

Approaches to Relation Extraction for Nested Named Entities

A. V. Yandutov^{1*} and N. V. Loukachevitch^{1**}

(Submitted by E. K. Lipachev)

¹*Lomonosov Moscow State University, Moscow, 119991 Russia*

Received November 18, 2022; revised November 29, 2022; accepted December 11, 2022

Abstract—The paper studies approaches to extract relations between so called nested named entities that is named entities, which can contain other named entities. We argue that most datasets annotated with nested named entities do not have relation markup. Therefore some state-of-the-art relation extraction models cannot extract relations between nested or overlapping entities because of the architecture or input format. We use recently created NEREL corpus for Russian annotated with nested named entities and relations between them.

DOI: 10.1134/S1995080223010456

Keywords and phrases: *relation extraction, nested named entities, russian language.*

1. INTRODUCTION

Relation Extraction (RE) between named entities is an important subtask of information extraction. Information extraction is a key step to construct knowledge bases from a large number of unstructured texts, which benefits many natural language processing applications, such as question answering and information retrieval.

Most datasets annotated with named entities contain a simplified named entity markup that does not assume that a named entity can be nested within another named entity (so-called flat entities). For example, *Moscow State University* is an organization, the name of which contains another named entity (*Moscow*). In flat named entity annotation schemes, only longer or shorter entities should be annotated, which leads to loss of information about mentioned named entities.

Currently, many new datasets are annotated using schemes allowing nested or intersecting named entities [1–3]. But most such current datasets with nested named entities are not annotated with relations. Therefore, relations between nested and overlapping entities have not been considered by state-of-the-art relation extraction methods based on machine learning. This simplification can cause information loss in extracting relations.

Recently, new corpus NEREL [4] annotated with nested named entities and relations between them was annotated for Russian. This gives new possibilities to study relation extraction methods applied to nested named entities. In this paper, we compare various approaches to extracting relations that can include nested entities and do experiments for the Russian language using the NEREL corpus.

*E-mail: yandutov99@gmail.com

**E-mail: louk_nat@mail.ru

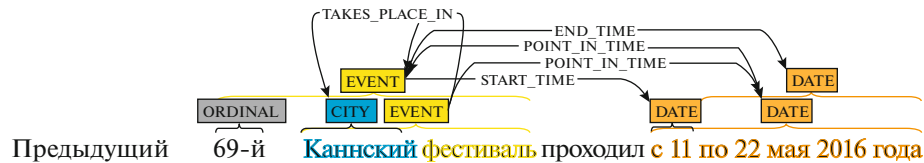


Fig. 1. Relations between nested entities in sentence “The previous 69th Cannes Film Festival took place from 11 to 22 May 2016”.

2. RELATED WORK

Early supervised relation extraction approaches based on classical machine-learning methods such as SVM, CRF etc, mainly relied on various hand-crafted features such as syntax groups or trees, semantic types of entities, distance between entities, etc [5]. Recently, neural network models have been widely used for relation extraction. These neural models can accurately capture textual relations without explicit linguistic features [6–8].

Most last approaches to relation extraction utilize transformer architecture [9, 10]. In [9], the authors propose schemes of using encoder representation from transformer (BERT) for the relation extraction task. To further improve the attention performance, some works incorporate information from knowledge bases [11]. More sophisticated mechanisms, such as reinforcement learning [12], generative neural networks [10] and span-prediction based approach, similar to question-answering [13], have also been adapted for RE recently. The OpenNRE package [14] provides several implemented methods for relation extraction including CNN-based and BERT-based models. However, most models do not consider extracting relations between nested entities because of absence of training data annotated with relations between nested entities. Some of them, due to their architecture or encoding method, cannot fully extract relations between such entities.

For Russian, there are several datasets annotated with relations [15–17]. Besides NEREL, only one dataset, FactRuEval [16], contains nested named entities (2 levels of nestedness) and relations between them. But the FactRuEval dataset is too small to be used for training of contemporary neural network models, it contains about 1 thousand annotated relations. In 2020, the RuReBus evaluation was organized devoted to relation extraction in Russian [17]. The best results in this evaluation were obtained using the R-BERT model [9].

3. DATASET

Supervised models for RE require adequate amounts of annotated data for their training. It is time-consuming to manually label large-scale training data.

