GenIE: Generative Information Extraction

Martin Josifoski,[1] Nicola De Cao,[2,3] Maxime Peyrard,[1] Fabio Petroni,[4] Robert West[1]
[1]Ecole Polytechnique Fédérale de Lausanne
[2]University of Amsterdam, [3]University of Edinburgh, [4]Meta AI
martin.josifoski@epfl.ch, nicola.decao@gmail.com
maxime.peyrard@epfl.ch, robert.west@epfl.ch, fabiopetroni@fb.com

Abstract

Structured and grounded representation of text is typically formalized by *closed information extraction*, the problem of extracting an exhaustive set of *(subject, relation, object)* triplets that are consistent with a predefined set of entities and relations from a knowledge base schema. Most existing works are pipelines prone to error accumulation, and all approaches are only applicable to unrealistically small numbers of entities and relations. We introduce GenIE (generative information extraction), the first end-to-end autoregressive formulation of closed information extraction. GenIE naturally exploits the language knowledge from the pre-trained transformer by autoregressively generating relations and entities in textual form. Thanks to a new bi-level constrained generation strategy, only triplets consistent with the predefined knowledge base schema are produced. Our experiments show that GenIE is state-of-the-art on closed information extraction, generalizes from fewer training data points than baselines, and scales to a previously unmanageable number of entities and relations. With this work, closed information extraction becomes practical in realistic scenarios, providing new opportunities for downstream tasks. Finally, this work paves the way towards a unified end-to-end approach to the core tasks of information extraction.

1 Introduction

The ability to extract structured semantic information from unstructured texts is crucial for many AI tasks such as knowledge discovery (Ji and Grishman, 2011; Trisedya et al., 2019), knowledge maintenance (Tang et al., 2019), symbolic representation, and reasoning (Ji et al., 2021). The interface between free text and structured knowledge is formalized by *knowledge base population* (KBP; Ji and Grishman, 2011), which proposes to represent the information contained in text using *(subject, relation, object)* fact triplets. In this work, we focus on closed information extraction (cIE), the problem of extracting exhaustive sets of fact triplets expressible under the relation and entity constraints defined by a Knowledge Base (KB) schema.

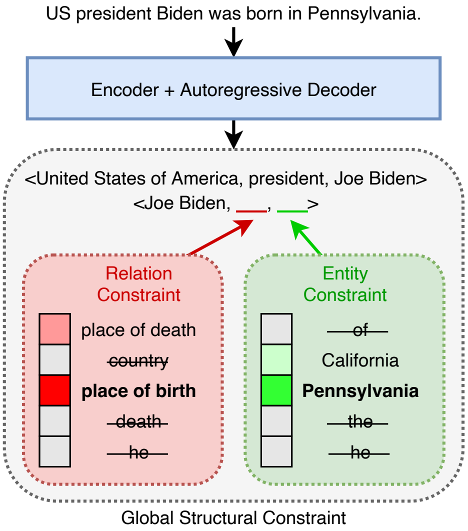
US president Biden was born in Pennsylvania.
Encoder + Autoregressive Decoder
<United States of America, president, Joe Biden>
<Joe Biden, ___, ___>
Relation Constraint
place of death
~~country~~
place of birth
~~death~~
~~he~~
Entity Constraint
~~of~~
California
Pennsylvania
~~the~~
~~he~~
Global Structural Constraint

Figure 1: **Overview of GenIE.** We use a transformer encoder-decoder model that takes unstructured text as input and autoregressively generates a structured semantic representation of the information expressed in it, in the form of (subject, relation, object) triplets. GenIE employs constrained beam search with: (i) a high-level constraint which asserts that the output corresponds to a set of triplets; (ii) lower-level constraints which use prefix tries to force the model to only generate valid entity or relation identifiers (from a predefined schema).

Traditionally, cIE was approached with pipelines that sequentially combine named entity recognition (Tjong Kim Sang, 2002), entity linking (Milne and Witten, 2008), and relation extraction (Miller et al., 1998). Entity linking and relation extraction serve as grounding steps, matching entities and relations to numerical identifiers in a KB, e.g., QIDs and PIDs for Wikidata. Recently, Trisedya et al. (2019) pointed out that such pipeline architectures suffer from the accumulation of errors and proposed an end-to-end alternative. Nevertheless, existing methods are still only practical for small schemas with unrealistically small numbers of relations and entities.

Alternatively, some works have focused on a simpler syntactic task: open information extraction (oIE), which produces free-form triplets from texts. In this setup, the entities and relations are not grounded in a KB and, usually, do not represent facts (Gashteovski et al., 2020). As oIE triplets contain only surface relations, they have ambiguous semantics, making them hard to use in downstream tasks (Broscheit et al., 2017) if not first aligned with a KB (Gashteovski et al., 2020). Since, in practice, oIE often consists of structured substring selection, it has recently been framed as an end-to-end sequence-to-sequence problem with great success (Huguet Cabot and Navigli, 2021; Dognin et al., 2021). Indeed, such autoregressive formulations can exploit the language knowledge already encoded in pre-trained transformers (Devlin et al., 2019). For example, some tokens can be more easily recognized as possible entities or relations thanks to the pre-training information.

Inspired by recent successes in oIE, we propose the first autoregressive end-to-end formulation of cIE that scales to many entities and relations, making cIE practical for more realistic KB schemas (i.e. schemas with millions of entities).[1] We employ a sequence-to-sequence BART model (Lewis et al., 2020), and exploit a novel bi-level constrained generation strategy operating on the space of possible triplets (from a fixed schema induced by Wikidata) to ensure that only valid triplets are generated. Our resulting model, *GenIE*, performs *Gen*erative *I*nformation *E*xtraction and combines the advantages of a known schema with an autoregressive formulation. The high-level overview of GenIE is provided in Fig. 1. The constrained generation encodes the known schema and enables the autoregressive decoder to generate textual tokens but only from the set of allowed entities or relations.

