DISCIE–Discriminative Closed Information Extraction

Cedric Möller[1(✉)] and Ricardo Usbeck[2]

[1] Department of Informatics, Semantic Systems, Universität Hamburg, Hamburg, Germany
cedric.moeller@uni-hamburg.de
[2] Institute for Information Systems, Artificial Intelligence and Explainability, Leuphana Universität Lüneburg, Lüneburg, Germany
ricardo.usbeck@leuphana.de

Abstract. This paper introduces a novel method for closed information extraction. The method employs a discriminative approach that incorporates type and entity-specific information to improve relation extraction accuracy, particularly benefiting long-tail relations. Notably, this method demonstrates superior performance compared to state-of-the-art end-to-end generative models. This is especially evident for the problem of large-scale closed information extraction where we are confronted with millions of entities and hundreds of relations. Furthermore, we emphasize the efficiency aspect by leveraging smaller models. In particular, the integration of type-information proves instrumental in achieving performance levels on par with or surpassing those of a larger generative model. This advancement holds promise for more accurate and efficient information extraction techniques.

1 Introduction

Today, our ability to generate data far surpasses our ability to understand it, particularly when that data is in textual form. As a potential solution, knowledge graphs (KGs), structured representations of data as interconnected nodes and links, offer the promise of making complex information machine-readable and easily interpretable [11].

However, the process of automatically transforming unstructured text into a meaningful KG is a significant unsolved problem. It encompasses numerous complex subproblems such as entity recognition, relation extraction and semantic understanding, where each represents a substantial field of study.

In general, this means that a text is translated to a set of triples. Each triple consists of a subject, a predicate and an object. Each subject and object is an entity while the predicate corresponds to a relation between the two entities.

In this work, we focus on Closed Information Extraction (CIE) [12]. Here, triples are extracted which are grounded in an underlying KG. This means that each of the extracted entities and relations have unique identifiers assigned. An example[1] is:

[1] Using QIDs and PIDs from www.wikidata.org. QIDs are the identifiers of entities and PIDs are the identifiers of relations.

G. Demartini et al. (Eds.): ISWC 2024, LNCS 15232, pp. 23–40, 2025.
https://doi.org/10.1007/978-3-031-77850-6_2

"Barack Obama was born in Hawaii" → `[Q76, P19, Q782]`

Recent methods like the State-of-the-Art model GenIE interpreted the task as an end-to-end machine translation task where the input is the text and the output are the triples. Generative models are employed to translate text directly to triples. While generative models proved to be very powerful, it is harder to incorporate external information (e.g., the underlying KG) into the generation process [12]. Hence, the generative model is forced to learn the entire KG during training. As the size of such a KG can be huge, this can inhibit performance. Also, this means such generative methods are not able to use an evolving version of the KG. Furthermore, the sequential nature of the decoding process often to leads lower efficiency, which is critical given the large amount of textual data available today. Lastly, the authors reported a lower performance on long-tail relations.[2]

Instead of using generative models, this work employs discriminative models. This means, we first identify salient segments of the input text. Subsequently, external information is introduced to distinguish these segments within a predefined set of classes. The discriminative process encompasses tasks such as recognizing mentions, disambiguating entities, and extracting relations. Also, in many subtasks relevant to the task of end-to-end entity linking are non-generative models still state-of-the-art [18,30]. This allows us to tackle three shortcomings of the generative State-of-the-Art model: efficiency, inclusion of external information and performance on long-tail relations.

We employ lightweight models in each step of our method which gives us a large efficiency boost. While such methods often perform worse than their larger counterparts, we investigate whether the utilization of fine-grained entity type information as external information into relation extraction step can alleviate the performance gap. Lastly, we explore whether this information has a positive impact on the performance on long-tail relations as well. The primary emphasis of this paper is centered around enhancements made to the relation extraction component. In terms of mention recognition and entity linking, our approach trains and uses models that have demonstrated high performance.

The contributions of this paper are as follows:

- Show that the inclusion of coarse-grained type information is not sufficient;
- Show that the inclusion of fine-grained type information has a large impact on relation extraction and hence CIE in general (in particular long-tail relations);
- Show that efficient lightweight discriminative models can outperform large-scale generative models when using fine-grained type information.

In the following, we will develop such a discriminative method and especially focus on the incorporation of type information into the relation extraction step.[3]

[2] Relations rarely occurring.

[3] The code can be found in: https://github.com/semantic-systems/discie.

2 Method

2.1 Problem Definition–Closed Information Extraction

We define the problem as follows: Given are a text t and a KG $\mathcal{G} = (V, R, E)$ where V are all entities in the graph, R all relations in the graph and $E \subseteq (V \times R \times V)$ all edges each of which connects two entities via a relation. Each text t contains triples of form $< v, r, w >$ with $v, w \in V$ and $r \in R$. The goal is to extract these triples from text.

