERDICT
Predicted	**[JOHNSON WINS UNANIMOUS POINTS VERDICT]**
Gold	BASKETBALL - **[BENETTON]** BEAT **[DINAMO]** 92 - 81
Predicted	**[BASKETBALL]** - **[BENETTON BEAT DINAMO]** 92 - 81

Table 2: Comparison of gold spans and spans predicted by SlotGAN.

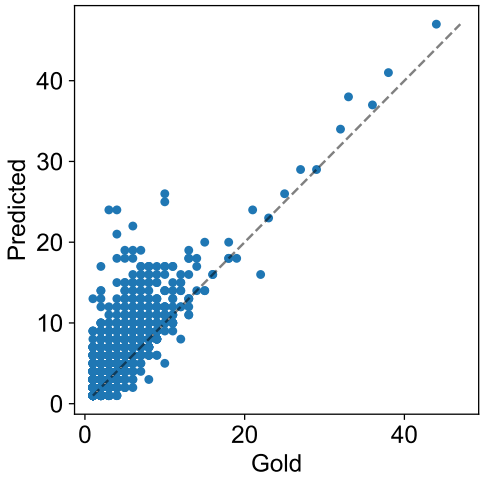

Figure 2: Number of words assigned to a mention per sentence, computed over gold and predicted spans.

Results We present MD results in Table 1. We observe that pretraining with Wikipedia and Wikidata entity names helps to improve the performance over a version trained with the CoNLL 2003 data only. The higher recall of SlotGAN in comparison with the string matching baseline shows that the generator is not simply memorizing the gazetteer and can thus detect mentions not seen during training. However, its precision and recall are low compared to other systems. We attribute this partly to the lack of strong supervision of the generator, which results in boundaries that differ from gold spans, and detection of more mentions than those present in the dataset. The overlap-based metrics show that on average, predicted spans overlap 93% and gold spans overlap 83% with the intersection. This indicates that extra words are added to predicted spans, and boundary mismatch, though these values of precision and recall are within 95% and 84% of the supervised baseline, respectively.

A closer analysis of the length of overlapping spans shows that in 69.4% of the cases the length is the same as gold spans, in 21.1% the predicted span is longer, and in 9.5% it is shorter. This often leads to mentions that are actually correct, as shown in Table 2. However, SlotGAN also produces spans that do not overlap with any gold span. This can be observed by plotting the average number of words assigned to a mention by the model versus the gold annotations, as shown in Fig 2. We see that across different numbers of mention words for the gold annotations, SlotGAN produces a higher number in average. We also find cases where it relies on capitalization only, which becomes problematic in upper case sentences: for regular sentences, there is no exact boundary match in 11% of the cases. For sentences in upper case, this increases to 23%.

5 Conclusion

We have presented SlotGAN, a method for training a mention detector that only requires unlabeled text and a list of entity names, that relies on implicit supervision provided by a discriminator that is also optimized during training. This results in spans that overlap well with gold spans, but also a tendency towards generating more and longer spans, and relying on capitalization only. This suggests that spans predicted by SlotGAN are likely to be correct,

but an additional mechanism is needed to filter them. This can be enforced via tighter constraints on generated spans, or a stronger discriminator.

Even though its performance is close to a supervised model according to overlap-based metrics, it cannot match other methods that also use a gazetteer or are unsupervised. In spite of this, we consider SlotGAN a promising framework for other IE tasks with less supervision, for example, where relations between slots could be induced. The end-to-end architecture also presents an opportunity for fine-tuning with gold labels, which we plan to explore in future work.

Acknowledgments

This project was funded by Elsevier's Discovery Lab.

References

Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Andrea Esuli and Fabrizio Sebastiani. 2010. Evaluating information extraction. In *Multilingual and Multimodal Information Access Evaluation, International Conference of the Cross-Language Evaluation Forum, CLEF 2010, Padua, Italy, September 20-23, 2010. Proceedings*, volume 6360 of *Lecture Notes in Computer Science*, pages 100–111. Springer.

Athanasios Giannakopoulos, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. 2017. Unsupervised aspect term extraction with B-LSTM & CRF using automatically labelled datasets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 180–188, Copenhagen, Denmark. Association for Computational Linguistics.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680.

Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5767–5777.

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. 2019. Better modeling of incomplete annotations for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 729–734, Minneapolis, Minnesota. Association for Computational Linguistics.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

Yuval Kirstain, Ori Ram, and Omer Levy. 2021. Coreference resolution without span representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 14–19, Online. Association for Computational Linguistics.

Jiacheng Li, Haibo Ding, Jingbo Shang, Julian McAuley, and Zhe Feng. 2021a. Weakly supervised named entity tagging with learnable logical rules. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International*

Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4568–4581, Online. Association for Computational Linguistics.

Jing Li, Deheng Ye, and Shuo Shang. 2019. Adversarial transfer for named entity boundary detection with pointer networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5053–5059. ijcai.org.