Contributions.

- We present the first end-to-end autoregressive formulation of closed Information Extraction.
- We describe a constrained decoding strategy that exploits the Wikidata schema to generate only valid fact triplets, demonstrating how constrained beam search can be applied on large, structured, and compositional spaces.
- Our model achieves state-of-the-art performance on the cIE task and scales to previously unmanageable numbers of entities (6M) and relations (more than 800).
- We point out and address weaknesses in the evaluation methodologies of recent previous works stemming from their small scale and the large imbalances in the available data per relation. We demonstrate the importance of reporting performance as a function of the number of relation occurrences in the data.
- We release pre-processed data, pre-trained models, and code within a general template designed to facilitate future research.

2 Background and Related Work

2.1 Closed Information Extraction

In this work, we address the task of closed information extraction (cIE), which aims to extract the exhaustive set of facts from natural language, expressible under the relation and entity constraints defined by a knowledge base (KB).

Most of the existing methods address the problem with a pipeline solution. One line of work starts by first extracting the entity mentions and the relations between them from raw text. This is followed by a disambiguation step in which the entity and relation predicates are mapped to their corresponding items in the KB. The sub-task of extracting the free-form triplets was originally proposed by Banko et al. (2007), and it is commonly referred to as open information extraction (oIE) or text-to-graph in the literature (Guo et al., 2020; Castro Ferreira et al., 2020; Huguet Cabot and Navigli, 2021; Dognin et al., 2021; Shen et al., 2015). Another line of work employs a pipeline of models for (i) named entity recognition (NER) – detecting the entity mentions; (ii) entity linking (EL) – mapping the mentions to specific entities from the KB; (iii) relation classification (RC) – detecting the relations that are expressed between the entities (Galárraga et al., 2014; Angeli et al., 2015b; Chaganty et al., 2017). Due to their architecture, pipeline methods are plagued by error propagation, which significantly affects their performance (Mesquita et al., 2019; Trisedya et al., 2019).

End-to-end systems that jointly perform the extraction and the disambiguation of entities and relations have been proposed to address the error propagation (Trisedya et al., 2019; Sui et al., 2021; Liu et al., 2018). To mitigate the propagation of errors,

[1] Note that current methods, due to the atomic classification, have high memory requirements, and suffer from performance deterioration as the number of entities or/and relations grows.

these systems are endowed with the ability to leverage entity information in the relation extraction and vice-versa, which has resulted in significant performance gains. Conceptually, for producing the output triplets, existing methods all rely on atomic, multi-class classification-based ranking of relations and entities. Classification methods particularly suffer from imbalances in the data. On the contrary, our proposition, GenIE, is auto-regressive and deals better with imbalances.

While cIE requires the constituent elements of the output triplets to be entities and relations associated with the KB, the output triplets in oIE are free-text. This makes the cIE task fundamentally harder than oIE, and renders the majority, if not all oIE methods, not applicable to the cIE setting. We report an additional discussion on relevant, but not fundamental, related work on oIE in Appendix A.

2.2 Autoregressive Entity Linking

The tasks of entity linking (EL) and entity disambiguation (ED) have been extensively studied in the past (Huang et al., 2015; Wu et al., 2020; Le and Titov, 2018; Kolitsas et al., 2018). Most existing approaches associate entities to unique atomic labels and cast the retrieval problem as multi-class classification across these labels. The match between the context and the label can then be represented as the dot product between the dense vector encodings of the input and the entity's meta information (Wu et al., 2020). This general approach has led to large performance gains.

Recently, De Cao et al. (2021a,b,c) have suggested that the classification-based paradigm for retrieval comes with several shortcomings such as (i) the failure to capture fine-grained interactions between the context and the entities; (ii) the necessity of tuning an appropriately hard set of negative samples during training. Building on these observations, they propose an alternative solution that casts the entity retrieval problem as one of autoregressive generation in which the entity names are generated token-by-token in an autoregressive fashion. The (freely) generated output will not always be a valid entity name, and to solve this problem De Cao et al. (2021b) propose a constrained decoding strategy that enforces this by employing a prefix trie. Their method scales to millions of entities achieving state-of-the-art performance on monolingual and multilingual entity linking.

Inspired by the intuition that language models are well suited for predicting entities, we propose a novel approach for cIE by framing the problem in an autoregressive generative formulation.

3 Method

In this section we formalize GenIE, an autoregressive end-to-end model for closed information extraction. Let's assume a knowledge base (KB) constituted from a collection of entities $\mathscr{E}$, a collection of relations $\mathscr{R}$, and a set of facts $\{(s,r,o) | (s,r,o) \in \mathscr{E} \times \mathscr{R} \times \mathscr{E}\}$ stored as (subject, relation, object) triplets. Additionally, we assume that each entity $e \in \mathscr{E}$ and relation $r \in \mathscr{R}$ is assigned to a textual label (corresponding to its name). The Wikidata KB (Vrandečić, 2012), with Wikipedia page titles as entity names, and the Wikidata relation labels as relation names satisfy these assumptions.

3.1 Model

We cast the task of information extraction as one of autoregressive generation. More concretely, given some text input x, GenIE generates the linearized sequence representation y of the exhaustive set of facts expressed in x. The conditional probability (parameterized by θ) assigned to the output y is computed in the autoregressive formulation: $p_\theta(y \mid x) = \prod_{i=1}^{|y|} p_\theta(y_i \mid y_{<i}, x)$. This can be seen as translating the unstructured text to a structured, unambiguous representation in a sequence-to-sequence formulation. GenIE employs the BART (Lewis et al., 2020) transformer architecture. It is trained to maximize the target sequence's conditional log-likelihood with teacher forcing (Sutskever et al., 2011, 2014), using the cross-entropy loss. We use dropout (Srivastava et al., 2014) and label smoothing for regularization (Szegedy et al., 2016).

