Joint Entity and Relation Extraction from Scientific Documents: Role of Linguistic Information and Entity Types

TYSS Santosh
Microsoft Corp.
Bengaluru, India
santoshtyss@gmail.com

Prantika Chakraborty
Indian Association for the Cultivation of Science
Kolkata, India
prantika.ch@gmail.com

Sudakshina Dutta
Indian Institute of Technology Goa
Ponda, India
sudakshina@iitgoa.ac.in

Debarshi Kumar Sanyal
Indian Association for the Cultivation of Science
Kolkata, India
debarshisanyal@gmail.com

Partha Pratim Das
Indian Institute of Technology Kharagpur
Kharagpur, India
ppd@cse.iitkgp.ac.in

ABSTRACT

Scientific articles contain various types of domain-specific entities and relations between them. The entities and their relations succinctly capture important information about the topic of the document and hence, they are crucial to the understanding and automatic analysis of the documents. In this paper, we aim to automatically extract entities and relations from a scientific abstract using a deep neural model. Given an input sentence, we use a pretrained transformer to produce contextual embeddings of the tokens which are then enriched with embeddings of their part-of-speech (POS) tags. A sequence of enriched token representations forms a span, and entities and relations are jointly learned over spans. Entity logits predicted by the entity classifier are used as features in the relation classifier. Our proposed model improves upon competitive baselines in the literature for entity and relation extraction on SciERC and ADE datasets.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Applied computing** → **Document management and text processing**.

KEYWORDS

entity extraction, relation extraction, deep learning, transformer, BERT, science IE

ACM Reference Format:
TYSS Santosh, Prantika Chakraborty, Sudakshina Dutta, Debarshi Kumar Sanyal, and Partha Pratim Das. 2021. Joint Entity and Relation Extraction from Scientific Documents: Role of Linguistic Information and Entity Types. In *EEKE@JCDL'21: 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Document, September 27–30, 2021, Online.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

EEKE@JCDL'21, Sep 27–30, 2021, Online

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The fast pace of modern scientific research and paper publication advances the state-of-the-art at a rapid rate. But it also makes it difficult for researchers to track all relevant publications even in their specialized domain. Therefore, machine learning algorithms are increasingly deployed to 'read' them at scale, extract useful information from them and organize the extracted information so that scholarly knowledge is more readily accessible to users. One important task in automatic analysis of research papers is the extraction of entities, i.e., entity mentions and their types, and the relation between entity pairs. These tasks are also called *named entity recognition (NER)* and *relation extraction (RE)*, respectively. For example, the following sentence **S1** contains two entities; we delineate the entity mentions by square brackets and the corresponding entity types as suffixes:

S1: The [generalized LR parsing]$_{\text{Method}}$ is enhanced in this [approach]$_{\text{Generic}}$.

The two entities above are related by the relation (or relation type) Used-for. NER and RE are useful for applications like knowledge graph construction [15], entity retrieval [1], semantic search [26], keyphrase extraction [20], question answering [21], summarization [6] and fact checking [28], and paper recommendation [8].

In this paper, we present a deep learning-based model to jointly extract entities and relations from abstracts in scientific papers. As a baseline, we use a recent model named SpERT [5] that uses a pretrained transformer [27] for the task. The transformer is first used to generate embeddings for the tokens in the abstract, then the embeddings of a *span* of tokens are combined into a span embedding on which a shallow entity classifier and a shallow relation classifier are applied to extract entities and relations, respectively. Many NLP tasks have benefited from the use of linguistic information such as part-of-speech tags [7], but they are less explored in deep neural models for NER and RE. For example, one can easily observe that entities tend to be noun phrases. Relations between entity pairs also appear to be related to the entity types. For example, we found that many entity pairs of type ('Method', 'Generic') are related by 'Used-for'. Therefore, we augment SpERT as follows: (1) we enrich the representations of the input tokens with linguistic information, in particular, *part-of-speech (POS) tags* of the words, and (2) include as inputs to the relation classifier the predicted *entity type logits*, or simply, *entity logits*. We call our model SpERT.PL[1]. Our model improves the state-of-the-art for entity and relation extraction on the benchmark datasets SciERC and ADE.