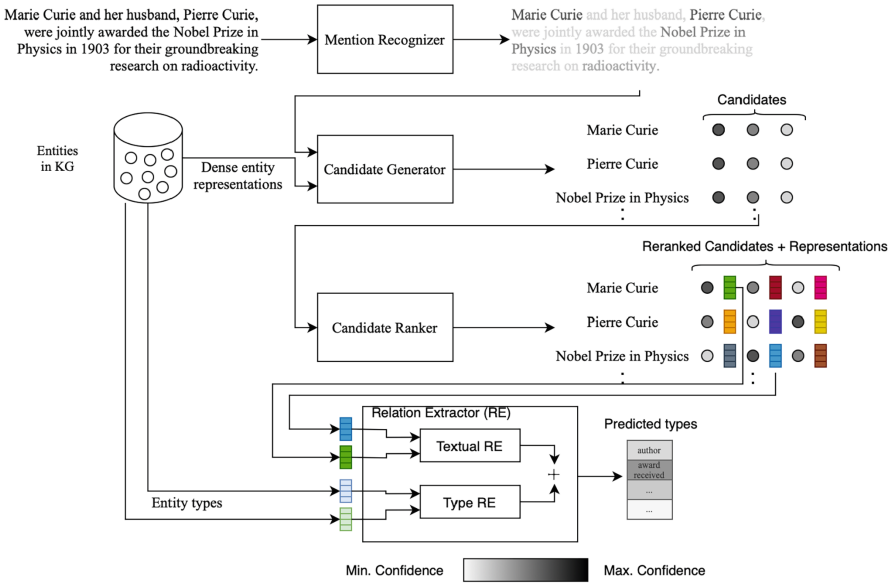

Fig. 1. DISCIE - Architecture. The intensity of the colors indicate the scores. Higher intensity resolves to a higher score. The likely outcome would be the triples: [(Q7186:Marie Curie, P166:award received, Q38104:Nobel Prize in Physics), (Q7186:Marie Curie, P26:spouse, Q37463:Pierre Curie)]

2.2 Model

Mention Recognizer. The mention recognizer is an encoder-only model that accepts the tokenized input text $t_1, \ldots, t_i, \ldots, t_n$. It encodes the whole sequence to get an embedded representation for each token $k_1, \ldots, k_i, \ldots, k_n$, where $k_i \in \mathbb{R}^d$. Then, each pair of subsequent tokens is combined by concatenation and fed into a linear layer $s_{i,j} = l(k_i \oplus k_j) \in \mathbb{R}$, classifying whether the pair denotes the first and last token of a mention or not. Overall it outputs $\frac{n(n+1)}{2}$ scores for a sequence of length n. All scores surpassing an initial threshold are taken as mention candidates and forwarded

to the entity candidate generation module.[4] The model is trained with the binary cross entropy loss function.

Entity Candidate Generator. The entity candidate generator is based on the bi-encoder architecture, more specifically a Siamese network [7]. The Siamese network is an encoder-only model. It encodes the textual mention representation and the textual entity representation into dense representations. The textual mention representation is of form

[CLS] `{mention}` `[CTX_L]` `{context_left}` `[CTX_R]` `{context_right}` [SEP]

where `context_left` and `context_right` is the text of a certain window size left and right of the identified mention. [CTX_L] and [CTX_R] are special tokens denoting the context. The textual entity representation is of form

[CLS] `{label}` `[DESC]` `{desc}` [SEP]

where `{label}` is the English label of the entity and `{desc}` is the short description text as available in Wikidata via the predicate `schema:description`.[5] Both, the mention and entity textual representations, are fed into the same encoder-only model and encoded to retrieve the final mention/entity representation, which is here taken as the embedded `[CLS]` token. We denote the embedded mention representations as b_m and the embedded entity representation as b_c. Finally, both representations, are compared via cosine similarity

$$\frac{< b_m, b_c >}{\|b_m\| \|b_c\|} \in \mathbb{R}$$

where $< \cdot, \cdot >$ denotes the dot product and $\|\cdot\|$ the euclidean norm.

The model is trained with the binary cross-entropy loss using in-batch negatives and mined hard-negatives [35]. When doing in-batch negative-based learning, all other entities in the current batch are interpreted as negatives. For mined hard-negatives, all entities are embedded after β epochs and for each training example, all γ nearest entities are found. All entities not being the ground-truth entity are now taken as negatives. The method returns for each mention m a set of candidates C_m. During training, $\beta = 1$ and $\gamma = 10$. After training, all entities are embedded and inserted into a vector index for fast retrieval.[6]

Entity Candidate Ranker. While the entity candidate generator alone could be used for entity disambiguation, it is usually less accurate. That is why in a subsequent step an entity candidate ranker is used that is less efficient but more accurate. It is applied

[4] Usually, mention recognition is solved by applying BIO sequence tagging. We trained and evaluated such a method but achieved a lower performance in comparison to the token-pair-based approach described above.

[5] This could be replaced with any other KG containing descriptions.

[6] https://faiss.ai.

to the subset of entities retrieved by the previous step. The candidate ranker re-ranks all the candidates C_m retrieved for a mention. It is based on the cross-encoder architecture [35]. It takes the concatenated textual representations of the mention and entity candidate and feeds it into an encoder-only model. The cross-encoder architecture allows cross-attention between the entity representation and the input text. This usually leads to higher performance than just comparing the bi-encoder representations directly [35].[7] Hence, the input is of form:

[CLS] `{label} [DESC] {desc} [SEP] {mention} [CTX_L] {context_left} [CTX_R] {context_right}` [SEP]

[DESC] is a special token denoting the entity description. The embedded `[CLS]` token is taken (denoted as $b_{m,c}$) and fed into a final linear layer to get a similarity score

$$s_{m,c} = h(b_{m,c}) \in \mathbb{R}.$$

During training, for each entity mention a set of hard negative entity candidates is sampled by using the entity candidate generator and the vector index. The model is trained via binary cross entropy loss including all the hard negatives and the positive entity candidate.

Relation Extractor

Textual Information. The relation extractor accepts a pair of entity mentions. Instead of only focusing on the input text, we incorporate candidate information as well. We take each mention m and its highest scoring candidate c, and combine both $f(m, c)$. Then, each $f(m, c)$ is compared to all other $f(m', c')$ where $m \neq m'$. As the combination of each mention and its candidate $f(m, c)$ we simply use the embedded `[CLS]` token output by the candidate ranker, so $f(m, c) = b_{m,c}$. The predicted relation is scored by first calculating whether a subject-object relationship holds between a pair

$$< l_s(b_{m,c}), l_o(b_{m',c'}) > \in \mathbb{R}$$

where l_s and l_o are learnable linear layers. Then, a score for each potential relation is calculated as

$$W_r \left[b_{m,c} + b_{m',c'}\right]$$

where $W_r \in \mathbb{R}^{|R| \times d}$ is a learnable matrix and d is the dimension of $b_{m,c}$ and $b_{m',c'}$.

