Global Span Selection for Named Entity Recognition

Urchade Zaratiana[*†], **Niama Elkhbir**[†], **Pierre Holat**[*†],
Nadi Tomeh[†], **Thierry Charnois**[†]
[*] FI Group, [†] LIPN, CNRS UMR 7030, France
{zaratiana,elkhbir,holat,tomeh,charnois}@lipn.fr

Abstract

Named Entity Recognition (NER) is an important task in Natural Language Processing with applications in many domains. In this paper, we describe a novel approach to named entity recognition, in which we output a set of spans (i.e., segmentations) by maximizing a global score. During training, we optimize our model by maximizing the probability of the gold segmentation. During inference, we use dynamic programming to select the best segmentation under a linear time complexity. We prove that our approach outperforms CRF and semi-CRF models for Named Entity Recognition. We make our code publicly available at https://github.com/urchade/global-span-selection.

1 Introduction

Named Entity Recognition is a crucial task in natural language processing whose purpose is to identify and classify salient entities in texts such as persons, organizations, and locations. Recognizing such entities is advantageous for applications such as relation extraction and machine translation. There are two main paradigms for NER: sequence labeling (SL) (Huang et al., 2015; Lample et al., 2016; Akbik et al., 2018) and span-based approaches (SB) (Sohrab and Miwa, 2018; Yu et al., 2020a; Li et al., 2021). SL frames NER as token-level prediction, using, for instance, the BIO (Ramshaw and Marcus, 1995) or BILOU (Ratinov and Roth, 2009) schemes, while SB considers spans (contiguous segments of tokens) as basic units instead of tokens and performs span-level classification by assigning a label to each entity and a special `null` label to non-entity spans.

SL is usually performed by representing the tokens using deep learning models, then using a Conditional Random Field (Lafferty et al., 2001) as the output layer. The best label sequence is computed using the Viterbi algorithm and learning typically maximizes the likelihood of gold sequences. In contrast, SB enumerates all candidate spans from an input text and computes their representation before feeding them into a softmax layer for classification.

One advantage of SBs is that they allow richer span representation compared to SL since span-level features are learned end to end. However, such *unstructured* SB models predict the label of each span independently. They are prone to produce overlapping entities which is forbidden in flat and nested NER. Prior works used a *greedy* decoding algorithm (Johnson, 1973; Yu et al., 2020b; Li et al., 2021) to obtain a set of non-overlapping entities. The highest-scoring entities are iteratively selected as long as they do not overlap with previously selected ones. Greedy decoding is efficient but tends to suffer from myopic bias. Choosing spans without regard to future decisions may results in suboptimal entity sets.

An alternative formulation of NER as joint segmentation and labeling with Semi-Markov CRFs has been proposed in the literature (Sarawagi and Cohen, 2005; Kong et al., 2016; Ye and Ling, 2018). This approach has two advantages: (a) it uses a globally-normalized model to compute the probability of each labeled segmentation as opposed to scoring each span independently; and (2) it guarantees no-overlap in the output entities by using a variant of the Viterbi algorithm for decoding. Nevertheless, semi-CRFs underperform in practice as we show in our experiments. We hypothesise that scoring segmentations composed of entities and non-entities is the main weakness. First, non-entity spans can be segmented in multiple ways all equally valid but only one of them is enforced by the semi-CRF, both during learning and inference. Furthermore, the majority of spans are non-entity, a considerable probability mass is wasted on uninteresting segmentations.

In this paper, we propose a new formulation for span-based NER that combines ideas from

Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS), pages 11 - 17
December 7, 2022 ©2022 Association for Computational Linguistics

two-steps (filtering and decoding) approaches and globally-normalized CRF-based models. Our approach starts by filtering all non-entity spans using a span classifier and constructing an *overlapping* graph of the remaining spans. A globally-normalized model is then used to compute the probability of each *maximal independent set (MIS)* within the graph. Each such set corresponds to a selection of non-overlapping entities. Learning and inference can be performed efficiently using dynamic programming as we explain in §2.2. Furthermore, we train the span classifier and the global entity selection model jointly using a multi-task objective. We show that our approach outperforms both SL and Semi-CRFs on all tasks and outperform two-step (filtering and greedy decoding) models on most.