3.2 Output Linearization

To represent the output with a sequence of symbols that is compatible with sequence-to-sequence architectures, we introduce the special tokens <sub>, <rel>, <obj> to demarcate the start of the subject entity, the relation type and the object entity for each triplet. The special token <et> is introduced to demarcate the end of the object entity, which is also the end of the triplet. We construct the sequence representation by concatenating the textual representations of its constituent triplets. While the sequence representation has an intrinsic notion of order, the output set of triplets does not. To mitigate the effects of this discrepancy, we enforce a consistent ordering of the target triplets during training. Concretely, whenever the triplets' entities are linked to the entity mentioned in the textual input, we consider first the triplets for which the subject entity appears earlier in the text. Ties are resolved by considering the appearance position of the object entity.

3.3 Inference with Constrained Beam Search

The space of triplets corresponds to $\mathscr{T} = \mathscr{E} \times \mathscr{R} \times \mathscr{E}$, and the target space, which consists of triplet sets of arbitrary cardinality, is equivalent to $\mathscr{S} = \bigcup_{i=0}^{\infty} [\mathscr{E} \times \mathscr{R} \times \mathscr{E}]^i$. At inference time, GenIE tackles the task of retrieving the linearized representation $y_S \in \mathscr{S}$ of a set of facts $S = \{t_1, \ldots, t_n\}$ constituted by triplets $t_i \in \mathscr{T}$ expressed in the input text x. Ideally, we would consider every element $y \in \mathscr{S}$ in the target space, assign it with a score $p_\theta(y \mid x)$ and retrieve the most probable one. Unfortunately, this is prohibitively expensive since we are dealing with a compositional target space whose size is gigantic (e.g., we consider a Wikidata entity catalog of $|\mathscr{E}| \approx$ 6M elements and a relation catalog of $|\mathscr{R}| \approx 1000$ relations, that can express a total of $|\mathscr{T}| \approx 10^{15}$ triplets; even if we limit ourselves to sentences that express only two facts, this provides us with $\approx 10^{30}$ different output options).

GenIE employs constrained Beam Search (BS; Sutskever et al., 2014; De Cao et al., 2021b) to resolve both of these problems. Instead of explicitly scoring all of the elements in the target space $\mathscr{S}$, the idea is to search for the top-k eligible options, using BS with k beams and a prefix trie. BS considers one step ahead – the next token to be generated – conditioned on the previous ones. The prefix trie restricts the BS to candidate tokens that could lead to valid identifiers. However, for the cIE setting we are interested in, the target space is prohibitively large to pre-compute the necessary trie. Therefore, we enforce a bi-level constraint on the output that allows for compositional, dynamic generation of the valid prefixes. More specifically, GenIE enforces: (i) a high-level structural constraint which asserts that the output follows the linearization schema defined in Sec. 3.2; (ii) lower level validity constraints which use an entity trie and a relation trie to force the model to only generate valid entity or relation identifiers, respectively – depending on the specific element of the structure that is being generated. This outlines a general approach for applying BS to search through large compositional structured spaces.

4 Experimental Setup

4.1 Knowledge Base: Wikidata

We use Wikidata[2] (Vrandečić, 2012) as the target KB to link to, filtering out all entities that do not have an English Wikipedia page associated with them. The filtering guarantees that all entity names are unique. Our final entity set $\mathscr{E}$ contains 5,891,959 items. We also define our relation set $\mathscr{R}$ as the super-set of all the relations considered in the datasets described below which consists of 857 relations. For different datasets, we consider only the subset of annotated relations to better compare with baselines. Although large, the number of entity (and relation) names is not a memory bottleneck as the generated prefix trie occupies ≈200Mb of storage (e.g., the entity linking system proposed by Wu et al. 2020 needs >20 times more storage).

4.2 Datasets and Evaluation Metrics

In this work, we further annotate and adapt REBEL (Huguet Cabot and Navigli, 2021) and Wiki-NRE for training, validation and testing. Additionally, we use Geo-NRE (Trisedya et al., 2019), and FewRel (Han et al., 2018) for testing purposes only. In Appendix B we report description of these datasets and their statistics. We measure standard micro and macro precision, recall and F1 for all settings (except for FewRel where only recall is a reliable metric). See Appendix C for a detailed and formal description of these metrics. We also report a 1 standard deviation confidence interval constructed from 50 bootstrap samples of the data.

4.3 Baselines

We compare GenIE against Set Generation Networks (SetGenNet; Sui et al., 2021) which is, to the best of our knowledge, the strongest model on Wiki-NRE and Geo-NRE. Note that authors did not release code or the model and there is no other model from the literature trained and evaluated on REBEL for cIE. SetGenNet (Sui et al., 2021) is an end-to-end state-of-the-art model for triplet extraction. It consists of a transformer encoder (Vaswani et al., 2017) that encodes the input followed by a non-autoregressive transformer decoder (Gu et al., 2018). The decoder generates embeddings that are

[2] Dumps from 2019/08/01

used to predict entities and relations. SetGenNet further uses candidate selection (Ganea and Hofmann, 2017; Kolitsas et al., 2018) to reduce the output space and a bipartite matching loss that handles different prediction orderings (i.e., it generates a set). Note that there are weaker baselines (e.g., Trisedya et al. 2019) we could have used to compare on REBEL but we were not able to reproduce their code. We report details on the effort made to use these baselines in Appendix D.