Both scores are then combined to get the final relation score

$$g[b_{m,c}, b_{m',c'}] = W_r \left[b_{m,c} + b_{m',c'}\right] + < l_s(b_{m,c}), l_o(b_{m',c'}) > \mathbf{1}$$

where $\mathbf{1} \in \mathbb{R}^{|R|}$ is a vector of ones. It holds that $g[b_{m,c}, b_{m',c'}] \in \mathbb{R}^{|R|}$.

[7] This was also observable in our use case.

Type Information. Additionally, we also incorporate fine-grained type information into the relation extraction process. This is based on the intuition that certain relations are usually restricted to combinations of certain entity types. To learn these dependencies, we calculate relation classification logits separately from the textual representations just using the type information of each candidate. Each entity candidate c has a set of types $T_c \subseteq T$ where T is the set of types available in Wikidata. Now, we assign each type $t \in T$ a learnable vector $e_t \in \mathbb{R}^{d_T}$. As an entity might have multiple types, we create a condensed representation of the candidate as $t_c = \frac{1}{|T_c|} \sum_{t \in T_c} e_t$.

Then, we calculate the type-based relation logits by feeding the concatenation of t_c and another candidate $t_{c'}$ into a linear layer:

$$h(t_c \oplus t_{c'}) \in \mathbb{R}^{|R|}$$

Finally, we sum up the contextual logits and the type logits to get the final logits:

$$k(m, c, m', c') = h(t_c \oplus t_{c'}) + g[f(m, c) \oplus f(m', c')]$$

The relation extractor is trained via binary cross-entropy loss.

Inference. First, we retrieve a set of suitable mentions by applying the mention recognizer. After mapping its output to $(0, 1)$ by applying the sigmoid function, we retrieve a score $s^m_{i,j}$ for each possible span. Now, all spans surpassing a threshold ϵ_m are taken as mention candidates. For each such mention, the entity candidate generator is applied to retrieve a set of candidates. Each candidate is reranked by applying the entity candidate reranker. Its scores $s^c_{m,c}$ are again mapped to $(0, 1)$. The final candidate score is then the average $s = \frac{s^m_{i,j} + s^c_{m,c}}{2}$. If the maximum score of all candidates surpasses a threshold ϵ_c, the candidate and its mention are accepted. Finally, the relations of each pair of mention-candidate combinations are calculated by using the relation extractor to get the relation scores s_r. Each relation score surpassing the final relation threshold ϵ_r is accepted.

Table 1. Results on REBEL and FewRel (Micro)

	REBEL				FewRel
Model	P	R	$F1$	$F2$	R
SOTA-Pipeline	43.30±0.15	41.73±0.13	42.50±0.13	–	17.89±0.24
GenIE	68.02±0.15	69.87±0.14	68.93±0.12	–	30.77±0.27
GenIE - PLM	59.32±0.13	77.78±0.12	67.31±0.10	–	46.95±0.27
DISCIE (F2 calibrated)	62.13±0.10	**81.93±0.07**	70.67±0.08	**77.02±0.06**	**47.10±0.28**
DISCIE (F1 calibrated)	**77.41±0.11**	72.68±0.08	**74.97±0.08**	73.58±0.07	34.39±0.29

Table 2. Results on GeoNRE and WikipediaNRE (Micro)

	GeoNRE			WikipediaNRE		
Model	P	R	$F1$	P	R	$F1$
SOTA-Pipeline	$66.65_{\pm 1.47}$	$66.22_{\pm 1.46}$	$66.43_{\pm 1.45}$	$65.17_{\pm 0.27}$	$54.40_{\pm 0.20}$	$59.30_{\pm 0.21}$
SetGenNet	$86.89_{\pm 0.51}$	$85.31_{\pm 0.47}$	$86.10_{\pm 0.34}$	$82.75_{\pm 0.11}$	$77.55_{\pm 0.27}$	$80.07_{\pm 0.27}$
GenIE	$91.77_{\pm 0.98}$	$\mathbf{93.20_{\pm 0.83}}$	$\mathbf{92.48_{\pm 0.88}}$	$91.39_{\pm 0.15}$	$\mathbf{91.58_{\pm 0.15}}$	$91.48_{\pm 0.12}$
DISCIE	$\mathbf{92.4_{\pm 0.90}}$	$87.2_{\pm 1.02}$	$89.71_{\pm 0.86}$	$\mathbf{91.57_{\pm 0.16}}$	$91.53_{\pm 0.13}$	$\mathbf{91.55_{\pm 0.12}}$

3 Evaluation

For evaluation, we used four different datasets: REBEL, WikipediaNRE, GeoNRE and FewRel. Here, REBEL [3] is a large-scale dataset while WikipediaNRE, GeoNRE [34] and FewRel [10] are of smaller size. In regard to relations, REBEl contains 857 different relations while the other three datasets all contain fewer than 157 relations. During the evaluation, FewRel is used as a recall-only benchmark dataset as it is not exhaustively annotated [12]. See Table 3 for information on the datasets. For the candidate representations, we use the concatenation of the Wikipedia title and the Wikidata description of the entity. The used Wikdata dump is from 2022. As for type information, we use the fine-grained types as given by the `P31` relation in Wikidata. Also, we extract for each type all supertypes and consider them as valid types of an entity. Finally, we restrict the set of types to the set as defined by Ayoola et al. [2] due to them showing great performance utilising them in the task of entity linking.[8]

Table 3. Statistics of the datasets

Dataset	Examples			Triples			# entities	# relations
	Train	Dev	Test	Train	Dev	Test		
Rebel	1,899,331	104,960	105,516	5,147,836	284,268	284,936	1,498,143	857
WikipediaNRE	223,536	980	29,619	298,489	1,317	39,678	278,204	157
GeoNRE	–	–	1,000	–	–	1,000	124	11
FewRel	–	–	27,650	–	–	27,650	64,762	80

Similar to Josifoski et al. [12] we follow two training regimes: For REBEL and FewRel, we train on the training dataset of REBEL and evaluate on the REBEL test and FewRel test set. For WikipediaNRE and GeoNRE, we finetune the already REBEL-trained model on the training dataset of WikipediaNRE and then evaluate on the test sets of WikipediaNRE and GeoNRE.[9]

[8] 930 types are used in total. They were filtered by exploring how useful they are for disambiguating between different entities.