Yangming Li, Lemao Liu, and Shuming Shi. 2021b. Empirical analysis of unlabeled entity problem in named entity recognition. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Pierre Lison, Jeremy Barnes, Aliaksandr Hubin, and Samia Touileb. 2020. Named entity recognition without labelled data: A weak supervision approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1518–1533, Online. Association for Computational Linguistics.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. 2020. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ying Luo, Hai Zhao, and Junlang Zhan. 2020. Named entity recognition only from word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8995–9005. Association for Computational Linguistics.

Christopher Manning. 2006. Doing named entity recognition? don't optimize for F1. *NLPers Blog*, 25. Accessed on November, 2021.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *CoRR*, abs/1411.1784.

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, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

John C. Platt and Alan H. Barr. 1987. Constrained differential optimization. In *Neural Information Processing Systems, Denver, Colorado, USA, 1987*, pages 612–621. American Institue of Physics.

Alexander J. Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3567–3575.

Esteban Safranchik, Shiying Luo, and Stephen H. Bach. 2020. Weakly supervised sequence tagging from noisy rules. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5570–5578. AAAI Press.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064, Brussels, Belgium. Association for Computational Linguistics.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Automated concatenation of embeddings for structured prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2643–2660. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533, Online. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Neural mention detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1–10, Marseille, France. European Language Resources Association.

Tao Zhang, Congying Xia, Philip S. Yu, Zhiwei Liu, and Shu Zhao. 2021. PDALN: Progressive domain adaptation over a pre-trained model for low-resource cross-domain named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5441–5451, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinyan Zhao, Haibo Ding, and Zhe Feng. 2021. GLaRA: Graph-based labeling rule augmentation for weakly supervised named entity recognition. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3636–3649, Online. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. 2019. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3461–3471, Florence, Italy. Association for Computational Linguistics.

A Architectures

In our implementation of SlotGAN, the embedding function $\texttt{emb}(w)$ used to obtain embeddings of sentences and names in the gazetteer uses the pretrained WordPiece embeddings from the input layer of BERT (Devlin et al., 2019).

The generator consists of BERT, which for an input sentence of length n, outputs a matrix $\mathbf{H} \in \mathbb{R}^{d \times n}$ where d is the dimension of the output layer of BERT, equal to 768. We use the `bert-base-cased` implementation in HuggingFace's Transformer library (Wolf et al., 2019).

The output matrix is passed to a modified Slot Attention layer (Locatello et al., 2020), which we use as a differentiable mechanism to assign n input embeddings to k slots. In the original implementation, Slot Attention would assign each of the n outputs in the columns of $\mathbf{H}$ to k slots, by using a differentiable clustering algorithm. This algorithm works for a variable number of slots, by sampling k initial slot representations from a Gaussian distribution. In our experiments we use $k = 10$, and the number of iterations of the clustering algorithm is set to 3.

In the MD case, for words that do not belong to any mention, we want the generator to be able to assign them to a "default" slot. We achieve this by introducing an extra slot, whose representation, instead of sampled, is a single vector with a learned representation. Slot Attention in the generator thus contains $k+1$ slots, but the default slot is discarded when passing generated spans to the discriminator.

After discarding the default slot, the result is an attention mask $\mathbf{M} \in \mathbb{R}^{k \times n}$ where the m_{ij} entry indicates the fraction of input j assigned to slot i, and each column is normalized to 1. The i-th span representation is then obtained as

$$\mathbf{S}_i = \texttt{emb}(w) \odot \mathbf{M}_{i:}, \tag{3}$$

where $\mathbf{M}_{i:}$ is the i-th row of $\mathbf{M}$, and $\odot$ is broadcast element-wise multiplication.

For the discriminator we use a temporal CNN, where convolutions are applied along the sequence axis. The input is a matrix of span representations of shape $d \times L$, and the output is a scalar. The architecture is described in Table 3.

Layer	Output features	Activation
3×3 Conv	128	ReLU
3×3 Conv	64	ReLU
3×3 Conv	64	ReLU
3×3 Conv	64	—
Flatten		—
Linear	32	ReLU
Linear	1	—

Table 3: Architecture of the discriminator used in our experiments.

B Training Procedure

We train SlotGAN with mini-batches of 32 sentences. We update the generator once for every 5 updates of the discriminator. To let the discriminator accept empty spans as valid, we replace names from the gazetteer with an empty span with a probability of 0.5. We use a gradient penalty coefficient (Gulrajani et al., 2017) of 10 when computing the discriminator loss.

We use a learning rate of 2×10^{-5}, with a linear warm-up schedule for the first 10% of epochs. For the Lagrange multiplier, we use the Modified Differential Method of Multipliers (Platt and Barr, 1987) with a constant learning rate of 1×10^{-3}.

We run our experiments in a workstation with an Intel Xeon processor, 1 NVIDIA GeForce GTX 1080 Ti GPU with 11GB of memory, and 60GB of RAM. When pretraining with Wikipedia and Wikidata, we train SlotGAN with 20,000 updates of the generator, and 5,000 when training with the CoNLL 2003 dataset.