NEREL [4] is a dataset in Russian for named entity recognition and relation extraction based on Russian Wikinews documents. NEREL is much larger than the existing Russian datasets: today it contains 933 documents annotated with 29 entity and 49 relations types. NEREL also contains an event annotation involving named objects and their roles in the events.

An important distinction of NEREL from the previous datasets is the annotation of nested named entities (up to 6 levels of nestedness), as well as relations within nested entities and at sentence and the document level. NEREL allows developing new models that can extract relations between nested named entities as well as relations at both the sentence and document levels. Fig. 1 shows an example of nested entities and relations between them.

All the annotated relations can be subdivided into cross-sentence (24% of the total number of relations) in which the subject and object are in different sentences and within sentence relations (76% of the total number of relations). Among relations within a single sentence, three types of relations can be distinguished:

1. traditional relations between entities, which are located separately, as “69-ый-Каннский фестиваль” (69th Cannes Film Festival) and “с 11 по 22 мая 2016 года” (from 11 to 22 May 2016) in Fig. 1—further, external relations (53.9%);

2. nested relations, i.e., relations within the longest span of a single nested entity, as “69-й” (69th) and “фестиваль” (festival) within “69-ый Каннский фестиваль” (69th Cannes Film Festival) in Fig. 1 (15%);
3. relations crossing entity boundaries, i.e., relations between an external named entity and an internal entity within a nested named entity—further cross-entity relations (7.1%).

Tables 1 and 2 presents with frequencies of relations within sentence, between nested and intersecting entities. The cross-entity relations can be illustrated as follows: in the sentence “Barack Obama is a member of the Democratic party” external entity *Barack Obama* has the IDEOLOGY_OF relation to entity *Democratic*, which is an internal IDEOLOGY entity in the longer entity *Democratic party*.

4. APPROACHES TO REPRESENTATION OF ENTITIES IN RELATION EXTRACTION TASK

The representation of nested named entities requires special formats. Some traditional formats for named entity datasets do not allow describing nestedness. For example, well-known TACRED dataset used for evaluating relation extraction models contains only flat entities represented at one level. nested named entities in such a format. The example below shows that internal entity *Moscow* within longer entity *Moscow State University* cannot be represented in the TACRED format because each token of a sentence can corresponds only to a single entity label (see “stanford-ner” line):

```
{
  relation: "LOCATED_IN"
  token: [ 0: "Moscow", 1: "State", 2: "University", 3: "was",
          4: "founded", 5: "in", 6: "1755"]
  stanford-ner: [0: "B-ORGANIZATION", 1: "I-ORGANIZATION", 2: "I-
ORGANIZATION", 3: "O", 4: "O", 5: "O", 6: "B-DATE"]
  subj_start: 0
  subj_end: 3
  obj_start: 0
  obj_end: 1
  subj_type: "ORGANIZATION"
  obj_type: "LOCATION"
  id: "115778"
}
```

There are several approaches for representing entities in relation extraction tasks. Some representation formats implemented in state-of-the-art methods of relation extraction do not allow representing relations between nested or intersecting named entities. Even if the dataset format allows representing nested named entities, the representation of nested or intersecting relations can be problematic in some relation representation models. Below we consider main formats for representing entities and relations in relation extraction models.

4.0.1. Position features. In the CNN-based approach [6], the authors appended a position feature vector to each word vector, i.e., the distance from the word to the two nominals. For example, in sentence *[People]₀ have₁ been₂ moving₃ back₄ into₅ [downtown]₆*.

The relative distances of *moving* to *people* and *downtown* are 3 and -3 , respectively. So for this word we can add feature vector $[-3, 3]$, which allows representing subject (*People*) and object (*downtown*) of the relation. Since encoding the subject and object is independent, there are no problems for the case of intersecting entities.

4.0.2. Entity masking. Entity masking schema [18, 19] replaces the subject and object entities by some masks like their NER tags such as

“[CLS]/[SUBJ-PER] was born in [OBJ-LOC], Michigan, . . .”