[9] When evaluating on GeoNRE or WikipediaNRE, we limited the set of available predictable relations and entities to the same set as used in the work by Josifoski et al. [12]. Therefore, we set prediction scores for out-of-scope relations to 0.0.

The thresholds ϵ_m, ϵ_c and ϵ_r necessary for inference are tuned on the validation sets of REBEL, respectively WikipediaNRE.

We use `distilbert-base-cased` for the mention recognizer, and `all-MiniLM-L12-v2` for the bi-encoder, cross-encoder and relation extractor. While larger models potentially perform better, due the efficiency objective and the fact that we are a small university lab, we rely on such lightweight models. We train each model for 10 epochs on two NVIDIA A6000s and select the best-performing model by evaluating on the validation datasets. We use a learning rate of $2 \cdot 10^{-5}$ for all models.

3.1 CIE Evaluation

For the closed information extraction task, we compare our trained model, denoted as DISCIE, to the same models as used in the works by Josifoski et al. [12]. GenIE is the SOTA model by Josifoski et al. utilizing a generative model trained from scratch. GenIE-PLM is the same model but initialized from pre-trained BART [14]. SetGenNet [32] is a encoder-decoder-based model utilising bi-partite matching for extracting triples. Finally, SOTA-Pipeline is a pipeline-based model by Josifoski et al. relying on a sequence of SOTA models for the tasks of mention recognition, entity linking and relation extraction. For more information on this pipeline, please refer to their paper.[10]

We report micro/macro precision, recall and F1 for all models as well as F2 for our model. Micro refers here to calculating the metric over all examples while macro calculates the metrics first for each relation separately and then averages them.

Similar to Josifoski et al., we report the metrics with a 1-standard-deviation confidence interval constructed from 50 bootstrap samples of the data for all results.

In Table 1, we show the results on the REBEL and the FewRel datasets. Our method outperforms the best-performing method GenIE by more than 5 F1-measure points.

It can be seen that DISCIE has much higher precision while lacking recall in comparison to GenIE. When tuning the thresholds for F2 instead F1 on the validation dataset[11], we see that the recall on the subset of data surpasses GenIE while also surpassing it in overall F1. On the recall-only benchmark FewREL, the F2-calibrated DISCIE performs slightly better than GenIE-PLM while the F1-calibrated DISCIE performs much worse. This is the case as the F1-calibrated DISCIE puts more emphasis on precision which leads to a reduced recall.

Table 2 presents the results for GeoNRE and WikipediaNRE. On GeoNRE, DISCIE performs nearly 3 F1 points worse while on WikipediaNRE it is only slightly better.

On REBEL, macro F1 of DISCIE surpasses the second-best method GenIE by nearly 7 points (see Table 4) with the F1 calibrated method and by more than 9 points with the F2 calibrated one. This means DISCIE performs more uniformly than GenIE across all relation types. This is especially important due to the large number of relations occurring in the dataset where many are long-tail relations.[12]

[10] We did not compare to SCICERO [8] as we were not able to adapt their code to our datasets.

[11] Hence putting more emphasis on recall.

[12] They occur only rarely in the training data.

Table 4. Results on REBEL (Macro)

Model	P_{Macro}	R_{Macro}	$F1_{\text{Macro}}$	$F2_{\text{Macro}}$
SOTA-Pipeline	$12.20_{\pm 0.35}$	$10.44_{\pm 0.22}$	$9.48_{\pm 0.21}$	–
GenIE	$33.90_{\pm 0.73}$	$30.48_{\pm 0.65}$	$30.46_{\pm 0.62}$	–
DISCIE (F2 calibrated)	$35.84_{\pm 0.59}$	$\mathbf{43.99_{\pm 0.61}}$	$\mathbf{39.50_{\pm 0.56}}$	$\mathbf{42.08_{\pm 0.57}}$
DISCIE (F1 calibrated)	$\mathbf{44.05_{\pm 0.84}}$	$42.29_{\pm 0.62}$	$37.27_{\pm 0.67}$	$34.11_{\pm 0.63}$

On WikipediaNRE and GeoNRE, while DISCIE sometimes underperformed or only matched the performance of GenIE on micro metrics, we see that in regard to macro metrics, it outperforms GenIE (see Table 5). On WikipediaNRE it outperforms GenIE by 8 points on F1. On GeoNRE, DISCIE surpasses GenIE by more than 3 F1 points. DISCIE is therefore also performing more uniformly on those datasets.

Table 5. Results on GeoNRE and WikipediaNRE (Macro)

	GeoNRE			WikipediaNRE		
Model	P_{Macro}	R_{Macro}	$F1_{\text{Macro}}$	P_{Macro}	R_{Macro}	$F1_{\text{Macro}}$
SOTA-Pipeline	$38.67_{\pm 5.72}$	$34.49_{\pm 5.99}$	$35.14_{\pm 5.09}$	$24.12_{\pm 1.46}$	$16.55_{\pm 1.00}$	$17.76_{\pm 1.01}$
GenIE	$\mathbf{75.77_{\pm 7.80}}$	$71.60_{\pm 7.95}$	$72.59_{\pm 7.32}$	$52.55_{\pm 2.12}$	$45.95_{\pm 1.67}$	$47.08_{\pm 1.68}$
DISCIE	$73.65_{\pm 6.61}$	$\mathbf{76.72_{\pm 6.54}}$	$\mathbf{75.05_{\pm 6.01}}$	$\mathbf{53.76_{\pm 2.14}}$	$\mathbf{51.80_{\pm 2.05}}$	$\mathbf{52.75_{\pm 1.90}}$

Figure 2 compares the F1 for all relations separated by their number of occurrences in the training data on REBEL. As can be seen, the performance of DISCIE is surpassing the performance of GenIE consistently. For long-tail entities with an occurrence count between 16 and 64 (2^4 and 2^6), the performance sometimes nearly doubles.