We also implement a pipeline baseline, consisting of 4 independent steps, namely: (i) *named entity recognition* (NER), which selects the spans in the input source likely to be entity mentions; (ii) *entity disambiguation* (ED), which links mentions to their corresponding identifiers in the KB; (iii) *relation classification* (RC), which predicts the relation between a given pair of entities, and finally; (iv) *triplet classification* (TC), which predicts whether a given triplet is actually entailed by the context. TC is necessary because the previous step (RC) predicts a relation for every pair of entities. Each step needs to be trained independently with a specific architecture tailored for the task, and we made an optimal choice for each step. For the NER component we used the state-of-the-art tagger FLAIR[3] (Akbik et al., 2019), while for ED we used the GENRE linker[4] (De Cao et al., 2021b). These two models were already trained, and we use them for inference only. For RC and TC, we trained a RoBERTa (Liu et al., 2019) model with a linear classification layer on top (as these two sub-tasks are typically cast as classification problems). Trisedya et al. (2019) also proposed many other pipeline baselines but ours outperforms them (see Table 5 in Appendix G for comparison).

5 Results

5.1 Performance Evaluation

Models performing cIE can base their predictions on different schemas. In this section, we distinguish between *small* and *large evaluation schema*. The *small evaluation schema* is consistent with previous approaches where models only have to decide between a small set of relations and entities (the schema induced by Wiki-NRE). In the *large evaluation schema*, models use the schema induced by REBEL. Models also use the large evaluation schema of REBEL when tested on FewRel, as a high-quality and challenging recall-based evaluation. We consider 3 training setups for GenIE and the pipeline baseline comprised of SotA components: (i) the training set of Wiki-NRE (W) only, (ii) the training set of REBEL (R) only, and (iii) pre-training on REBEL and fine-tuning on Wiki-NRE (R+W). The implementation details are given in Appendix E. We report the macro and micro precision, recall, and F1 in Table 1. Unfortunately, as the code for SetGenNet is not available, we cannot compute its macro performance, thus we report the micro scores only.

First, we observe a large and significant F1 improvement of 8 and 28 absolute points obtained by GenIE over SetGenNet and the pipeline baseline, respectively, when trained on the same dataset (W). Despite the much bigger schema employed by REBEL, pre-training on it and then fine-tuning (R+W), improves the performance on Wiki-NRE and Geo-NRE for 3% and 5%, respectively. This highlights that: (i) GenIE can effectively transfer knowledge across datasets/schemas; (ii) GenIE can quickly adapt to new schemas. Due to its rigid, monolithic relation classifier, the pipeline baseline does not possess these qualities. However, the pre-training does improve its macro scores.

Only the newly developed pipeline baseline and GenIE can scale-up to the larger schema[5], and as expected, this setting is more challenging for both models. However, GenIE still preserves a good F1 score of 68 micro and 34 macro, which is a relative increase of 60% and 320%, respectively, over the baseline. While the pipeline has a steeper drop from micro to macro scores, in general, a significant difference between the two is observed in every setting. This suggests that the models perform better for relations associated with many training examples and significantly worse for the rest. These findings call for the fine-grained analysis of performance in Sec. 5.2 that partitions the relations according to their occurrence count in the training data. For completeness, we also provide an analysis of performance as a function of the number of relations considered, in Appendix F.1.

Finally, on FewRel, recall is the only well-defined metric. In this setting as well, GenIE greatly outperforms the baseline for $11-13\%$ in both micro and macro recall, which are close to each other as the dataset is class-balanced.

[3] https://github.com/flairNLP/flair

[4] https://github.com/facebookresearch/GENRE

[5] See notes on reproducibility in Appendix D.

	Small Evaluation Schema						***Large Evaluation Schema***			
		Wiki-NRE			Geo-NRE			REBEL		FewRel
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Recall
Micro										
SetGenNet (W)	82.75 ± 0.11	77.55 ± 0.27	80.07 ± 0.27	86.89 ± 0.51	85.31 ± 0.47	86.10 ± 0.34	–	–	–	–
SotA Pipeline (W)	67.43 ± 0.28	54.22 ± 0.21	60.11 ± 0.22	64.60 ± 1.46	64.05 ± 1.46	64.32 ± 1.45	–	–	–	–
SotA Pipeline (R)	50.78 ± 0.20	62.17 ± 0.24	55.90 ± 0.20	60.28 ± 1.45	60.78 ± 1.49	60.53 ± 1.45	43.30 ± 0.15	41.73 ± 0.13	42.50 ± 0.13	17.89 ± 0.24
SotA Pipeline (R+W)	65.17 ± 0.27	54.40 ± 0.20	59.30 ± 0.21	66.65 ± 1.47	66.22 ± 1.46	66.43 ± 1.45	–	–	–	–
GenIE (W)	88.18 ± 0.13	88.31 ± 0.16	88.24 ± 0.13	86.46 ± 1.05	87.14 ± 1.03	86.80 ± 1.03	–	–	–	–
GenIE (R)	27.98 ± 0.13	67.16 ± 0.20	39.50 ± 0.14	39.69 ± 1.65	59.01 ± 1.56	47.45 ± 1.62	68.02 ± 0.15	69.87 ± 0.14	68.93 ± 0.12	30.77 ± 0.27
GenIE (R+W)	**91.39 ± 0.15**	**91.58 ± 0.14**	**91.48 ± 0.12**	**91.77 ± 0.98**	**93.20 ± 0.83**	**92.48 ± 0.88**	–	–	–	–
Macro										
SotA Pipeline (W)	11.96 ± 0.72	10.73 ± 0.46	10.56 ± 0.43	24.82 ± 3.61	22.54 ± 3.67	20.39 ± 2.72	–	–	–	–
SotA Pipeline (R)	19.39 ± 1.18	17.41 ± 0.99	15.93 ± 0.93	28.80 ± 3.86	30.24 ± 4.46	25.24 ± 3.21	12.20 ± 0.35	10.44 ± 0.22	9.48 ± 0.21	19.67 ± 0.26
SotA Pipeline (R+W)	24.12 ± 1.46	16.55 ± 1.00	17.76 ± 1.01	38.67 ± 5.72	34.49 ± 5.99	35.14 ± 5.09	–	–	–	–
GenIE (W)	44.22 ± 2.40	36.79 ± 1.62	38.39 ± 1.71	57.13 ± 6.83	52.83 ± 6.84	52.79 ± 6.27	–	–	–	–
GenIE (R)	30.63 ± 1.40	41.97 ± 1.92	29.27 ± 1.26	32.38 ± 5.86	40.39 ± 5.17	30.67 ± 5.23	33.90 ± 0.73	30.48 ± 0.65	30.46 ± 0.62	30.78 ± 0.26
GenIE (R+W)	**52.55 ± 2.12**	**45.95 ± 1.67**	**47.08 ± 1.68**	**75.77 ± 7.80**	**71.60 ± 7.95**	**72.59 ± 7.32**	–	–	–	–

Table 1: **Main results.** “R” indicates training on REBEL, and “W” indicates training on Wiki-NRE.