2 RELATED WORK

NER traditionally deals with the task of identifying names of organizations, people, geographic locations, currency, time and percentage expressions [14]. RE is an allied field of study that aims to identify a well-defined relationship between two or more named entities [19]. Deep learning is a popular technique for NER and RE. Recently, researchers have extended NER to include concrete (e.g., names of diseases) and abstract (e.g., gravity) entities in scientific documents. While early works used separate models to extract entities and relations, more recent approaches focus on joint extraction frameworks as they typically reduce inter-task error propagation and utilize the interconnection between NER and RE [30]. Many joint models [2, 3, 17, 32] predict BILOU tags (BILOU = 'beginning, inside, last, outside, unit') for tokens to identify entities. Use of BILOU tags preclude inclusion of a token in multiple entities. In contrast, span-based approaches like ours first construct spans of tokens and then label the spans with entity types, thereby allowing *overlapping entities* as a token can be part of multiple spans. SpERT [5], which is extended by this paper, uses a pretrained transformer (BERT [10] and its variants) to generate span representations from which entities and relations are extracted. Notably, Luan et al. proposed different models like BiLSTM network [15], dynamic span graph [16] and transformers [29] for scientific entity and relation extraction. SpERT, though simpler, outperforms them all. None of these approaches used linguistic information to construct span representations, or the predicted entity types for RE.

3 PROPOSED APPROACH

We use a pretrained transformer, a POS encoder, a fusion module, a shallow entity classifier and a shallow relation classifier. We assume the predefined set of entities is $\mathcal{E}$ and that of relations is $\mathcal{R}$.

Pretrained Transformer. The first layer of the transformer contains the WordPiece tokenizer [22] that splits the input sentence into a sequence of tokens $D = ([\text{CLS}], t_1, \cdots, t_n, [\text{SEP}])$. Note that the tokenizer may fragment a word into multiple subword tokens. For example, if the word "gpu" is absent in the tokenizer's dictionary, it may be split into two tokens: ["gp", "##u"]. [CLS] and [SEP] are special symbols. [CLS] captures the context of the whole sentence while [SEP] acts as a separator between adjacent sentences. The WordPiece tokens are passed through the inner layers of a pretrained transformer like BERT [10] to obtain an embedding sequence

$$(\mathbf{b}_{[\text{CLS}]}, \mathbf{b}_1, \cdots, \mathbf{b}_n, \mathbf{b}_{[\text{SEP}]}) = \text{Transformer}(D)$$

where each embedding vector $\mathbf{b} \in \mathbb{R}^{d_1}$ where d_1 is the embedding dimension.

POS Encoder. We use ScispaCy [18] to generate POS tags of the input sentence. ScispaCy is a Python NLP library for processing biomedical or scientific text. Since the WordPiece algorithm may split a word into many tokens, we assign the POS tag of the parent word to each subword token born of it. We use a dedicated embedding matrix to generate embeddings, each of dimension d_2, of the POS tags.

Fusion Module. For every token, the fusion module concatenates the BERT embedding of the token and the POS embedding of its POS tag. This produces enriched representations of the input sentence: $(\mathbf{c}_{[\text{CLS}]}, \mathbf{c}_1, \cdots, \mathbf{c}_n, \mathbf{c}_{[\text{SEP}]})$ where $\mathbf{c}_i \in \mathbb{R}^{d_1+d_2}$. Note that the POS embeddings of [CLS] and [SEP] tokens are not meaningful, and will not be used for further processing.

Entity Classifier. To detect entities, every sequence s of k ($\leq k_{\max} = 10$) consecutive tokens is considered, and their embeddings $(\mathbf{c}_i, \cdots, \mathbf{c}_{i+k-1})$ are max-pooled to form a vector

$$\mathbf{v}(s) = \text{maxpool}(\mathbf{c}_i, \mathbf{c}_{i+1}, \ldots, \mathbf{c}_{i+k-1}) \in \mathbb{R}^{d_1+d_2}$$

Long spans are unlikely to represent valid entities and so, span width is an important feature for entity classification. So a width embedding matrix is trained to contain an embedding $\mathbf{w}_k \in \mathbb{R}^{d_3}$ for a span of length k. The span width embedding $\mathbf{w}_k$ is concatenated with $\mathbf{v}(s)$ to form the entity representation:

$$\mathbf{e}(s) = \mathbf{v}(s)||\mathbf{w}_k \in \mathbb{R}^{d_1+d_2+d_3}$$

Finally, $\mathbf{b}_{[\text{CLS}]}$, which represents the sentence context, is concatenated with $\mathbf{e}(s)$ to obtain the vector

$$\mathbf{x}(s) = \mathbf{e}(s)||\mathbf{b}_{[\text{CLS}]} \in \mathbb{R}^{2d_1+d_2+d_3}$$