To predict an entity class, a linear classifier is added on the top of the [CLS] token in BERT-like model [20] to predict the relation type. This approach helps to provide the model with information about the

Table 1. Relation types in NEREL by the categories described in Section 3

	Relation	Total	Insentence	External	Nested	Cross
1	WORKPLACE	4702	4204	828	2035	1341
2	ALTERNATIVE_NAME	4479	892	662	29	201
3	WORKS_AS	4476	3576	3455	21	100
4	PARTICIPANT_IN	4165	3344	3056	128	160
5	POINT_IN_TIME	1950	1835	1661	104	70
6	TAKES_PLACE_IN	1904	1730	1409	182	139
7	HEADQUARTERED_IN	1883	1777	268	1410	99
8	ORIGINS_FROM	1742	1579	1124	393	62
9	LOCATED_IN	1485	1146	773	236	137
10	AGENT	1354	1192	1181	3	8
11	AGE_IS	1007	848	828	10	10
12	PRODUCES	755	549	385	142	22
13	HAS_CAUSE	743	578	556	14	8
14	AWARDED_WITH	584	409	406	0	3
15	PART_OF	519	480	45	409	26
16	IDEOLOGY_OF	430	386	188	124	74
17	MEMBER_OF	426	314	202	62	50
18	SUBEVENT_OF	426	265	209	51	5
19	CONVICTED_OF	404	283	280	0	3
20	KNOWS	393	292	290	0	2
21	INANIMATE_INVOLVED	383	291	285	4	2
22	SUBORDINATE_OF	359	326	114	148	64
23	MEDICAL_CONDITION	357	242	235	0	7
24	PARENT_OF	337	210	207	2	1
25	PLACE_RESIDES_IN	312	227	205	11	11
26	FOUNDED_BY	240	197	106	82	9
27	OWNER_OF	236	196	99	81	16
28	ABBREVIATION	233	157	99	14	44
29	ORGANIZES	206	166	140	16	10
30	PENALIZED_AS	197	146	142	0	4
31	SPOUSE	191	148	143	4	1
32	DATE_OF_CREATION	169	151	129	21	1
33	DATE_OF_BIRTH	161	148	147	1	0

type of entity. Also the benefit of using masking entities is that we can keep the model from overfitting by entities [21].

But the scheme is not suitable for relations between nested entities, since if one entity is inside

Table 2. Relation types in NEREL by the categories described in Section 3 (continuation of Table 1)

	Relation	Total	Insentence	External	Nested	Cross
34	RELIGION_OF	155	122	28	77	17
35	DATE_OF_DEATH	155	137	123	0	14
36	AGE_DIED_AT	142	115	115	0	0
37	PLACE_OF_BIRTH	142	121	121	0	0
38	SCHOOLS_ATTENDED	138	67	65	0	2
39	SIBLING	120	98	94	4	0
40	PLACE_OF_DEATH	109	94	92	0	2
41	CAUSE_OF_DEATH	82	61	61	0	0
42	START_TIME	75	65	50	3	12
43	DATE_FOUNDED_IN	68	64	62	1	1
44	INCOME	64	36	34	0	2
45	RELATIVE	61	44	42	1	1
46	EXPENDITURE	50	34	33	0	1
47	PRICE_OF	49	41	35	5	1
48	END_TIME	46	44	30	3	11
49	DATE_DEFUNCT_IN	9	7	7	0	0

another, then such a mask closes also the nested entity. For example, in relation between “*Moscow*” and “*Moscow State University*” from sentence “*Moscow state university was found in 1755.*” the mask [SUBJ-ORG] completely covers the internal entity: “*[SUBJ-ORG] was found in 1755.*”

The approach is used in some well-known models for relation extraction such as SpanBERT [18, 19].

4.0.3. Entity markers. The approach [22] is to place tags on the borders of entities. The schema modifies the last example with nested entities to the following form

“*<E1> <E2> Moscow <E2> State University <E1> was found in 1755.*”

This format allows the use in the task for extracting nested relations. Markers can also contain information about entities adding information into the model. The scheme works better by performance metrics than masking, but it can have overfitting on entities.

5. METHODS FOR EXTRACTING RELATIONS BETWEEN NESTED NAMED ENTITIES

5.1. Separate Extraction of Internal and External Relations

To solve the problem using state-of-the-art models that do not support nested entities, as a basic approach, it was proposed to divide the problem into two subtasks

1. Extracting external relations

Such relations are the majority among in-sentence relations in the NEREL dataset. These relations are the most complex, but this subtask is solved by most of the published models in the field of Relation Extraction, so large state-of-the-art models are used for this subproblem.

2. Extracting nested relations using separate model

The following hypotheses for nested relations have been proposed:

- Subject and object of internal relation are located within external entity, therefore, an external context is not needed to define such relation.