		REBEL		**FewRel**
	Precision	Recall	F1	Recall
Micro				
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
GenIE - GENRE	64.14 ± 0.14	76.58 ± 0.11	**69.81 ± 0.10**	**46.62** ± 0.25
GenIE unconstrained	65.30 ± 0.14	67.12 ± 0.12	66.20 ± 0.11	26.15 ± 0.27
Macro				
GenIE	**33.90 ± 0.73**	30.48 ± 0.65	30.46 ± 0.62	30.78 ± 0.26
GenIE - PLM	30.66 ± 0.68	**43.33** ± 0.63	**33.85** ± 0.58	**46.96** ± 0.25
GenIE - GENRE	32.02 ± 0.67	39.14 ± 0.68	**33.40** ± 0.62	**46.63** ± 0.24
GenIE unconstrained	32.25 ± 0.66	27.59 ± 0.53	28.20 ± 0.50	26.14 ± 0.24

Table 2: **Ablation study on the weights initialization and the constrained generation strategy.**

Ablation study. In Table 2 we summarize the results of an ablation study considering the pre-training and the constrained generation. We consider three different starting points: (i) a random initialization; (ii) BART (Lewis et al., 2020) pre-trained language model (PLM); (iii) a pre-trained autoregressive entity retrieval model GENRE (De Cao et al., 2021b). The pre-trained models are better in terms of recall and exhibit a better out-of-domain generalization on FewRel. In contrast, they are slightly worse in terms of precision, which translates to maximum improvement of a single point in F1 on REBEL. Another salient advantage of pre-training is reducing the training steps necessary for achieving good results. Indeed, when starting from GENRE, 3-5k steps are sufficient for competitive performance; stating from a PLM, necessitates 5-10k steps; while a random initialization requires 40-50k steps for competitive performance. Additionally, the pre-trained versions converge to a lower validation loss (see Fig. 5 in Appendix G).

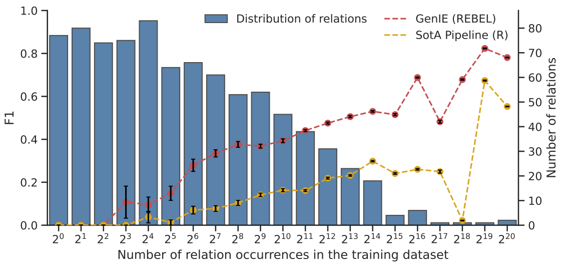

Figure 2: **Impact of the number of relation occurrences.** Relations are bucketed based on their number of occurrences; bucket 2^i contains relations occurring between 2^i and 2^{i+1} times. The histogram shows the number of relations per bucket. The line plots depict the F1 scores of GenIE and the baseline per bucket together with confidence intervals computed per bucket with bootstrap resampling.

In addition to ensuring a structure on the output, the constraint generation strategy results in an increase of 2-3 absolute points in terms of F1.

5.2 Analysis of Performance as a Function of the Relation Occurrence Count

The datasets naturally present large imbalances, where few relations occur a lot, but most relations are rare. In the previous section, we already observed a large difference between macro and micro F1 scores of models indicating that the number of occurrences impacts model performances. Thus, we now measure F1 scores after bucketing relations according to their number of occurrences in the training dataset. In Fig. 2, we create buckets ($i \in \{0, \ldots, 20\}$) where bucket i contains all the relations occurring at least 2^i times and less than 2^{i+1} times in the REBEL dataset. The height of the histogram for bucket i shows how many relations

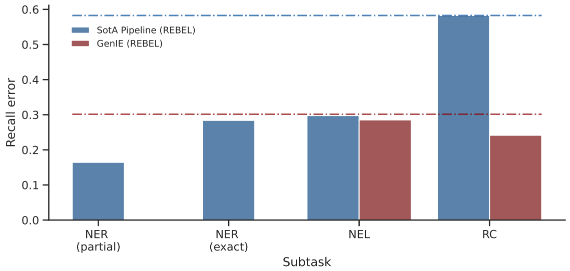

Figure 3: **Attribution of error to each of the cIE subtasks.** The dashed lines equal the overall recall error of the system. Lower is better.

are contained in this bucket. Finally, we report the F1 scores of GenIE and the pipeline of SotA components per bucket. Note that micro F1 from Table 1 is equivalent to putting all relations in one single bucket (equal weight to each data point), and macro F1 is equivalent to averaging the F1 with one bucket per relation (equal weight to each relation).

The histogram first confirms that most of the training data concern relations with few occurrences. Models thus need to perform few-shot learning for most of the relations. GenIE is significantly better than the pipeline baseline for all the buckets. Finally, it is important to highlight that even though the performance of both methods, unsurprisingly, declines for relations that appear less often in the training data, GenIE already performs well for relations with at least $2^6 = 64$ occurrences. On the contrary, the baseline needs $2^{14} = 16384$ samples to reach a comparable level of performance, and scores better than GenIE for the $2^6 = 64$ bucket, only after seeing at least $2^{19} = 524288$ samples. This confirms that GenIE is not only better at macro and micro F1, but it is also capable of performing fewer-shot learning than the baseline. It further shows that, contrary to the baseline, GenIE's good scores do not come solely from its ability to perform well on the few most frequent relations.