2 Two-step Span-based NER

State-of-the-art span-based approaches employ a locally-normalized, unstructured span classifier to filter non entity spans, followed by greedy decoding to select a set of non-overlapping entities (Li et al., 2021; Fu et al., 2021). We describe these two steps in this section.

2.1 Span Classification

This step consists of enumerating all the spans from the input sequence and computing their representation using pre-trained transformers such as BERT. Following previous work (Lee et al., 2017; Luan et al., 2019), the representation s_{ij} of a span (i, j) of length k is computed by concatenating the representation of its left and right endpoint tokens (h_i and h_j respectively) along with a learned span width feature f_k. A 2-layer Multilayer Perceptron with *ReLU* activation is applied to the features to get the final span representation:

$$s_{ij} = \texttt{MLP}([h_i; h_j; f_k]) \tag{1}$$

Then, the span representation is fed into a linear layer (or an MLP) for span classification. A NER task with L entity types would have $L + 1$ labels since we allocate a `null` label for non-entity spans. The score of label y for a span (i, j) is computed as:

$$\phi(i, j, y) = w_y^T s_{ij} \tag{2}$$

where w_y is a learnable weight vector (we omit bias term for readability). These scores are further normalized using the softmax function.

The model is trained to minimize the negative log-likelihood of gold spans in the training set $\mathcal{T}$:

$$\mathcal{L}_{clf} = -\sum_{(i,j,y)\in\mathcal{T}} \log \frac{\exp\{\phi(i, j, y)\}}{\sum_{y'} \exp\{\phi(i, j, y')\}} \tag{3}$$

During inference, each span (i, j) is assigned the label $y(i, j) = \arg\max_y \phi(i, j, y)$ with score $k(i, j) = \max_y \phi(i, j, y)$. We call $\mathcal{C}$ the set of candidate entities which is the set of all spans assigned a label different from `null`. This set may contain overlapping spans which is not allowed in flat NER tasks, a decoding step is therefore required.

2.2 Maximum Weight Independent Set in Interval Graphs

An *overlap graph* over $\mathcal{C}$ is the graph G whose nodes are the elements of $\mathcal{C}$ and contains an edge between each pair of overlapping entities. This graph can also be called an *interval graph* since spans can be seen as intervals over their start and end positions. An *Independent Set (IS)* of the graph G is a set of nodes such that no two nodes in the set are joined by an edge. An independent set is said to be *maximal* if it is not properly contained in another independent set. Each node (i, j) in the graph is assigned a real number $r(i, j)$, the graph G is said to be a *weighted* graph. For each subset of nodes $\mathcal{S} \subseteq \mathcal{C}$, $\sum_{(i,j)\in\mathcal{S}} r(i, j)$ is called the weight of $\mathcal{S}$. A *Maximum Weight Independent Set (MWIS)* is an independent set such that its weight is maximum amongst all independent sets. Under this formulation, the decoding problem amounts to finding an MWIS in the graph G:

$$\hat{\mathcal{S}} = \arg\max_{\mathcal{S}\in\Psi(\mathcal{C})} \sum_{(i,j)\in\mathcal{S}} r(i, j) \tag{4}$$

where $\Psi(\mathcal{C})$, the set of all MIS of G.

Greedy Decoding Greedy decoding constructs an approximation to $\hat{\mathcal{S}}$ by iteratively adding the highest-scoring entity in C which does not overlap with any previously selected one. This algorithm has a complexity of $O(n \log n)$ with $n = |C|$.

In the next section we propose an exact alternative which uses a globally-normalized model.

Exact decoding The exact solution to Eq. (4) can be obtained by dynamic programming using an MWIS algorithm presented by Gupta et al. (1982); Hsiao et al. (1992). This algorithm has a linear time complexity $O(n)$ with n being the number of

nodes in the graph which is supposed to be sorted by interval endpoints (otherwise, it can be sorted in $O(n \log n)$ time. In practice, thenumber of nodes n is much lower than the input sequence length.