- Relations within an external entity depend mainly on the subject and the object. There is a simpler nature of connections between them, they are less diverse compared to external relations

5.2. Joint Extraction of Relations

The approach assumes that all relations will be extracted by a single model. It was shown above that the using position embeddings and entity markers scheme for representing allow processing overlapping entities. For the analysis, we chose the entity marking approach. Since it is used in the latest models and allows representing additional information such as entity types.

We found that in some architecturally correct models, the encoding implementation leads to errors when encoding nested objects. For example, in the considered implementation of BERT adaptation for RE from OpenNRE [14]. Sentence “*Moscow State University was found in 1755.*” with relation between entities “*Moscow State University*” and “*Moscow*” is converted into the format

“ $\langle E2_{start} \rangle Moscow \langle E2_{end} \rangle \langle E1_{start} \rangle Moscow State University \langle E1_{end} \rangle was found in 1755.$ ”

We noticed that the words of entities are mistakenly duplicated in the case of entities overlapping. It leads to changes in the distribution of the input texts, affecting the quality of the model. To conduct experiments, this module has been improved.

In OpenNRE, relations between entities are encoded as follows. First, entities are formed in the format

“ $\langle E1_{start} \rangle Entity_1 \langle E1_{end} \rangle$ ” and “ $\langle E2_{start} \rangle Entity_2 \langle E2_{end} \rangle$.”

Then, they were sequentially concatenated with the remaining parts of the sentence to the format

“ $[CLS]' + sent_0 + ent_0 + sent_1 + ent_1 + sent_2 + [SEP]'$ ”

In the scheme $sent_0$ is the text fragment before the first entity, $sent_1$ —the text fragment between entities, $sent_2$ —the fragment after the second entity.

The algorithm has been improved as follows. The pairs below were taken and sorted by the first element:

$(ent1_start_token_position, \langle E1_{start} \rangle)$

$(ent1_end_token_position, \langle E1_{end} \rangle)$

$(ent2_start_token_position, \langle E2_{start} \rangle)$

$(ent2_end_token_position, \langle E2_{end} \rangle)$

For example, for the above-mentioned sentence “*Moscow State University was found in 1755.*” we get

$(0, \langle E1_{start} \rangle)$

$(0, \langle E2_{start} \rangle)$

$(1, \langle E2_{end} \rangle)$

$(3, \langle E1_{end} \rangle)$

Then, the text was cut into positions and entity tags were inserted in the corresponding positions. This made it possible to correctly represent entities without duplication:

“ $\langle E1_{start} \rangle \langle E2_{start} \rangle Moscow \langle E2_{end} \rangle State University \langle E1_{end} \rangle was found in 1755.$ ”

This format improves the representation of relations between intersecting or nested entities.

Table 3. Result of external relation extraction on NEREL

Model	F1-Score
SpanBERT, mBERT weights	76.1
SpanBERT, RuBERT weights	78
TRE	67.2
OpenNRE BERT: ent.marker-ent	81
OpenNRE BERT: ent.marker-cls	78
OpenNRE BERT: mask-ent	64
OpenNRE BERT: mask-cls	62

6. EXPERIMENTS

6.1. Separate Extraction of Internal and External Relations

To implement the proposed approach, various datasets were compiled for each of the subtasks

1. Separately allocated external and nested entities.
2. Within sentence (within the longest entity for the second subtask), all possible pairs of entities were taken, and the relation was indicated if it was present, or otherwise “no_relation” was noted: (Ti, Tj, no_relation).
3. Further, such an example was converted into the data format required by the model.

6.1.1. Extracting external relations. As part of the experiments, the SpanBERT [19] and TRE [10] models were further trained and tested, achieving results close to state-of-the-art on the TACRED [18] and RE-TACRED [23] English dataset. This models reflects the main approaches of transformer-type models to solving the problem of extracting relations.

SpanBERT—BERT adapted to work with text fragments; for uptraining, the pretrained weights of the multilingual BERT model trained on a set of documents in different languages, including Russian, as well as the weights of the RuBERT model [24] were taken. In this work, we used the implementation provided in the Facebook repository [19]. SpanBERT uses entity masking and cannot be directly used for extracting named entities.

TRE [10]—based on GPT model [25]. TRE pre-trained mainly in English, but the dictionary of this model used in the input text encoder module also contains Russian language tokens, which allows you to apply this model to texts in Russian.