5.3 Disentangling the Errors

The task of cIE, explicitly or implicitly, encompasses NER, NEL and RC as its subtasks. Failure in any of them directly translates to failure on the original task. Therefore, to effectively compare different cIE models and accurately characterize their behavior, we need to evaluate their performance on each of the subtasks.

The separation of responsibility between the pipeline components leads to a natural error attribution for the SotA pipeline model. To estimate the NER error, we take the entity mentions predicted by the NER component and compare them with the corresponding mentions of the constituent entities of the output triplets. All triplets that concern an entity whose mention was not retrieved by the NER component are considered erroneous. We differentiate between two settings: (i) exact, which requires that the generated mention exactly matches the target mention; and (ii) partial, for which any overlap between the generated and the target triplet is sufficient. The NEL error is calculated by considering the output of the NEL component and comparing the linked entities to those in the output triplets. Again, all of the triplets that concern an incorrectly linked entity are considered erroneous. Finally, for the pipeline, every correctly predicted relation label translates to a correctly extracted triplet. Therefore, we calculate the RC error using the cIE definition of recall given in Appendix C.

End-to-end systems tackle all of the sub-tasks jointly, which makes the error attribution, in this setting, less obvious. To estimate errors corresponding to a particular target triplet, we need a reference triplet – among the predicted ones – for comparison. We start by outlining a bipartite matching procedure. Let each triplet be a node in a graph. We add an edge between each target-prediction pair of triplets. The edges are assigned a weight determined as a function of the pair of triplets they connect. Concretely, an edge e that connects the target triplet $t_t = (s_t, r_t, o_t)$ and the predicted triplet $t_p = (s_p, r_p, o_p)$ will be assigned a weight of: 1) if the triplets are the same; 2) if they express the same relation, but either the subject or the object differ; 3) if they concern the same entities, but the relation differs; 4) they share only a single entity; 5) they share only a single relation; 6) they have nothing in common. We construct the matching by selecting edges in a greedy fashion until all of the target triplets have been matched. The procedure ensures that every target triplet is paired with its closest match. Finally, we estimate the NEL error as the portion of edges that were assigned a weigh $w \in \{2,4,5,6\}$, and the RC error as the portion of edges that where assigned a weight $w \in \{3,4,6\}$.

The results of this analysis are summarized in Fig. 3. Immediately, the NER component in the pipeline method introduces an 18% error by completely missing on relevant entity mentions. An additional 12 absolute points hinge on a partial matching. The NEL component matches most of the entity mentions that are retrieved, but at this point, even with a (hypothetical) perfect RC, the performance of the pipeline will be only on par with GenIE. In practice, the RC component adds 30% to the inherited error, effectively doubling it.

On another note, the absolute error attributed to NEL by the pipeline and GenIE differs in a few absolute points only, while the difference for the (non-inherited) error stemming from RC is less than 10%. Adding these two together leaves us much shorter than the actual gap of 28 absolute points in performance on the cIE task. This suggests a strong correlation between the performance on NEL and RC for GenIE, which is fueled by the increased flow of information between the subtasks. The previous allows for more fine-grained interactions between the entities, the relations, and the context to be captured, consequently improving the overall performance. Alternatively, whenever the model captures a misleading correlation/interaction, it is amplified and hinders the performance on both subtasks. This result is echoed by the fact that the sum of the errors attributed to NEL and RC is significantly smaller than the error on the cIE task. Based on this observation, we hypothesize that any improvement on NEL will overflow to the RC subtask – and vice-versa – thereby directly translating to performance gains on the overall task.

6 Discussion

Unifying the cIE spectrum. There is a full spectrum of tasks that are closely related to cIE and are central to the field of information extraction. The typical setup assumes a KB associated with entities and relations, and the goal is to either annotate the text with information from the KB, or extract structured unambiguous information from the text. The tasks of entity linking and relation classification, already discussed in Sec. 2 and Sec. 4.3, are two such examples. Another example is slot filling (SF), the task of extracting information for a specific entity and relation (e.g., entity *Mick Jagger*, relation *member of*) from natural language (Surdeanu, 2013; Petroni et al., 2021).

All of these problems rely on the same set of logical tasks: identifying entities from the KB in text and understanding how they interact. Therefore, it would be beneficial to assume a single model, or a set of models that share parts of the weights and collectively solve all of the tasks. This would allow for the information from a dataset collected for one task (e.g., RC) to be leveraged for improving the performance of another (e.g., SF or cIE).

Bridging the gap between oIE and cIE. In this work, we only considered triplets for which both entities are element in the entity catalog. However, for many useful relations one of the objects is a literal (Mesquita et al., 2019), e.g., date of birth, length, size, number of employees or others. GenIE can be readily extended to accommodate for this, by adapting the decoding strategy allowing that for specific relations the entity can be a substring from the input. This is subtle connection to oIE which has thus far been treated as a separate problem. Current state-of-the-art methods on the oIE task address the problem in a similar autoregressive formulation (see Appendix A for more discussion).

Real world implications. Generative models have been shown to be very effective even in massive multilingual settings—e.g., De Cao et al. (2021c) proposed mGENRE, a multilingual version of GENRE trained and tested on more than 100 language. Our GenIE formulation would not need substantial modifications to adapt to such setting. Having a single model that works in hundreds of languages would be extremely useful and a very promising direction for future work.

While autoregressive models have a non-negligible computation footprint, De Cao et al. (2021a) show that autoregressive EL can be sped-up 70x, with no cost on performance. The fact that this solution can be adapted to GenIE makes the practical impact of our method even greater.