2.3 A Globally-Normalized MWIS Model

One way of estimating the weights $r(i, j)$ of the graph nodes is to use the scores produced by the local classifiers: $r(i, j) = k(i, j)$. In this section we propose to learn a dedicated probabilistic model of of MIS globally-normalized and learned to maximize the probability of the gold MIS.

The probability of an MIS is computed given by:

$$P(\mathcal{S}) = \mathcal{Z}^{-1} \exp\left\{ \sum_{(i,j)\in\mathcal{S}} r(i, j) \right\} \quad (5)$$

The unnormalized score of an MIS is still simply the sum of individual span weights where each is a linear projection of the span representation:

$$r(i, j) = w^T s_{ij} \quad (6)$$

where w is a parameter vector to be learned. The normalization constant is given by:

$$\mathcal{Z} = \sum_{\mathcal{S}\in\Psi(\mathcal{C})} \exp\left\{ \sum_{(i,j)\in\mathcal{S}} r(i, j) \right\} \quad (7)$$

While $\mathcal{Z}$, the partition function, can be ignored during inference, it has to be computed for learning as we use the negative log probability of the gold MIS as a loss function. The partition function can be computed efficiently using a modification to the dynamic program of the MWIS algorithm, however, in practice, we simply enumerate all MIS, which is feasible since the number of remaining spans is low. The enumeration can be done in time $O(n^2 + \beta)$ where n is the number of spans and β the sum of the numbers of spans of all enumerated sets (Leung, 1984; Liang et al., 1991).

During training, we modify the set $\mathcal{C}$, i.e. the output of the local classifier, so that (1) it contains all the gold spans, and (2) it does not contains spans that do not overlap with the gold spans. By doing this, we ensure that gold spans form an MIS in the overlap graph over $\mathcal{C}$. Finally, we use a multitask loss function that is the sum of the local classifier loss (Eq. (3)) and the global model loss.

3 Experiments

3.1 Setup

Baselines We compare our approach to a CRF tagger, the standard span-based model and the span-based model with Semi-Markov CRF. For all the models, we used pretrained transformers for token representation.

Datasets We evaluate our model on diverse NER datasets: TDM (Hou et al., 2021), Conll-2003 (Tjong Kim Sang and De Meulder, 2003), and OntoNotes 5.0 (Weischedel et al., 2013) for English data, and ACE05 for Arabic data (Walker et al., 2006). The details about the dataset can be found in the appendix A.1.

Evaluation metrics We evaluate the models using the exact matching between the predicted and true entities. We report the Precision, Recall and F1.

Hyperparameters For Conll-2003 and Ontonotes datasets we use `bert-base-cased` (Devlin et al., 2019) to produce contextual representation, for TDM we use SciBERT (Beltagy et al., 2019) and for Arabic ACE we use `bert-base-arabertv2` (Antoun et al., 2020). We use the base size, with 12 transformer layers, for all the models. We do not use any auxiliar embeddings (eg. *character embeddings*) for simplicity. All the models are trained with *Adam* optimizer (Kingma and Ba, 2017) with a learning rate of 2e-5, a batch size of 10 and a maximal epoch of 25. We keep the best checkpoint on the validation set for testing. We trained all the models in a server with V100 GPUs.

3.2 Results

The results of our experiments are shown in Table 1. We report the results for the four datasets using CRF, Semi-CRF, Standard and Global span-based models. For both Standard and Global models, we report the results obtained by using (cf. + Global lines) or not using decoding (cf. + Greedy lines).

Main results From Table 1, we can see that holistically, our global models with global decoding achieve the best results on most of the datasets (all except on OntoNotes). Moreover, Semi-CRF has the lowest score on all data, which may explain its low adoption over the years compared to the standard CRF.

