Injecting Knowledge Base Information into End-to-End Joint Entity and Relation Extraction and Coreference Resolution

Severine Verlinden*, **Klim Zaporojets***, **Johannes Deleu**, **Thomas Demeester**, **Chris Develder**
Ghent University – imec, IDLab
Ghent, Belgium
{first_name.last_name}@ugent.be

Abstract

We consider a joint information extraction (IE) model, solving named entity recognition, coreference resolution and relation extraction jointly over the whole document. In particular, we study how to inject information from a knowledge base (KB) in such IE model, based on unsupervised entity linking. The used KB entity representations are learned from either (i) hyperlinked text documents (Wikipedia), or (ii) a knowledge graph (Wikidata), and appear complementary in raising IE performance. Representations of corresponding entity linking (EL) candidates are added to text span representations of the input document, and we experiment with (i) taking a weighted average of the EL candidate representations based on their prior (in Wikipedia), and (ii) using an attention scheme over the EL candidate list. Results demonstrate an increase of up to 5% F1-score for the evaluated IE tasks on two datasets. Despite a strong performance of the prior-based model, our quantitative and qualitative analysis reveals the advantage of using the attention-based approach.

1 Introduction

Information extraction (IE) comprises several subtasks, e.g., named entity recognition (NER), coreference resolution (coref), relation extraction (RE). State-of-the-art results mainly report performance on single tasks, usually solving them on a sentence level (especially NER, RE). However, in practice, IE system decisions should be consistent on the document level, e.g., when processing news articles to automatically link entities (aside from potentially learning, e.g., new relations). Yet, the challenge of solving the tasks jointly on a document level has not received as much attention and remains hard (Durrett and Klein, 2014; Yao et al., 2019; Zaporojets et al., 2021).

*Equal contribution

On the other hand, it is well established that IE models benefit from incorporating background information of knowledge bases (KBs). Still, so far this has been shown from the perspective of solving individual tasks such as relation classification or entity typing (e.g., Peters et al. (2019); Liu et al. (2020)). Integrating KBs in joint models, realizing and analyzing the more complex end-to-end setting, has been left unexplored.

In terms of the nature of KBs adopted in IE, current approaches use either (i) structured knowledge *graphs* comprising `(subj,rel,obj)` triples, e.g., Wikidata (Yang and Mitchell, 2017; Han et al., 2018; Zhang et al., 2019), or (ii) *textual* descriptions, usually in hyperlinked documents, e.g., Wikipedia (Martins et al., 2019; Yamada et al., 2020). It has not been established to what extent KB-text and KB-graph entity representations complement each other in boosting IE performance.

We address both research gaps of (a) integrating KB information into a joint end-to-end IE model for solving named entity recognition, coreference resolution and relation extraction, and (b) analyzing what KB representation is more beneficial for IE, either *KB-graph* trained on Wikidata, or *KB–text* trained directly on Wikipedia. We particularly contribute: (i) a first span-based end-to-end architecture incorporating KB knowledge in a joint entity-centric setting, exploiting unsupervised entity linking (EL) to select KB entity candidates, (ii) exploration of prior- and attention-based mechanisms to combine the EL candidate representations into the model, (iii) assessment of the complementarity of KB-graph and KB-text representations, and (iv) consistent gains of up to 5% F1-score when incorporating KB knowledge in 3 document-level IE tasks evaluated on 2 different datasets.

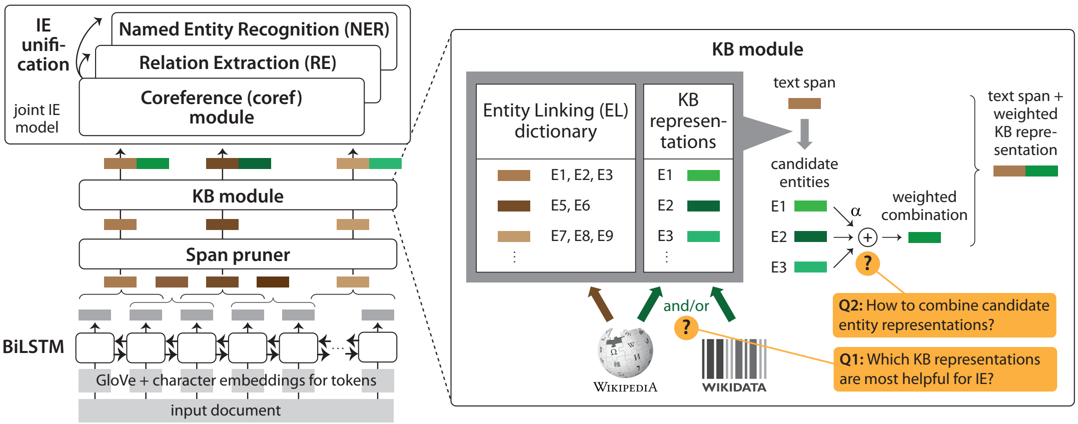

Figure 1: Joint information extraction (IE) model with addition of a knowledge base (KB) module.