7 Conclusion

This paper provides a new view on closed information extraction (cIE) by casting the problem as autoregressive sequence-to-sequence generation. Our method, GenIE, leverages the autoregressive formulation to capture the fine-grained interactions expressed in the text and employs a bi-level constrained generation strategy to effectively retrieve the target representation from a large, structured, compositional predefined space of outputs. Experiments show that GenIE achieves state-of-the-art performance on cIE and can scale to a previously unmanageable number of entities and relations. We believe that our autoregressive formulation of cIE, coupled with constrained decoding, is a stepping stone towards a unified approach for addressing the core tasks in information extraction.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015a. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Gabor Angeli, Victor Zhong, Danqi Chen, Arun Tejasvi Chaganty, Jason Bolton, Melvin Jose Johnson Premkumar, Panupong Pasupat, Sonal Gupta, and Christopher D. Manning. 2015b. Bootstrapped self training for knowledge base population. In *Proceedings of the 2015 Text Analysis Conference*.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, page 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Samuel Broscheit, Kiril Gashteovski, and Martin Achenbach. 2017. OpenIE for Slot Filling at TAC KBP 2017 - System Description. In *proceedings of the Text Analysis Conference*. NIST.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Arun Chaganty, Ashwin Paranjape, Percy Liang, and Christopher D. Manning. 2017. Importance sampling for unbiased on-demand evaluation of knowledge base population. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1038–1048, Copenhagen, Denmark. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

Luciano Del Corro and Rainer Gemulla. 2013. ClausIE: clause-based open information extraction. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 355–366. International World Wide Web Conferences Steering Committee / ACM.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021a. Highly parallel autoregressive entity linking with discriminative correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7662–7669, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021b. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2021c. Multilingual autoregressive entity linking. *ArXiv preprint*, abs/2103.12528.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Pierre Dognin, Inkit Padhi, Igor Melnyk, and Payel Das. 2021. ReGen: Reinforcement learning for text and knowledge base generation using pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1084–1099, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. 2014. Canonicalizing open knowledge bases. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 1679–1688. ACM.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark. Association for Computational Linguistics.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. MinIE: Minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640, Copenhagen, Denmark. Association for Computational Linguistics.

Kiril Gashteovski, Rainer Gemulla, Bhushan Kotnis, Sven Hertling, and Christian Meilicke. 2020. On aligning OpenIE extractions with knowledge bases: A case study. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 143–154, Online. Association for Computational Linguistics.

Adam Grycner and Gerhard Weikum. 2016. POLY: Mining relational paraphrases from multilingual sentences. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2183–2192, Austin, Texas. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Qipeng Guo, Zhijing Jin, Xipeng Qiu, Weinan Zhang, David Wipf, and Zheng Zhang. 2020. CycleGT: Unsupervised graph-to-text and text-to-graph generation via cycle training. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 77–88, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Hongzhao Huang, Larry Heck, and Heng Ji. 2015. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *ArXiv preprint*, abs/1504.07678.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158, Portland, Oregon, USA. Association for Computational Linguistics.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2021. A survey on knowledge graphs: Representation, acquisition and applications. *IEEE transactions on neural networks and learning systems*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Phong Le and Ivan Titov. 2018. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Yue Liu, Tongtao Zhang, Zhicheng Liang, Heng Ji, and Deborah L. McGuinness. 2018. Seq2RDF: An End-to-end Application for Deriving Triples from Natural Language Text. In *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, October 8th - to - 12th, 2018*.

Filipe Mesquita, Matteo Cannaviccio, Jordan Schmidek, Paramita Mirza, and Denilson Barbosa. 2019. KnowledgeNet: A benchmark dataset for knowledge base population. 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)*, pages 749–758, Hong Kong, China. Association for Computational Linguistics.

Scott Miller, Michael Crystal, Heidi Fox, Lance Ramshaw, Richard Schwartz, Rebecca Stone, and Ralph Weischedel. 1998. Algorithms that learn to extract information BBN: TIPSTER phase III. In *TIPSTER TEXT PROGRAM PHASE III: Proceedings of a Workshop held at Baltimore, Maryland, October 13-15, 1998*, pages 75–89, Baltimore, Maryland, USA. Association for Computational Linguistics.

David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, page 509–518, New York, NY, USA. Association for Computing Machinery.

Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. 2012. PATTY: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1135–1145, Jeju Island, Korea. Association for Computational Linguistics.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8528–8535.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Dianbo Sui, Chenhao Wang, Yubo Chen, Kang Liu, Jun Zhao, and Wei Bi. 2021. Set generation networks for end-to-end knowledge base population. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9650–9660, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mihai Surdeanu. 2013. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. *Theory and Applications of Categories*.

Ilya Sutskever, James Martens, and Geoffrey E. Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 1017–1024. Omnipress.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. Let's Stop Incorrect Comparisons in End-to-end Relation Extraction! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3689–3701, Online. Association for Computational Linguistics.

Jizhi Tang, Yansong Feng, and Dongyan Zhao. 2019. Learning to update knowledge graphs by reading news. 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)*, pages 2632–2641, Hong Kong, China. Association for Computational Linguistics.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240, Florence, Italy. Association for Computational Linguistics.

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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Denny Vrandečić. 2012. Wikidata: A new platform for collaborative data collection. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, page 1063–1064, New York, NY, USA. Association for Computing Machinery.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30, Online. Association for Computational Linguistics.

A Additional Background and Related Work

A.1 Generative Open Information Extraction

Early work had focused on pipeline architecture for oIE. In general, these methods first detect the entity mentions present in the text and then, for pair of entities, in a classification setting, predict the existence of a relation between the two entities and the relation type (Angeli et al., 2015a; Corro and Gemulla, 2013). The advent of transformers (Devlin et al., 2019; Lan et al., 2020; Liu et al., 2019) and pipeline architectures that allow for information to flow between the two subtasks – usually by sharing some parameters of the encoder – have allowed these models to do well on the oIE task. However, they do come with some general limitations: (i) assuming the existence of a single relation between a pair of entities; (ii) inability to capture the interactions between triplets.