OpenNRE BERT [14]—implementation of BERT for RE from the most popular frameworks for extracting relations, which implements various models. OpenNRE [14] is designed for various scenarios for RE, including sentence-level RE, bag-level RE, document-level RE, and few-shot RE.

Sentence-level Relation Extraction in OpenNRE presented by two different encoders: CNN [6] and BERT [20]. For CNN, framework follows the setting of Nguyen and Grishman [7], including using word and position embeddings. For BERT, OpenNRE follows the setting of Soares et al. [22].

Experiments were carried out for different sets of parameters:

- *ent.marker/mask* refers to using entity markers or masks in model input;
- *ent/cl*s refers to taking entity embedding (entity marker or entity mask) or [CLS] token as model output.

Table 4. Result of nested relation extraction on NEREL

Model	F1-Score
SVM + fastText	51.1
Dense + fastText	64.2
Dense + fastText + TypeBin	69
Dense + fastText + TypeEmb	72.5
Dense + fastText + TypeEmb + NestFlag	75.2

Table 5. Result of joint extraction of relations on NEREL

Model	F1-Score
OpenNRE BERT: ent.marker-ent	80.1
OpenNRE BERT: ent.marker-cls	75.9

Using entity markers for represent relation in model input and output achieves better f1-score than other approaches (Table 3).

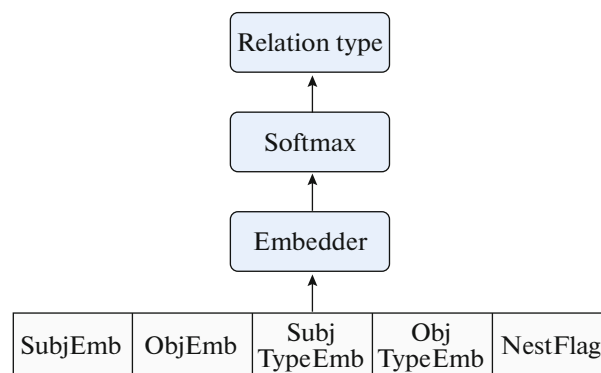
6.1.2. Extraction of internal relations. To extract internal relations without considering the context, a baseline model (Fig. 2) was developed based on a fully connected network with an architecture that accepts a vector representation of words without context as input. The approach of most models that solve the Relation extraction problem is taken as a basis—to obtain a vector representation of the entities under consideration and feed it to the input of the linear layer with softmax.

The following features were used:

- *Embedder*—Dense layer/SVM;
- *SubjEmb*, *ObjEmb*—FastText embeddings of relation subject and object;
- *SubjTypeEmb*, *ObjTypeEmb*—learned embeddings of relation entity types obtained from Embedding Layer;
- *NestFlag*—indicator of entities nesting.

Table 4 shows the contribution of each feature to the improvement of the model.

6.1.3. Aggregation of results. In the proposed approach of extraction relations from texts containing nested entities, different subtasks are solved separately. Therefore, in order to get the final

**Fig. 2.** Baseline model for nested relations.

result for detecting relations on the NEREL dataset, it is necessary to calculate a weighted metric according to how many relations (not including “no_relation”) were considered in each subtask on the dataset tests. In accordance with the proportions of types of relations, we obtain the metric for best models $F1 = 81\% \times 0.71 + 75.2\% \times 0.19 = 71.8\%$.

6.2. Joint Extraction of Relations

For this approach, BERT from OpenNRE was used with entity markers in input and the modification described in paragraph 5.2.

This approach showed a better final metric (Table 5) than the separate extraction approach. Also this method is more convenient and does not require additional steps.

7. CONCLUSIONS

In this paper we study the problem of applying state-of-the-art models to relation extraction for nested entities. Most datasets annotated with nested named entities do not have relation markup. Therefore some state-of-the-art relation extraction models cannot extract relations between nested or overlapping entities because of their architecture or input format.

As a solution, we consider the approach of reducing the problem to two subtasks and a joint extraction method. For the first subtask, state-of-the-art models are applied. For the second subtask, a neural network model is implemented. This task can be solved with quality not much worse than heavy architectures.

We compare various approaches to representation of relation entities. It is shown that using position features and entity markers allow solving the problem of relation extraction for intersecting entities.

We improved the OpenNRE model to solve the problem with one model for all relation types. It is presented that this approach allows achieving the best quality in this task.