2 Model

Figure 1 illustrates our model architecture. Input document tokens are represented using concatenated GloVe (Pennington et al., 2014) and character embeddings (Ma and Hovy, 2016) and pushed through a BiLSTM to obtain contextualized token representations, which are combined into spans. Similar to Luan et al. (2019); Zaporojets et al. (2021), a span pruner limits the number of spans for downstream modules. The *KB module* (§2.2) combines span representations with KB entity representations (§2.1), trained either on Wikidata (*KB-graph*) or Wikipedia (*KB-text*). The KB-enriched span representations then serve as input for joint predictions on downstream IE tasks (§2.3).

2.1 Entity Representations

We experiment with 3 possible entity representations: *KB-text*, *KB-graph*, and concatenating *both*.
KB-text: We follow Yamada et al. (2016) to obtain the entity representations using a skip-gram architecture (Mikolov et al., 2013a,b), training to jointly predict (i) the linked entities (through Wikipedia hyperlinks) given the target entity, and (ii) the neighboring words for a given entity hyperlink.
KB-graph: We adopt Joulin et al. (2017) to train the entity embeddings directly on Wikidata triples `(subj,rel,obj)` by optimizing a linear classifier to predict the `obj` entity from the `subj` entity and the relation type `rel`.

2.2 KB module

For a span s_i from token l to r, we obtain the representation $\mathbf{g}_i$ as input to the KB module by concatenating the respective hidden LSTM states $\mathbf{h}_l$ and $\mathbf{h}_r$, and an embedding ψ_{r-l} for the corresponding span width $r-l$:

$$\mathbf{g}_i = [\mathbf{h}_l; \mathbf{h}_r; \psi_{r-l}]. \quad (1)$$

We look up a given span s_i in a dictionary built from Wikipedia, to determine its candidate entities set[1] C_i, as well as the prior probability p_{ij} for each $c_{ij} \in C_i$, as per Yamada et al. (2016, §3).

To combine the KB candidates c_{ij}, we either use (i) a uniform average (*Uniform*), (ii) the prior weights p_{ij} (*Prior*), (iii) an attention scheme (*Attention*), or (iv) attention with prior information (*AttPrior*). The unnormalized attention scores for *Attention* and *AttPrior* are:

$$\Phi_{Attention}(s_i, c_{ij}, \mathrm{K}) = \mathcal{F}_A([\mathbf{g}_i; \boldsymbol{\xi}_\mathrm{K}(c_{ij})]) \quad (2)$$

$$\Phi_{AttPrior}(s_i, c_{ij}, \mathrm{K}) = \mathcal{F}_{AP}\big([\mathbf{g}_i; \boldsymbol{\xi}_\mathrm{K}(c_{ij}); p_{ij}]\big) \quad (3)$$

where $\mathrm{K} \in \{$*KB-text*, *KB-graph*, *both*$\}$ refers to the entity representations from §2.1, $\boldsymbol{\xi}_\mathrm{K}$ returns such representation for c_{ij}, and $\mathcal{F}_*$ is a feed-forward neural network (FFNN). The KB representation for span s_i is a weighted average of its candidates C_i:

$$\mathbf{e}_i^\mathrm{K} = \sum_{c_{ij} \in C_i} \alpha_{ij} \cdot \boldsymbol{\xi}_\mathrm{K}(c_{ij}) \quad (4)$$

where weights α_{ij} either are uniform $(1/|C_i|)$, the prior p_{ij}, or softmax-normalized attention scores (softmax over Φ from eq. (2) or eq. (3)). The concatenation $[\mathbf{g}_i; \mathbf{e}_i^\mathrm{K}]$ forms the KB-enriched representation for span s_i, as input for IE modules (§2.3).

2.3 Joint IE model

The joint IE model comprises 3 modules (Fig. 1) using the same KB-enriched representations $[\mathbf{g}_i; \mathbf{e}_i^\mathrm{K}]$,

[1] We limit this to the 16 most frequent ones.