The POS tag of the [CLS] token is not meaningful, so we simply take the BERT embedding of the [CLS] token. The vector $\mathbf{x}(s)$ is passed through a shallow entity classifier, which is a single layer feed-forward neural network (FFNN) that outputs entity logits:

$$\mathbf{p}(s) = \mathbf{W}\mathbf{x}(s) + \mathbf{b} \in \mathbb{R}^{d_4}$$

where $d_4 = |\mathcal{E}| + 1$; "+1" is due to the 'null' entity $\emptyset$ that denotes the absence of entity. $\mathbf{W}, \mathbf{b}$ are the learnable weight matrix and bias of the FFNN, respectively. The logits $\mathbf{p}(s)$ are passed through a softmax function to predict the entity type.

Relation Classifier. Those spans that are classified as $\emptyset$ by the entity classifier are filtered out. For the remaining spans, the next task is to identify the relation between every pair of them. Consider a pair of spans (s_1, s_2) where s_1 occurs before s_2 in the input sentence. We assume relations to be asymmetric, so the relation between (s_1, s_2) may be different from that between (s_2, s_1). We take the representations, $(\mathbf{c}_i, \cdots \mathbf{c}_j)$, where $\mathbf{c}_i$ is the embedding of the first token following s_1 and $\mathbf{c}_j$ is that of the last token preceding s_2 in the sentence, and max-pool them:

$$\mathbf{v}(s_1, s_2) = \text{maxpool}(\mathbf{c}_i, \cdots, \mathbf{c}_j) \in \mathbb{R}^{d_1+d_2}$$

Next, the candidate relation from span s_1 to s_2 is encoded as

$$\mathbf{r}_{s_1 \to s_2} = \mathbf{e}(s_1)||\mathbf{v}(s_1, s_2)||\mathbf{e}(s_2)||\mathbf{p}(s_1)||\mathbf{p}(s_2) \in \mathbb{R}^{3d_1+3d_2+2d_3+2d_4}$$

where $\mathbf{p}(s_i) \in \mathbb{R}^{d_4}$ denotes the logits for span s_i. Finally, $\mathbf{r}_{s_1 \to s_2}$ is passed through using a single layer FFNN with sigmoid of size $|\mathcal{R}|$ and threshold α. As relations can be asymmetric, $\mathbf{r}_{s_2 \to s_1} = \mathbf{e}(s_2)||\mathbf{v}(s_1, s_2)||\mathbf{e}(s_1)||\mathbf{p}(s_2)||\mathbf{p}(s_1)$ is constructed and classified. The loss function of the *joint model* is the *sum* of the cross-entropy loss of the entity classifier and that of the relation classifier. The model

[1]Code is available here: https://github.com/dksanyal/SpERT.PL.

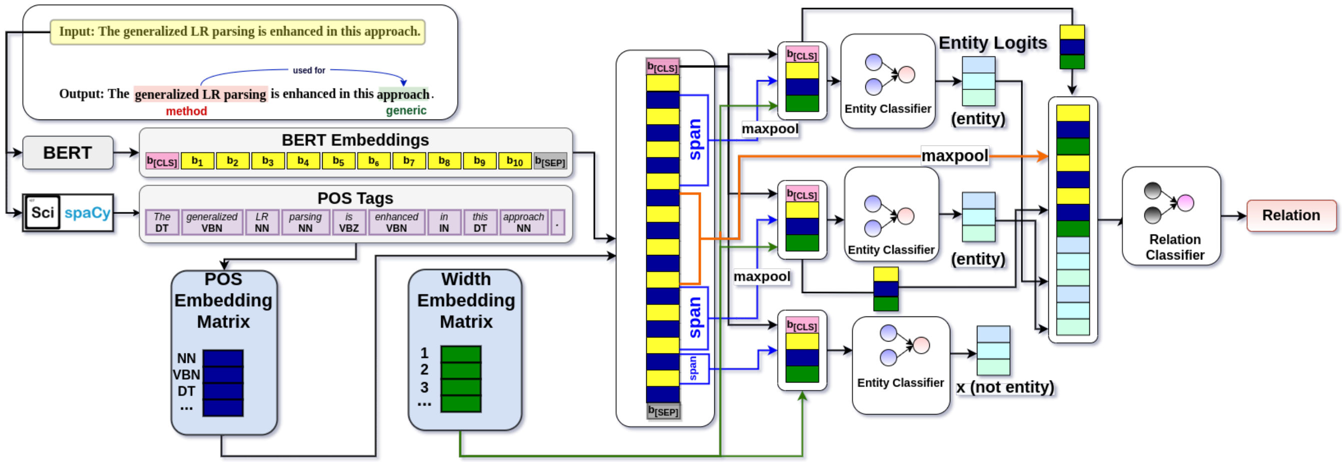

Figure 1: Architecture of our proposed model, SpERT.PL.