Models	Conll-2003			OntoNotes 5.0			TDM			Arabic ACE		
	P	R	F	P	R	F	P	R	F	P	R	F
CRF	92.64	91.82	92.23	87.77	89.47	88.61	69.77	73.65	71.66	82.79	84.44	83.61
Semi-CRF	91.46	90.77	91.11	87.44	88.85	88.14	69.38	72.85	71.05	82.97	84.24	83.60
Standard	93.40	91.68	92.53	89.47	90.00	89.73	67.75	69.88	68.78	83.21	83.76	83.48
+ Greedy	93.82	91.40	92.60	90.43	89.04	89.73	75.12	67.82	71.26	83.73	83.56	83.64
+ Global	93.83	91.51	92.65	90.58	89.45	**90.01**	75.25	68.12	71.48	83.72	83.55	83.63
Global	94.84	90.72	92.73	89.05	89.77	89.41	63.30	72.75	67.53	83.54	83.65	83.60
+ Greedy	95.07	90.42	92.69	89.98	88.44	89.21	74.16	68.23	71.07	83.87	82.75	83.31
+ Global	95.11	90.52	**92.76**	90.18	88.85	89.51	75.55	70.34	**72.84**	84.14	83.35	**83.74**

Table 1: **Experimental results**. We report the average over three random seeds.

Global vs. Greedy decodings For both the span-based approaches, we can see that decoding generally improves F1 score performance and Precision while decreasing Recall. We explain this behavior by the fact that when using decoding, non-confident spans are removed, so Precision increases. However, some false negatives may be also removed, hence the slight decrease in recall. Moreover, for standard models, greedy and global decoding have similar performance, while for globally trained models, global decoding always has the best performance, which shows the effectiveness of our approach. Also, we can further observe on the Conll-2003, Arabic ACE and OntoNotes 5.0 datasets that greedy decoding can even decrease the performance of the model which may be an effect of the myopic bias.

4 Related Works

Approaches for NER Traditionally, NER tasks are designed as sequence labeling (Lample et al., 2016; Akbik et al., 2018), i.e., token-level classification. Recently, many approaches have been proposed that go beyond token-level prediction. For instance, some works have approached NER as question answering (Li et al., 2020) and others use sequence-to-sequence models (Yan et al., 2021; Yang and Tu, 2022). In this work, we focused on span-based methods (Liu et al., 2016; Sohrab and Miwa, 2018; Fu et al., 2021; Zaratiana et al., 2022; Corro, 2022) where all spans are enumerated and then classified into entity types.

Decoding for NER NER is a task for which a decoding algorithm must be applied to ensure that the model outputs are well trained. For example, CRF (Lafferty et al., 2001) has been proposed for sequence labeling and Semi-CRF for the span-based approach. Due to the low performance of Semi-CRF (Sarawagi and Cohen, 2005), researchers have proposed to train a local span-based method and use greedy decoding to guarantee non-overlapping entities for decoding. In this work, we propose exact/global decoding to produce a set of non-overlapping spans that maximize the global score to avoid the myopic bias of the greedy approach.

5 Conclusion

In this work, we proposed a new approach for span-based NER. During learning, our model maximizes the probability of the best segmentation while during inference, the final spans are selected according to a global score using dynamic programming. Our model mitigates the myopic bias of the greedy decoding of the standard span-based approach and it scores best on most datasets compared to other structured models such as CRF or Semi-CRF. For future work, it would be interesting to model the interaction between the spans to compute the global score.

6 Limitations

The main limitation of our model is that it is not suitable for recognizing nested named entities since the output structure is a set of non-overlapping spaces. Moreover, our model performed worse on the OntoNotes dataset: the cause may be due to some negative interference from our multitasking loss that makes learning difficult for large type sets. We will address these mentioned weaknesses in futur works.

Acknowledgments

This work is partially supported by a public grant overseen by the French National Research Agency (ANR) as part of the program Investissements d'Avenir (ANR-10-LABX-0083). This work was granted access to the HPC/AI resources of [CINES/IDRIS/TGCC] under the allocation 20XX-AD011013096 made by GENCI.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem M. Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *ArXiv*, abs/2003.00104.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text.

Caio Corro. 2022. A dynamic programming algorithm for span-based nested named-entity recognition in o(n^2). *ArXiv*, $abs/2210.04738$.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jinlan Fu, Xuanjing Huang, and Pengfei Liu. 2021. SpanNER: Named entity re-/recognition as span prediction. 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 7183–7195, Online. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

U. I. Gupta, D. T. Lee, and Joseph Y.-T. Leung. 1982. Efficient algorithms for interval graphs and circular-arc graphs. *Networks*, 12:459–467.