Dataset	# Entity clusters	# Entity types	# Relations	# Relation types
DWIE	23,130	311	21,749	65
DocRED	98,610	6	50,503	96

Table 1: Dataset statistics.

and using a weighted combination of the 3 module losses to minimize during training. Note that NER and RE are framed as multi-label classification.

NER module: We use a FFNN on each span s_i to produce scores $\mathbf{\Phi}_{\text{NER}}(s_i) \in \mathbb{R}^{|L_E|}$, with L_E the set of possible entity types. At inference, we accept type $l \in L_E$ for span s_i if $\mathbf{\Phi}_{\text{NER}}(s_i)_l > 0$.

Coref module: We use the coreference scheme proposed by Lee et al. (2017), using a FFNN to produce scores $\Phi_{\text{coref}}(s_i, s_j)$: at inference time, the highest scoring antecedent of span s_j is then chosen (potentially s_j itself). Indeed, to allow for singletons we accept self-references (s_j, s_j) if NER predicts the span s_j to be an entity.

RE module: Similar to Luan et al. (2019, 2018), we use a FFNN to produce scores $\mathbf{\Phi}_{\text{RE}}(s_i, s_j) \in \mathbb{R}^{|L_R|}$ for each pair of spans (s_i, s_j), with L_R the set of relation types. We accept relation $l \in L_R$ for pair (s_i, s_j) if $\mathbf{\Phi}_{\text{RE}}(s_i, s_j)_l > 0$.

IE unification: Above modules make span level predictions. We obtain entity-centric predictions using the coref clusters, by assigning the union of predicted entity/relation types within a coref cluster to all its members, as do Zaporojets et al. (2021).

3 Experimental Setup

We evaluate our proposed models[2] on entity-centric multi-task datasets, summarized in Table 1: DWIE (Zaporojets et al., 2021) and DocRED (Yao et al., 2019). We report on coreference resolution (coref), NER and relation extraction (RE). For coref, we report the average of 3 common F1 scores, as implemented by Pradhan et al. (2014): MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998) and CEAF_e (Luo, 2005). Since we focus on entity-centric, document-level IE, for NER and RE we use *hard* metrics (Zaporojets et al., 2021) on the level of entity clusters (i.e., aforementioned coref clusters): predictions are counted as correct only if (i) all mentions (with exact boundary match) are present in the entity cluster, and (ii) the predicted entity type (for NER) or relation type between two clusters (for RE) is correct.

Our experiments address 2 main questions (see Fig. 1): **(Q1)** Which type of KB representation is most helpful for IE (*KB-text*, *KB-graph*, or *both*; see §2.1)? **(Q2)** Which weighting scheme to use for α (*Uniform*, *Prior*, *Attention*, *AttPrior*; see §2.2)?

4 Results

We summarize the comparison of various model choices for both DWIE and DocRED datasets in Table 2. First, looking into **(Q1)**, we note that including background information from *KB-graph* and *KB-text* significantly boosts performance compared to the *Baseline* without any KB. Additionally, our model outperforms the results from Zaporojets et al. (2021) (not listed in the table) by about 2 percentage points F1, using the same input (GloVe) representations. Furthermore, we observe a general improvement in results when combining *both* representations, suggesting that a (hyper)text corpus (Wikipedia) and a knowledge graph (Wikidata) embed complementary information for raising IE performance.

Deeper analysis reveals that adding KB representations mainly benefits performance for “rare” entity types: e.g., limiting the test set to entity types that occur $\leq$50 times in the training set for DWIE, compared to *Baseline*, F1 for NER goes up by $+$13.9 for *KB-both* with *AttPrior*, while the benefit gradually decreases for more frequently occurring entity types. For RE, we note that overall we also see a clear performance gain from adding KB information (e.g., +5.1% F1 for *both* KB sources with *AttProp* compared to *Baseline* for DWIE), yet the boost is not as clear for relations with fewer training instances. (The latter makes sense, since we inject KB representations of entities rather than explicitly also for relations; we leave studying adding relation embedding information for future work.)

Second, for **(Q2)**, we note that the *AttPrior* scheme is the overall winner among the different EL candidate weigthing schemes. We observed that in terms of ranking EL candidates, *Prior* performs quite well on DWIE — for 86.5% of entity mentions it assigns the highest score to the correct EL candidate, while *Attention* and *AttPrior* achieve it for 46.2%, resp. 77.2% of the mentions — which basically confirms that DWIE has a similar entity distribution as Wikipedia.[3] Yet, it seems necessary to include alternative candidates, and

[2]Code and models available at `https://github.com/klimzaporojets/e2e-kb-ie`.