is trained in end-to-end fashion by backpropagation. The transformer is fine-tuned during training. To train the entity classifier, we use gold standard entity spans as positive examples and randomly sample non-entity spans from the input sentence as negative samples. For relation classification, like [5], we treat the ground truth relations as positive samples, and exploit the following as negative samples: (i) entity span pairs without any relation, and (ii) non-entity span pairs, both from within the same sentence. While the first strategy helps the model to label the relations accurately across all entities, the second strategy makes the relation classifier more robust to the errors in entity classification step.

4 EXPERIMENTS AND RESULTS

4.1 Datasets

4.1.1 SciERC. SciERC dataset [15] comprises 500 abstracts of AI papers; includes 6 scientific entities: Task, Method, Metric, Material, Other-Scientific-Term, and Generic, and 7 relation types: Compare, Conjunction, Evaluate-For, Used-For, Feature-Of, Part-Of, and Hyponym-Of, in a total of 2,687 sentences. The official split has 3 parts: train (1861 sentences), dev (275 sentences) and test (551 sentences). However, like [5], we use (train + dev) for training as we do not perform hyperparameter tuning.

4.1.2 ADE. ADE dataset [9] consists of 4272 sentences and 6,821 relations extracted from medical reports. It contains a single relation type Adverse-Effect and the two entity types Adverse-Effect and Drug. Due to absence of an official split, we conduct a 10-fold cross validation like the other existing works. We considered 2 cases: (1) *with overlap:* all entities and relations are retained; (2) *without overlap:* Around 120 relations with overlapping entities (e.g., 'lithium' is a drug included in 'lithium intoxication') are removed.

4.2 Implementation

We use SciBERT [4] as the pretrained transformer for SciERC. We experiment with both SciBERT and BioBERT [11], separately, for ADE. The dimension of span width embedding (d_2) and that of POS embedding (d_3) are both 25. We did not tune the hyperparameters but use those in [5]. Specifically, we train the model for 20 epochs using Adam optimizer with linear warmup, linear decay and peak learning rate $5e-5$; set the threshold for sigmoid activation in relation classifier to $\alpha = 0.4$; and sample 100 negative samples for both the tasks. We use a training batch size of 10.

4.3 Evaluation Metrics

For every span of text (of length $k \leq 10$), the proposed model performs NER in which an *entity* is considered correct if the entity type and span are predicted correctly. Given two text spans, the model also performs RE. Following [2, 5], we define its correctness in two ways: (1) **Strict RE**: the relation type and the two related entities (i.e., both span and entity type) must be correct. (2) **Boundaries RE**: the relation type and the spans of the two related entities must be correct. Following the literature [5], we report only micro-average for SciERC, both micro- and macro-average for ADE, and only strict RE for ADE. Since only one relation occurs in ADE, averaging method for RE does not matter.

4.4 Results

4.4.1 Performance on SciERC. We report the performance of SpERT.PL on SciERC dataset in Table 1. Due to the large variance in the measured values for SpERT.PL – a similar observation is made by Taillé et al. [23] for SpERT – we report the mean and standard deviation of the scores from 15 observations for SpERT.PL. Compared to SpERT (that also uses SciBERT), there is a slight fall in precision but an increase in recall and F1-score for all the 3 tasks. SpERT.PL also outperforms other joint entity-relation extraction approaches like SciIE [15], DyGIE [16] and DyGIE++ [29] and a recent pipelined approach called PURE [33], even when PURE uses cross-sentence context to build better contextual representations of spans.

4.4.2 Performance on ADE. Table 2 shows SpERT.PL outperforms SpERT and establishes new state-of-the-art results for ADE. Notably, using BioBERT as a pretrained transformer in SpERT.PL generally produces higher performance than using SciBERT. This is not surprising as BioBERT is pretrained entirely on biomedical papers

Table 1: Performance on SciERC. Micro-average scores are reported.