3.2 Ablation

To identify what aspects of the relation extractor contributed the most to the performance, we conducted an ablation study on REBEL. Table 6 compares regular DISCIE to DISCIE without any type information (w/o types), DISCIE without candidate descriptions (w/o desc.) and DISCIE with coarse-grained types (w/ coarse).

Excluding type information (w/o types) leads to a large decrease in performance. Especially the precision decreases by many points. Therefore, implicit KG information given by type information helps the model to more precisely decide on the right relation while filtering out relations not compatible with the types provided by the entities.

Not using candidate information (w/o candidate description) and only relying on the textual information at hand leads to a slight decrease in performance by around 0.5 F1 points. Hence, the information contained in the description is not fully replaced by the available type information.

Table 6. Ablation study of the relation extractor evaluated over REBEL dataset (w/o types: relation extractor does not use type information, w/o desc.: relation extractor does not use candidate descriptions, w/o text: relation extractor does neither use candidate descriptions nor the input text, w/ coarse: regular relation extractor but coarse-grained types are used)

Model	P	R	F1
DISCIE w/o types	62.41±0.07	69.08±0.08	65.58±0.06
DISCIE w/o desc.	76.82±0.11	72.14±0.07	74.41±0.07
DISCIE w/o text	59.75±0.24	35.87±0.09	44.83±0.10
DISCIE w/ coarse	68.32±0.08	68.31±0.08	68.32±0.06
DISCIE	**77.41**±0.11	**72.68**±0.08	**74.97**±0.08

Only using type information (w/o text) and not relying on any textual or candidate description information leads to the largest decrease in performance. The model is still able to often predict the correct relation by just using the available type information. Some combination of types therefore strongly predict the occurrence of certain relations in the text. However, the task is not trivial and therefore textual information at hand is still a necessity.

Lastly, replacing the fine-grained types with coarse-grained types of form `PER`, `ORG`, `LOC`, `MISC` (w/ coarse) leads to an increase in performance in comparison to not using type information at all. Nevertheless, using fine-grained types increases the performance much more.

3.3 Efficiency

We evaluate the efficiency via the GeoNRE dataset by running GenIE and DISCIE three times on its evaluation dataset.[13] Due to the length differences of the examples, the average number of seconds per 1000 examples can vary between datasets. In the GeoNRE dataset, DISCIE is approximately 27 times as fast as GenIE while outperforming it or matching it on several benchmarks (see Table 7).

3.4 Error Analysis

Figure 3 shows what components amount to what percentage of error on REBEL. As can be seen, is the candidate generation the least prone to errors. Usually, the correct candidate is in the generated candidate set. Candidate ranking is more prone to errors than the candidate generation. Sometimes, the wrong candidate is ranked the highest. The components that contribute the most to the errors are the relation extraction and mention recognition.

Additionally, we compare the results for three different examples for both GenIE and DISCIE in Table 8. Example 1 shows that DISCIE often performs better when

[13] GenIE takes a long time to evaluate on the other datasets on a single GPU. Therefore we opted for only running the efficiency tests on the smallest dataset. While the average speed differs between the datasets, DISCIE was considerably faster for all of them.

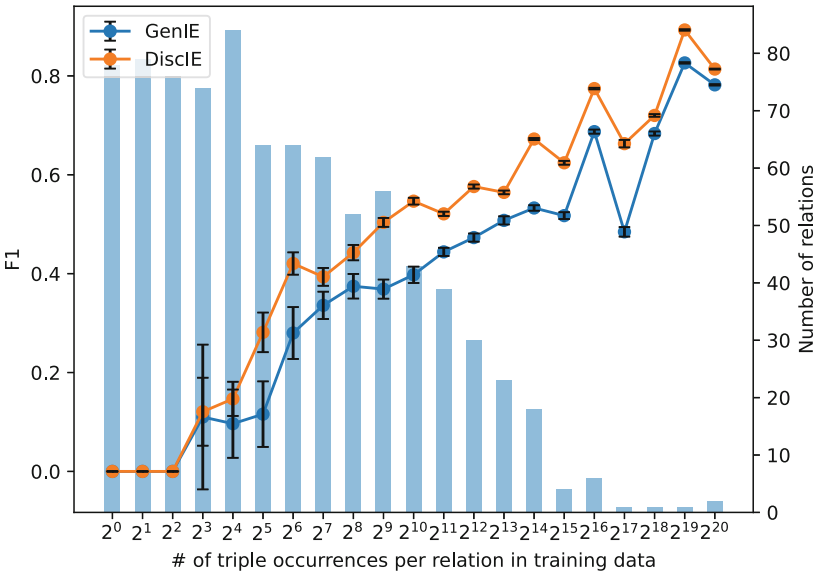

Fig. 2. F1 for GenIE and DISCIE over REBEL plotted for buckets of relations; each bucket contains all relations occurring a specific number of times in the training data. Each blue bar shows the number of relations occurring in the # of triples as given by the x-axis (see right vertical axis). (Color figure online)

focusing on long-tail relations. Here, DISCIE predicts both the relations `employer` and `musical conductor` while GenIE only predicts `member of`. While `member of` is close to `employer`, `employer` is more specific. On the other hand, `musical conductor`, a long-tail relation, is not predicted by GenIE while DISCIE can predict these. In Example 2 GenIE predicted the correct relation `employer` while DISCIE mistakenly predicted `educated at`. Here, `educated at` can also be seen as a fitting relation but it was just not labeled. Note, that this a common occurrence with GenIE, namely triples are predicted that are reasonable but not labeled. Lastly, Example 3 shows a case where both GenIE and DISCIE fail. Here, both methods generate more triples than necessary. Most of them describe implicit relations. A notable exception is the triple `(Spain, capital, Madrid)` that is predicted by both models. This relation is not stated in the input text but both models likely just predict it due to it often being seen during training.