[3]DWIE is a news article corpus.

KB Source	Setup	DWIE Coref	DWIE NER	DWIE RE	DocRED Coref	DocRED NER	DocRED RE
–	Baseline	90.0±0.2	71.7±0.5	47.0±1.4	81.9±0.3	68.5±0.3	23.5±0.6
KB-text	Uniform	90.7±0.2	73.5±0.5	48.5±1.1	82.9±0.1	70.7±0.2	24.5±0.3
	Attention	**90.7±0.3**	73.4±0.8	49.0±0.4	**83.4±0.1**	71.2±0.1	24.5±0.3
	AttPrior	90.7±0.3	73.7±0.6	**49.6±0.8**	83.2±0.2	**71.3±0.2**	24.8±0.4
	Prior	90.7±0.2	**73.8±0.5**	49.4±0.4	82.9±0.2	70.9±0.3	**25.3±0.4**
KB-graph	Uniform	91.0±0.3	73.6±0.4	48.0±1.2	83.3±0.2	71.1±0.2	24.9±0.2
	Attention	91.2±0.3	73.9±0.5	50.1±1.1	**83.7±0.1**	**71.6±0.1**	25.0±0.4
	AttPrior	**91.3±0.2**	**74.6±0.3**	**50.5±1.0**	83.5±0.3	71.5±0.2	25.1±0.2
	Prior	90.8±0.3	73.6±0.6	49.6±1.1	83.4±0.1	71.1±0.1	**25.2±0.2**
both (KB-graph + KB-text)	Uniform	91.1±0.1	74.1±0.5	49.3±0.5	83.5±0.1	71.3±0.2	24.8±0.1
	Attention	91.2±0.3	74.3±0.6	51.3±1.3	83.5±0.2	71.5±0.1	24.8±0.3
	AttPrior	**91.5±0.2**	**75.0±0.4**	**52.1±1.2**	**83.6±0.2**	**71.8±0.3**	**25.7±0.7**
	Prior	90.8±0.1	73.8±0.2	49.8±1.2	83.2±0.1	71.2±0.1	25.1±0.3

Table 2: Main results of the experiments in F1 scores grouped by the background KB source. We report Avg. F1 scores of MUC, B^3 and $CEAF_e$ for Coref, and hard F1 metrics for NER and RE. **Bold** font indicates the best results for each of the different *KB source* types. Additionally, the best overall results are underlined.

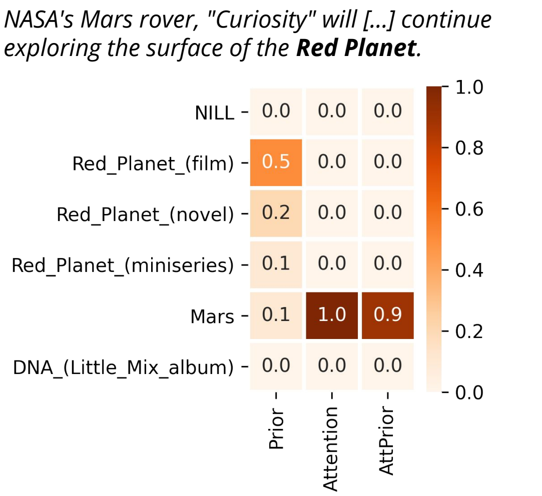

Figure 2: Illustration of EL candidate weighting: the α weights for top candidates for "Red Planet" from the example sentence at the top. Attention-based weighting (*Attention*, *AttPrior*) correctly identify the "Mars" entity, while the Wikipedia-based *Prior* fails, as most of Wikipedia's "Red Planet" links refer to the film.

the attention-based schemes thus can correct EL mistakes of *Prior*, as illustrated in Fig. 2. This correction leads to a resulting boost for the IE tasks as reported in Table 2. E.g., we found that for DWIE, looking at clusters with entity mentions for which *Prior* makes wrong EL predictions, the *AttPrior* weighting scheme retrieves +3.7% more of the gold standard annotated named entities (as opposed to just +0.6% in the clusters with correct *Prior* EL candidates). Perfecting the EL prediction would potentially boost IE performance even more.