Aric A. Hagberg, Daniel A. Schult, and Pieter Swart. 2008. Exploring network structure, dynamics, and function using networkx.

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin, and Debasis Ganguly. 2021. TDMSci: A specialized corpus for scientific literature entity tagging of tasks datasets and metrics. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 707–714, Online. Association for Computational Linguistics.

Ju Yuan Hsiao, Chuan Yi Tang, and Ruay Shiung Chang. 1992. An efficient algorithm for finding a maximum weight 2-independent set on interval graphs. *Information Processing Letters*, 43(5):229–235.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

David S. Johnson. 1973. Approximation algorithms for combinatorial problems. *Proceedings of the fifth annual ACM symposium on Theory of computing*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Lingpeng Kong, Chris Dyer, and Noah A. Smith. 2016. Segmental recurrent neural networks. *CoRR*, abs/1511.06018.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Joseph Y.-T. Leung. 1984. Fast algorithms for generating all maximal independent sets of interval, circular-arc and chordal graphs. *J. Algorithms*, 5:22–35.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified mrc framework for named entity recognition.

Yangming Li, lemao liu, and Shuming Shi. 2021. Empirical analysis of unlabeled entity problem in named entity recognition. In *International Conference on Learning Representations*.

Y.D. Liang, S.K. Dhall, and S. Lakshmivarahan. 1991. On the problem of finding all maximum weight independent sets in interval and circular-arc graphs. In *[Proceedings] 1991 Symposium on Applied Computing*, pages 465–470.

Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, and Ting Liu. 2016. Exploring segment representations for neural segmentation models. In *IJCAI*.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. *ArXiv*, cmp-lg/9505040.

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. Association for Computational Linguistics.

Alexander M. Rush. 2020. Torch-struct: Deep structured prediction library.

Sunita Sarawagi and William W Cohen. 2005. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. 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.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.

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. *ArXiv*, abs/1910.03771.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various ner subtasks. In *ACL*.

Songlin Yang and Kewei Tu. 2022. Bottom-up constituency parsing and nested named entity recognition with pointer networks. In *ACL*.

Zhixiu Ye and Zhen-Hua Ling. 2018. Hybrid semi-Markov CRF for neural sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 235–240, Melbourne, Australia. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020a. Named entity recognition as dependency parsing. In *ACL*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020b. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2022. GNNer: Reducing overlapping in span-based NER using graph neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 97–103, Dublin, Ireland. Association for Computational Linguistics.

A Appendix

A.1 Dataset details

Conll-2003 (Tjong Kim Sang and De Meulder, 2003) is a dataset from the news domain that was designed for extracting entities such as Person, Location and Organisation. OntoNotes 5.0 (Weischedel et al., 2013) is a large corpus comprising various genres of text, including newswire, broadcast news, and telephone conversation. It contains a total of 18 different entity types, such as Person, Organization, Location, Product or Date. TDM (Hou et al., 2021) is a NER dataset that was recently published and it was designed for extracting Tasks, Datasets, and Metrics entities from Natural Language Processing papers. Arabic ACE is

the Arabic portion of the multilingual information extraction corpus, ACE 2005 (Walker et al., 2006). It includes texts from a wide range of genres, such as newswire, broadcast news, and weblogs. It contains a total of 7 entity types.

Dataset	Entity types	Train / Dev / Test
Conll-2003	4	14987 / 3466 / 3684
OntoNotes 5.0	18	48788 / 7477 / 5013
TDM	3	1000 / 500 / 500
Arabic ACE	7	2433 / 500 / 500

Table 2: Dataset statistics

A.2 Librairies

In this research, we used Pytorch (Paszke et al., 2019) to implement the models for its flexibility and ability to run on GPU machines. The pre-trained models were loaded from the HuggingFace Transformers library (Wolf et al., 2019), and some data processing was done using AllenNLP (Gardner et al., 2018). Our semi-CRF implementation is based on the pytorch-struct library (Rush, 2020). For evaluating the models, we adapted some code from the seqeval library (Nakayama, 2018). We employed Netwokx library (Hagberg et al., 2008) for graph processing in our decoding algorithm.