4 Related Work

Entity Linking has a long history of research [20]. Recent methods can be categorized into two types. First, discriminative methods that are based on the bi-encoder/cross-encoder pairing [2,17,35]. Both encoders are commonly BERT-like models. The bi-encoder encodes the description of each entity and matches it to the text by using

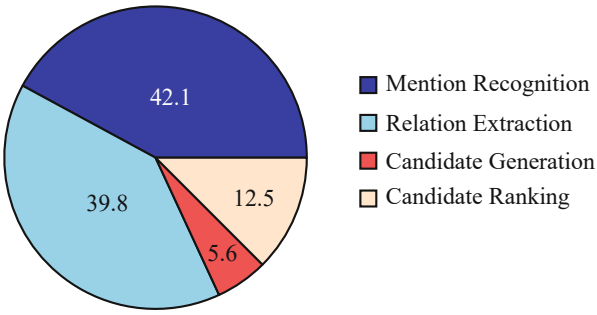

Fig. 3. Error distribution over all components on REBEL

Table 7. Efficiency on GeoNRE dataset run on a single NVIDIA A6000 GPU

Model	Seconds/1000 Examples
DISCIE	$21.17_{\pm 0.62}$
GenIE	$571.95_{\pm 7.08}$

an approximate nearest neighbor search. This is important as the next step, the cross-encoding, is expensive. Here, those neighbors are reranked by applying a cross-encoder to the concatenation of both, the input text and the entity description. The highest-ranked entity is then the final linked one. In the past, type information was used in several works in the entity linking domain. Incorporating it lead to a large increase in performance [2,27,28]. In contrast to that, we do not employ type information during entity linking but during relation extraction. Another type of entity linker is based on generative models [4,5]. Here, instead of using some external description of an entity, the whole model memorizes the knowledge graph (KG) during training. The linked entity is then directly generated by the model. Such methods skip the problem of mining negatives which are crucial for a good performance of bi-encoder-based methods [4].

Relation extraction methods usually assume that the entities in the input text are already identified. The task is then to classify whether a relation between two entities is expressed in the text and if it is, what relation holds. Recent methods rely either on CNN [21,29,36], RNN [19,22] or transformer networks [31,39]. Also, generative models are applied, usually by extracting entities and relations jointly [25,37] but also methods solely focusing on relation extraction (RE) exist [3,23]. In contrast to DISCIE, these methods generally focus on a small number of relations and do not consider the entity linking task. Zhang et al. [38] include fine-grained information into a generative joint entity and relation extraction method. But in contrast to us they only focus on entity extraction and not entity linking. Furthermore, they only incorporate a single type per entity.

There exist two directions of research related to closed information extraction. First, pipeline-based approaches. For that, initially, the entities in the text were recognized, then the relations between the entities were identified and finally, relations and entities are linked to the KG [1,6,9]. While the modularity of pipeline-based approaches makes

Table 8. Comparison of the performance of DISCIE and GenIE on three different examples. Ground-truth triples are shown in bold

Method	**DISCIE**	**GenIE**
Ex. 1	In 2009, VÃąsquez was named a Gustavo Dudamel conducting fellow with the Los Angeles Philharmonic.	
Result 1	**(Gustavo Dudamel, employer, Los Angeles Philharmonic), (Los Angeles Philharmonic, musical conductor, Gustavo Dudamel)**	(Gustavo Dudamel, member of, Los Angeles Philharmonic)
Ex. 2	She earned her Ph.D in mathematics from the University of Illinois at Urbana-Champaign in 1919 under the supervision of Arthur Byron Coble.	
Result 2	(Arthur Byron Coble, educated at, University of Illinois at Urbana-Champaign)	**(Arthur Byron Coble, employer, University of Illinois at Urbana-Champaign)**
Ex. 3	The Santiago BernabÃl'u Stadium (,) is a football stadium in Madrid, Spain.	
Result 3	**(Santiago BernabÃl'u Stadium, sport, Association football), (Santiago BernabÃl'u Stadium, located in the administrative territorial entity, Madrid)**, (Santiago BernabÃl'u Stadium, country, Spain), (Madrid, country, Spain),(Spain, capital, Madrid), (Santiago BernabÃl'u Stadium, instance of, stadium)	**(Santiago BernabÃl'u Stadium, sport, Association football), (Santiago BernabÃl'u Stadium, located in the administrative territorial entity, Madrid)**, (Santiago BernabÃl'u Stadium, country, Spain), (Madrid, country, Spain),(Spain, capital, Madrid)

it possible to simply exchange some modules with a better one, they suffer from error propagation. To combat that, recent methods focus on tackling the problem end-to-end [16,32,34]. Here, each step of the pipeline is jointly executed at once. This enables the models to have interaction between the entity recognition, relation extraction and entity linking process. Lately, generative models like BART, T5, GPT-4 became more popular [15,24,26]. Usually, the tasks are here simply modeled as the translation of text to text. In 2022, Josifoski et al. [12,13] applied such a generative model to the CIE task reaching SOTA. Furthermore, they are the first two evaluate the CIE task on a large dataset with hundreds of relations and millions of entities. Our method is the first discriminative approach focusing on the large-scale closed-information extraction task. In contrast to GenIE by Josifoski et al., we do not rely on a generative model, but a discriminative one. Furthermore, instead of performing relation extraction solely on the textual data, we incorporate the entity candidate information in form of their descriptions and types. Both features prove to be especially valuable when doing the relation extraction task on datasets with a large number of relations.