5 Related Work

As stated earlier, we studied how to integrate (i) knowledge base information into IE, and particularly (ii) end-to-end IE combining multiple tasks (NER, relation extraction, coreference resolution), while (iii) taking an entity-centric perspective, i.e., focus on making consistent decisions on the document level. For (i), integrating KB into IE has been applied for individual tasks: relation classification (Poerner et al., 2020; Zhang et al., 2019; Yang and Mitchell, 2017), entity typing (Peters et al., 2019) and NER (Yamada et al., 2020). For (ii), recently span-based architectures (Lee et al., 2017; Luan et al., 2019; Wadden et al., 2019; Fei et al., 2020) have been proposed. Our work unifies the KB integration concept into such span-based IE system, in particular an entity-centric one (as per (iii)), building on Jia et al. (2019); Zaporojets et al. (2021). For the KB integration approach, we exploit entity representations trained on a hypertext corpus, as in (Yamada et al., 2016; Ganea and Hofmann, 2017; Yamada et al., 2020) or learnt from a knowledge graph (Yang and Mitchell, 2017; Han et al., 2018; Zhang et al., 2019). Our results show that both offer complementary value for IE. Similarly to our work, Yamada and Shindo (2019) also explore using an attention-weighted combination of entity representations, but they use it to build a full document representation (with mentions having the entities as candidates) for a text classification task. In contrast, our span-based attention model is able to "inject" knowledge in each of the mentions separately, for more fine-grained downstream IE tasks that are mention-dependent, e.g., coreference resolution, relation extraction and NER.

6 Conclusion

We propose an end-to-end model for joint IE (NER + relation extraction + coreference resolution) incorporating entity representations from a background knowledge base (KB), using a span-based system. We find that representations built from a knowledge graph and a hypertext corpus are complementary in boosting IE performance. To combine candidate entity representations for text spans, we explore various weighting schemes: an attention-based combination is successful in combining prior frequency information from a hypertext corpus with contextual information to identify the relevant entity, and achieves highest IE performance.

Acknowledgments

Part of the research leading to these results has received funding from (i) the European Union's Horizon 2020 research and innovation programme under grant agreement no. 761488 for the CPN project,[4] and (ii) the Flemish Government under the programme "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen".

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the 1998 International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference (LREC 1998)*, pages 563–566.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics (TACL 2014)*, 2:477–490.

Hao Fei, Yafeng Ren, and Donghong Ji. 2020. Boundaries and edges rethinking: An end-to-end neural model for overlapping entity relation extraction. *Information Processing & Management*, 57(6):102311.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 2619–2629.

[4] https://www.projectcpn.eu/

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *Proceedings of the 2018 Conference on Artificial Intelligence (AAAI 2018)*, volume 32.

Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level n-ary relation extraction with multiscale representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 3693–3704.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Maximilian Nickel, and Tomas Mikolov. 2017. Fast linear model for knowledge graph embeddings. *arXiv:1710.10881*.

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 (EMNLP 2017)*, pages 188–197.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: Enabling language representation with knowledge graph. In *Proceedings of the 2020 Conference on Artificial Intelligence (AAAI 2020)*, pages 2901–2908.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 3219–3232.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 3036–3046.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP 2005)*, pages 25–32.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 2016 Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1064–1074.

Pedro Henrique Martins, Zita Marinho, and André FT Martins. 2019. Joint learning of named entity recognition and entity linking. In *Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics: Student Research Workshop (ACL 2019, SRW)*, page 190.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of the 2013 International Conference on Learning Representations (ICLR 2013)*, pages 1–12.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 2013 International Conference on Neural Information Processing Systems (NIPS 2013)*, pages 3111–3119.

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 2014)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 43–54.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 2014 Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 30–35.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 1995 conference on Message understanding (MUC6, 1995)*, pages 45–52.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 5788–5793.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pages 6442–6454.

Ikuya Yamada and Hiroyuki Shindo. 2019. Neural attentive bag-of-entities model for text classification. In *Proceedings of the 2019 Conference on Computational Natural Language Learning (CoNLL 2019)*, pages 563–573.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 2016 SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016)*, pages 250–259.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in LSTMs for improving machine reading. In *Proceedings of the 2017 Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1436–1446.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 2019 Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 764–777.

Klim Zaporojets, Johannes Deleu, Chris Develder, and Thomas Demeester. 2021. DWIE: An entity-centric dataset for multi-task document-level information extraction. *Information Processing & Management*, 58(4):102563.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 3016–3025.