FUNDING

The project is supported by the Russian Science Foundation, grant no. 20-11-20166.

REFERENCES

1. B. Plank, K. N. Jensen, and R. van der Goot, “Dan+: Danish nested named entities and lexical normalization,” in *Proceedings of the 28th International Conference on Computational Linguistics* (2020), pp. 6649–6662.
2. W. Buaphet et al., “Thai nested named entity recognition corpus,” in *Findings of the Association for Computational Linguistics: ACL 2022* (Assoc. Comput. Linguist., Dublin, Ireland, 2022), pp. 1473–1486.
3. N. Ringland et al., “NNE: A dataset for nested named entity recognition in english newswire,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019), pp. 5176–5181.
4. N. V. Loukachevitch et al., “NEREL: A Russian dataset with nested named entities and relations,” in *Proceedings of the International Conference on Recent Advances in Natural Language Processing RANLP-2021* (2021), pp. 876–885.
5. N. Bach and S. Badaskar, “A survey on relation extraction,” Presentation (Language Technol. Inst., Carnegie Mellon Univ., 2007).
6. D. Zeng et al., “Relation classification via convolutional deep neural network,” in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (2014), pp. 2335–2344.
7. T. H. Nguyen and R. Grishman, “Combining neural networks and log-linear models to improve relation extraction,” in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing* (2015), pp. 39–48.
8. S. Zhang et al., “Bidirectional long short-term memory networks for relation classification,” in *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation, Shanghai, China* (2015), pp. 73–78.
9. S. Wu and Y. He, “Enriching Pre-trained Language Model with Entity Information for Relation Classification,” in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019), pp. 2361–2364. <https://doi.org/10.48550/ARXIV.1905.08284>

10. A. Christoph and L. H. Marc Hubner, "Improving relation extraction by pre-trained language representations," arXiv: 1906.03088 (2019).
11. X. Wang and T. Gao, "Kepler: A unified model for knowledge embedding and pre-trained language representation," *Trans. Assoc. Comput. Linguist.* **9**, 176–194 (2021).
12. J. Feng et al., "Reinforcement learning for relation classification from noisy data," in *Proceedings of the AAAI Conference on Artificial Intelligence AAAI-2018* (2018), pp. 5779–5786.
13. A. D. Cohen, S. Rosenman, and Y. Goldberg, "Relation classification as two-way span-prediction," arXiv: 2010.04829 (2020).
14. X. Han, T. Gao, and Y. Yao, "Opennre: An open and extensible toolkit for neural relation extraction," 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* (2019), pp. 169–174.
15. D. Gordeev et al., "Relation extraction dataset for the russian language," in *Computational Linguistics and Intellectual Technologies: Proceedings of the Dialog International Conference* (2020).
16. A. Starostin et al., "Factrueval 2016: Evaluation of named entity recognition and fact extraction systems for russian," in *Computational Linguistics and Intellectual Technologies: Proceedings of the Dialog International Conference* (2016), pp. 702–720.
17. V. Ivanin et al., "Rurebus-2020 shared task: Russian relation extraction for business," in *Computational Linguistics and Intellectual Technologies: Proceedings of the Dialog International Conference* (2020), pp. 416–431.
18. Y. Zhang et al., "Position-aware attention and supervised data improve slot filling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017), pp. 35–45. <https://doi.org/10.18653/v1/D17-1004>
19. M. Joshi et al., "Spanbert: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguist.* **8**, 64–77 (2019).
20. J. Devlin et al., "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* (2019), Vol. 1, pp. 4171–4186.
21. H. Peng et al., "Learning from context or names? An empirical study on neural relation extraction," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing EMNLP-2020* (2020), pp. 3661–3672.
22. L. Baldini Soares et al., "Matching the blanks: Distributional similarity for relation learning," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics ACL-2019* (2019), pp. 2895–2905.
23. G. Stoica, E. A. Platanios, and B. Poczoz, "Re-tacred: Addressing shortcomings of the TACRED dataset," in *Proceedings of the AAAI Conference on Artificial Intelligence AAAI-2021* (2021), Vol. 35, pp. 13843–13850.
24. Y. Kuratov and M. Arkhipov, "Adaptation of deep bidirectional multilingual transformers for russian language," arxiv: 1905.07213.
25. A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training" (2018).