Model	NER			Boundaries RE			Strict RE		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SpERT.PL (SciBERT)	69.82 (±0.44)	**71.25** (±0.51)	**70.53** (±0.37)	51.94 (±0.72)	**50.62** (±0.94)	**51.25** (±0.55)	39.94 (±0.9)	**38.98** (±0.89)	**39.41** (±0.77)
SpERT [5]	**70.87**	69.79	70.33	**53.4**	48.54	50.84	**40.51**	36.82	38.57
DyGIE++ [29]	-	-	67.5	-	-	48.4	-	-	-
DyGIE [16]	-	-	65.2	-	-	41.6	-	-	-
SciIE [15]	67.2	61.5	64.2	47.6	33.5	39.3	-	-	-
PURE (Single sentence) [33]	-	-	66.6	-	-	48.2	-	-	35.6
PURE (Cross sentence) [33]	-	-	68.9	-	-	50.1	-	-	36.8

Table 2: Performance on ADE. * indicates that the corresponding paper does not state if NER performance is micro-average or macro-average, though we use the micro-average columns for these cases.

	Model	NER (Micro-average)			NER (Macro-average)			Strict RE		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
With Overlap	**SpERT.PL** (BioBERT)	**90.05**	91.69	**90.86**	**90.33**	**92.03**	**91.17**	**80.11**	**84.18**	**82.03**
	SpERT.PL (SciBERT)	89.15	**91.73**	90.4	89.43	91.96	90.72	78.54	83.98	81.16
	SpERT [5]	88.69	89.2	88.95	88.99	89.59	89.28	77.77	79.96	78.84
Without Overlap	**SpERT.PL** (BioBERT)	**90.44**	**91.3**	**90.86**	**90.66**	**91.64**	**91.14**	**80.33**	**84.57**	**82.39**
	SpERT.PL (SciBERT)	89.89	91.16	90.52	89.15	90.75	89.94	79.04	84.39	81.62
	CMAN [31]	-	-	-	-	-	89.4	-	-	81.14
	Table Sequence [30]	-	-	-	-	-	89.7	-	-	80.1
	SpERT [5]	89.02	88.87	88.94	89.26	89.26	89.25	78.09	80.43	79.24
	Relation-Metric [25]*	86.16	88.08	87.1	-	-	-	77.36	77.25	77.29
	Multi-head + AT [2]	-	-	86.7	-	-	-	-	-	75.52
	Multi-head [3]	84.72	88.16	86.4	-	-	-	72.1	72.24	74.58
	BiLSTM + SDP [12]*	82.7	86.7	84.6	-	-	-	67.5	75.8	71.4
	CNN + Global features [13]*	79.5	79.6	79.5	-	-	-	64	62.9	63.4

Table 3: Ablation study of SpERT.PL on SciERC.

Model	NER			Boundaries RE			Strict RE		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SpERT.PL (SciBERT)	69.87	71.47	70.66	52.06	51.26	51.65	40.49	39.87	40.18
− POS embeddings	69.52	70.66	70.09	51.64	50.82	51.23	39.59	38.95	39.26
− entity logits	69.41	70.49	69.96	51.34	49.66	50.48	39.51	38.23	38.86

while SciBERT also includes computer science papers. When overlapping entities are included, SpERT.PL records gains of 1.91% in micro-average F1-score for NER, 1.89% in macro-average F1-score for NER, and 3.19% in F1-score for strict RE over the second best performer. When overlapping entities are excluded, the corresponding gains are 1.92%, 1.44% and 1.25%, and SpERT.PL not only outperforms SpERT but also more efficient approaches like CMAN [31] and Table Sequence [30]. Both SpERT.PL and SpERT score over many other recent approaches like [2, 3, 12, 13, 25]. Note that only SpERT and SpERT.PL allow non-overlapping entities.

4.5 Ablation Study

The ablation study in Table 3 shows the role of POS embeddings and entity logits on the final classification scores. The reported figures for each model are the average of 3 runs. We observe that removing POS embeddings from SpERT.PL causes a drop of 0.57%, 0.42%, and 0.92% in F1-score for NER, boundaries RE, and strict RE, respectively. The drop is not substantial as BERT already captures the grammatical features of the input [24]. Removing entity logits from SpERT.PL reduces F1-score by 0.7%, 1.17%, and 1.32% for NER, boundaries RE, and strict RE, respectively. Thus, entity logits have more pronounced effect on relation extraction, more so when the associated entities must be correctly identified in span and type.