5 Conclusion and Future Work

In this work, we showed that including fine-grained type information into a discriminative closed information extraction method leads to a large improvement. By using the type information, the model can learn the implicit ontological information contained in the underlying KG. It especially leads to an **increased performance on long-tail relations**. Furthermore, due to the reliance of DISCIE on only smaller language models, it can deliver great performance while being much **more efficient**. This allows our model to match or even surpass the performance of larger end-to-end CIE information models while being much faster.

A generative model such as GenIE can be trained on the closed information extraction task without having access to the entity mention positions. In contrast to that, our training setup relies on them. In future work, we want to investigate whether the model can be modified to skip the mention recognition. Furthermore, the inference procedure is currently performed in a greedy way. We suspect that globally optimizing the disambiguation graph can lead to an increase in performance, which we also want to pursue further in the future. Also, incorporating the type information into the entity linking module might lead to improvement. Finally, analysing which types have a bigger impact on performance is worth exploring.

Limitations. SynthIE [13] showed that the REBEL dataset suffers from some qualitative problems such as false negatives. We did not compare our method against a larger generative model, such as LLama [33]. Although an adapter-fine-tuned variant of such a large language model might potentially outperform our method, it would require a significantly larger parameter count and be less efficient. Our objective was to demonstrate that substantial performance improvements can be achieved even with a smaller parameter count and some external data.

Supplemental Material Statement: Source code for our System is available from: https://github.com/semantic-systems/discie

Acknowledgments. This project was supported by the Hub of Computing and Data Science (HCDS) of Hamburg University within the Cross-Disciplinary Lab program. Additionally, support was provided by the Ministry of Research and Education within the SifoLIFE project “RESCUE-MATE: Dynamische Lageerstellung und Unterstützung für Rettungskräfte in komplexen Krisensituationen mittels Datenfusion und intelligenten Drohnenschwärmen” (FKZ 13N16836).

References

1. Angeli, G., et al.: Bootstrapped self training for knowledge base population. In: Proceedings of the 2015 Text Analysis Conference, TAC 2015, Gaithersburg, Maryland, USA, November 16-17, 2015, 2015. NIST (2015). https://tac.nist.gov/publications/2015/participant.papers/TAC2015.Stanford.proceedings.pdf
2. Ayoola, T., Tyagi, S., Fisher, J., Christodoulopoulos, C., Pierleoni, A.: Refined: An efficient zero-shot-capable approach to end-to-end entity linking. In: Loukina, A., Gangadharaiah, R., Min, B. (eds.) Proceedings of the 2022 Conference of the North American Chapter of the

Association for Computational Linguistics: Human Language Technologies: Industry Track, NAACL 2022, Hybrid: Seattle, Washington, USA + Online, C41555 July 2022, pp. 209–220. Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.naacl-industry.24

3. Cabot, P.H., Navigli, R.: REBEL: relation extraction by end-to-end language generation. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November 2021, pp. 2370–2381. Association for Computational Linguistics (2021). https://doi.org/10.18653/V1/2021.FINDINGS-EMNLP.204
4. Cao, N.D., Izacard, G., Riedel, S., Petroni, F.: Autoregressive entity retrieval. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3-7 May 2021. OpenReview.net (2021). https://openreview.net/forum?id=5k8F6UU39V
5. Cao, N.D., et al.: Multilingual autoregressive entity linking. Trans. Assoc. Comput. Linguistics **10**, 274–290 (2022). https://doi.org/10.1162/tacl_a_00460
6. Chaganty, A.T., Paranjape, A., Liang, P., Manning, C.D.: Importance sampling for unbiased on-demand evaluation of knowledge base population. In: Palmer, M., Hwa, R., Riedel, S. (eds.) Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9-11 September 2017. pp. 1038–1048. Association for Computational Linguistics (2017). https://doi.org/10.18653/v1/d17-1109
7. Chicco, D.: Siamese neural networks: an overview. Artifi. Neural Netw., 73–94 (2021)
8. Dessì, D., Osborne, F., Recupero, D.R., Buscaldi, D., Motta, E.: SCICERO: a deep learning and NLP approach for generating scientific knowledge graphs in the computer science domain. Knowl. Based Syst. **258**, 109945 (2022). https://doi.org/10.1016/J.KNOSYS.2022.109945
9. Galárraga, L., Heitz, G., Murphy, K., Suchanek, F.M.: Canonicalizing open knowledge bases. In: Li, J., Wang, X.S., Garofalakis, M.N., Soboroff, I., Suel, T., Wang, M. (eds.) Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, 3-7 November 2014, pp. 1679–1688. ACM (2014). https://doi.org/10.1145/2661829.2662073
10. Han, X., et al.: Fewrel: a large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October - 4 November 2018, pp. 4803–4809. Association for Computational Linguistics (2018). https://doi.org/10.18653/V1/D18-1514, https://doi.org/10.18653/v1/d18-1514
11. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A survey on knowledge graphs: representation, acquisition, and applications. IEEE Trans. Neural Networks Learn. Syst. **33**(2), 494–514 (2022). https://doi.org/10.1109/TNNLS.2021.3070843
12. Josifoski, M., Cao, N.D., Peyrard, M., Petroni, F., West, R.: Genie: Generative information extraction. In: Carpuat, M., de Marneffe, M., Ruíz, I.V.M. (eds.) Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, 10-15 July 2022, pp. 4626–4643. Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.naacl-main.342
13. Josifoski, M., Sakota, M., Peyrard, M., West, R.: Exploiting asymmetry for synthetic training data generation: Synthie and the case of information extraction. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, 6-10 December 2023. pp. 1555–1574. Association for Computational Linguistics (2023). https://doi.org/10.18653/V1/2023.EMNLP-MAIN.96

14. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, J5-10 July 2020, pp. 7871–7880. Association for Computational Linguistics (2020). https://doi.org/10.18653/V1/2020.ACL-MAIN.703
15. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5-10 July 2020, pp. 7871–7880. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.703
16. Liu, Y., Zhang, T., Liang, Z., Ji, H., McGuinness, D.L.: Seq2rdf: an end-to-end application for deriving triples from natural language text. In: van Erp, M., Atre, M., López, V., Srinivas, K., Fortuna, C. (eds.) Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, USA, 8th - to -12th October 2018. CEUR Workshop Proceedings, vol. 2180. CEUR-WS.org (2018). https://ceur-ws.org/Vol-2180/paper-37.pdf
17. Logeswaran, L., Chang, M., Lee, K., Toutanova, K., Devlin, J., Lee, H.: Zero-shot entity linking by reading entity descriptions. In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July - 2 August 2019, Volume 1: Long Papers, pp. 3449–3460. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/p19-1335
18. Ma, Y., Wang, A., Okazaki, N.: DREEAM: guiding attention with evidence for improving document-level relation extraction. In: Vlachos, A., Augenstein, I. (eds.) Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, 2-6 May 2023, pp. 1963–1975. Association for Computational Linguistics (2023). https://doi.org/10.18653/V1/2023.EACL-MAIN.145
19. Miwa, M., Bansal, M.: 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, ACL 2016, 7-12 August 2016, Berlin, Germany, Volume 1: Long Papers. The Association for Computer Linguistics (2016). https://doi.org/10.18653/v1/p16-1105
20. Möller, C., Lehmann, J., Usbeck, R.: Survey on english entity linking on wikidata: datasets and approaches. Semantic Web **13**(6), 925–966 (2022). https://doi.org/10.3233/SW-212865
21. Nguyen, T.H., Grishman, R.: Relation extraction: perspective from convolutional neural networks. In: Blunsom, P., Cohen, S.B., Dhillon, P.S., Liang, P. (eds.) Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, VS@NAACL-HLT 2015, June 5, 2015, Denver, Colorado, USA. pp. 39–48. The Association for Computational Linguistics (2015). https://doi.org/10.3115/v1/w15-1506
22. Ni, J., Florian, R.: Neural cross-lingual relation extraction based on bilingual word embedding mapping. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3-7 November 2019, pp. 399–409. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1038
23. Ni, J., Rossiello, G., Gliozzo, A., Florian, R.: A generative model for relation extraction and classification. CoRR abs/ arXiv: 2202.13229 (2022)
24. OpenAI: GPT-4 technical report. CoRR abs/ arXiv: 2303.08774 (2023)
25. Paolini, G., et al.: Structured prediction as translation between augmented natural languages. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event,

Austria, 3-7 May 2021. OpenReview.net (2021). https://openreview.net/forum?id=US-TP-xnXI
26. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020). http://jmlr.org/papers/v21/20-074.html
27. Raiman, J.: Deeptype 2: Superhuman entity linking, all you need is type interactions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 8028–8035 (2022)
28. Raiman, J., Raiman, O.: Deeptype: multilingual entity linking by neural type system evolution. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
29. dos Santos, C.N., Xiang, B., Zhou, B.: Classifying relations by ranking with convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, 26-31 July 2015, Beijing, China, Volume 1: Long Papers, pp. 626–634. The Association for Computer Linguistics (2015). https://doi.org/10.3115/v1/p15-1061
30. Shavarani, H., Sarkar, A.: Spel: Structured prediction for entity linking. In: Bouamor, H., Pino, J., Bali, K. (eds.) Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, 6-10 December 2023, pp. 11123–11137. Association for Computational Linguistics (2023). https://doi.org/10.18653/V1/2023.EMNLP-MAIN.686
31. Soares, L.B., FitzGerald, N., Ling, J., Kwiatkowski, T.: Matching the blanks: Distributional similarity for relation learning. In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July - 2 August 2019, Volume 1: Long Papers, pp. 2895–2905. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/p19-1279
32. Sui, D., Wang, C., Chen, Y., Liu, K., Zhao, J., Bi, W.: Set generation networks for end-to-end knowledge base population. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pp. 9650–9660. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.emnlp-main.760
33. Touvron, H., et al.: Llama: Open and efficient foundation language models. CoRR abs/arXiv: 2302.13971 (2023)
34. Trisedya, B.D., Weikum, G., Qi, J., Zhang, R.: Neural relation extraction for knowledge base enrichment. In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July - 2 August 2019, Volume 1: Long Papers, pp. 229–240. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/p19-1023
35. Wu, L., Petroni, F., Josifoski, M., Riedel, S., Zettlemoyer, L.: Scalable zero-shot entity linking with dense entity retrieval. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, 16-20 November 2020. pp. 6397–6407. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-main.519
36. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: Hajic, J., Tsujii, J. (eds.) COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 23-29 August 2014, Dublin, Ireland, pp. 2335–2344. ACL (2014), https://aclanthology.org/C14-1220/
37. Zhang, R.H., Liu, Q., Fan, A.X., Ji, H., Zeng, D., Cheng, F., Kawahara, D., Kurohashi, S.: Minimize exposure bias of seq2seq models in joint entity and relation extraction. In: Cohn,

T., He, Y., Liu, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020. Findings of ACL, vol. EMNLP 2020, pp. 236–246. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.23

38. Zhang, S., Ng, P., Wang, Z., Xiang, B.: Reknow: Enhanced knowledge for joint entity and relation extraction. CoRR abs/ arXiv: 2206.05123 (2022)
39. Zhong, Z., Chen, D.: A frustratingly easy approach for entity and relation extraction. In: Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y. (eds.) Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, 6-11 June 2021, pp. 50–61. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.naacl-main.5