5 CONCLUSION

We proposed a deep neural model called SpERT.PL for entity and relation extraction from scientific documents. We found that POS information and predicted entity logits boost the classification performance. In future, we will explore if dependency parse of the input sentences can further improve the classification accuracy.

ACKNOWLEDGMENTS

This work is supported by research grant from Department of Science and Technology, Government of India at IACS, Kolkata and *National Digital Library of India Project* sponsored by the Ministry of Education, Government of India at IIT Kharagpur.

REFERENCES

[1] Krisztian Balog. 2018. *Entity-oriented search.* Springer Nature.

[2] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Adversarial training for multi-context joint entity and relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2830–2836.

[3] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications* 114 (2018), 34–45.

[4] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: Pretrained Language Model for Scientific Text. In *EMNLP*. arXiv:arXiv:1903.10676

[5] Markus Eberts and Adrian Ulges. 2019. Span-based Joint Entity and Relation Extraction with Transformer Pre-training. *arXiv preprint arXiv:1909.07755* (2019).

[6] Shai Erera, Michal Shmueli-Scheuer, Guy Feigenblat, Ora Peled Nakash, Odellia Boni, Haggai Roitman, Doron Cohen, Bar Weiner, Yosi Mass, Or Rivlin, et al. 2019. A Summarization System for Scientific Documents. 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): System Demonstrations*. 211–216.

[7] Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. 2006. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

[8] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[9] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of biomedical informatics* 45, 5 (2012), 885–892.

[10] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. 4171–4186.

[11] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.

[12] Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. 2017. A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics* 18, 1 (2017), 1–11.

[13] Fei Li, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016. Joint Models for Extracting Adverse Drug Events from Biomedical Text. In *IJCAI*, Vol. 2016. 2838–2844.

[14] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[15] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *EMNLP*. 3219–3232.

[16] 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, Volume 1 (Long and Short Papers)*. 3036–3046.

[17] Makoto Miwa and Mohit Bansal. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1105–1116.

[18] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task, BioNL@ACL 2019*. 319–327.

[19] Sachin Pawar, Girish K Palshikar, and Pushpak Bhattacharyya. 2017. Relation extraction: A survey. *arXiv preprint arXiv:1712.05191* (2017).

[20] Tokala Yaswanth Sri Sai Santosh, Debarshi Kumar Sanyal, Plaban Kumar Bhowmick, and Partha Pratim Das. 2020. DAKE: Document-Level Attention for Keyphrase Extraction. In *ECIR*. Springer, 392–401.

[21] Mourad Sarrouti and Said Ouatik El Alaoui. 2020. SemBioNLQA: a semantic biomedical question answering system for retrieving exact and ideal answers to natural language questions. *Artificial intelligence in medicine* 102 (2020), 101767.

[22] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1715–1725.

[23] Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. Let's Stop Incorrect Comparisons in End-to-end Relation Extraction! *arXiv preprint arXiv:2009.10684* (2020).

[24] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*. 4593–4601.

[25] Tung Tran and Ramakanth Kavuluru. 2019. Neural metric learning for fast end-to-end relation extraction. *arXiv preprint arXiv:1905.07458* (2019).

[26] Vu Tran, Van-Hien Tran, Phuong Nguyen, Chau Nguyen, Ken Satoh, Yuji Matsumoto, and Minh Nguyen. 2021. CovRelex: A COVID-19 Retrieval System with Relation Extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL): System Demonstrations*. 24–31.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*.

[28] Nikhita Vedula and Srinivasan Parthasarathy. 2021. FACE-KEG: Fact checking explained using knowledge graphs. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 526–534.

[29] 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 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5788–5793.

[30] Jue Wang and Wei Lu. 2020. Two Are Better than One: Joint Entity and Relation Extraction with Table-Sequence Encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1706–1721.

[31] Shan Zhao, Minghao Hu, Zhiping Cai, and Fang Liu. 2020. Modeling Dense Cross-Modal Interactions for Joint Entity-Relation Extraction. In *IJCAI*. 4032–4038.

[32] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1227–1236.

[33] Zexuan Zhong and Danqi Chen. 2021. A Frustratingly Easy Approach for Entity and Relation Extraction. In *NAACL*.