Much of the current research is focused on studying the oIE problem in the autoregressive generative setting, which seamlessly mitigates the limitations mentioned above (Huguet Cabot and Navigli, 2021; Dognin et al., 2021; Nayak and Ng, 2020). For instance, ReGen (Dognin et al., 2021) significantly improves upon published results and establishes state-of-the-art results on the dataset used in the WebNLG 2020+ Challenge (Castro Ferreira et al., 2020). REBEL (Huguet Cabot and Navigli, 2021), on the other hand, achieves state-of-the-art performances across a suite of oIE benchmarks. Moreover, both of these methods address the problem in a similar formulation that takes the text as input context and generates the output triplets token-by-token in an autoregressive fashion.

The output triplets in oIE are free-text, while cIE requires the constituent elements of the output triplets to come from the entity and relation sets associated with the KB. This makes the cIE task fundamentally harder than oIE, and renders these methods not applicable to the cIE setting.

Finally, Taillé et al. (2020) make an effort to describe the many issues with the evaluation of oIE systems in literature and call for a unified evaluation setting for a fair comparison between systems. These problems get only exacerbated in cIE where the performance of a model would highly depend on the entity and relation catalogue considered. To alleviate some of these issues, we annotate the REBEL dataset (Huguet Cabot and Navigli, 2021) with unique textual entity identifiers and textual relation labels, and propose a suite of meaningful evaluation settings while considering an approximately 6 million long entity catalogue comprised of all the entities in the English Wikipedia, and 857 long relation catalogue supported by the dataset.

B Datasets

Table 3 summarizes the statistics of all datasets used in this work. For each dataset, we remove datapoints containing triplets with entities that do not have an associated Wikipedia page (i.e., entities not associated to a unique name). This filtering removes a negligible portion of the data in most cases (i.e., <0.5%) except for REBEL where 3.4% of datapoints were removed.

We evaluate the models in a standard setups for Wiki-NRE and Geo-NRE. For these datasets, the schema is unrealistically small: ≈300K entities with 157 relations for Wiki-NRE and 124 entities with 11 relations for Geo-NRE. Therefore, we scale to previously unexplored schema sizes for cIE using the REBEL dataset (≈6M entities and 857 relations). We also use FewRel as a high-quality dataset for recall evaluation using the large schema from REBEL.

REBEL (Huguet Cabot and Navigli, 2021) is a dataset created from Wikipedia abstracts. It consists of an alignment between sentences, Wikipedia hyperlinks and their corresponding Wikidata entities, and relations. REBEL proposed an alignment expanding on Elsahar et al. (2018), a pipeline of mention detection, coreference resolution, entity disambiguation and then mapping triplets to each sentence. Huguet Cabot and Navigli (2021) further filtered false positives using an Natural Language Inference (NLI) model to check if the relation was truly entailed by the text. In this setting, we consider the full ≈6M long entity and 857 long relation catalog. We use this dataset for both training and testing. Additionally, we employ REBEL to analyze the performance as a function of the number of relations, by simulating different environments pertaining to subsets of the top-n most frequent relations.

Wiki-NRE (Trisedya et al., 2019) is a dataset created from Wikipedia. Authors aligned hyperlinks to Wikidata entities as in REBEL but they applied a different filtering: they (i) extracted sentences that contain implicit entity names using co-reference resolution (Clark and Manning, 2016), and (ii) they

Dataset	Documents			Triplets			$\lvert\mathscr{E}\rvert^{\dagger}$	$\lvert\mathscr{R}\rvert^{\dagger}$
	training	validation	test	training	validation	test		
REBEL	1,899,331	104,960	105,516	5,147,836	284,268	284,936	1,498,143	857
Wiki-NRE	223,536	980	29,619	298,489	1,317	39,678	278,204	157
Geo-NRE	–	–	1,000	–	–	1,000	124	11
FewRel	–	26,892*	27,650	–	26,892*	27,650	64,762	80

Table 3: **Statistics of the datasets.** †With an abuse of notation here we indicate the amount of unique entities and relations for each dataset and not the size of the Knowledge Base associated with it (see Section 4 for more details). * Note that we do not use the validation FewRel data in our experiment, but we release this split as well.

filtered and assigned relations to sentences using paraphrase detection from different sources (Nakashole et al., 2012; Ganitkevitch et al., 2013; Grycner and Weikum, 2016). We used this dataset for both training and testing.

Geo-NRE (Trisedya et al., 2019) is constructed in the same way as Wiki-NRE but from a collection of user reviews on 100 popular landmarks in Australia, instead of Wikipedia. Due to its small size and to compare with the literature, we used this dataset only for testing.

FewRel (Han et al., 2018) is also extracted from Wikipedia where Wikidata is the KB. Contrary to the other datasets, FewRel does not provide distant supervision but it is fully annotated by humans. The dataset was first automatically constructed and then filtered as annotators were asked to judge whether the relations are explicitly expressed in the sentences. Each input in FewRel is associated with a single triplet only, and not all of the triplets entailed by it. Therefore, this dataset can be used for precisely measuring recall (but not precision or F1). We employ it only for testing. To simulate a more realistic scenario, we train the models on many relations, and leverage the high quality FewRel data to calculate the performance metrics for the subset of relations annotated.

C Performance Metrics

We measure standard precision, recall and F1 for all settings. A fact is regarded as correct if the relation and the two corresponding entities are all correct. More precisely, we denote the set of all predicted triplets of a document $d \in \mathscr{D}$ as P_d, and the set of gold triplets as G_d. Then:

$$\text{micro-precision} = \sum_{d\in\mathscr{D}} |P_d \cap G_d| \Big/ \sum_{d\in\mathscr{D}} |P_d| \,, \quad (1)$$

and

$$\text{micro-recall} = \sum_{d\in\mathscr{D}} |P_d \cap G_d| \Big/ \sum_{d\in\mathscr{D}} |G_d| \,. \quad (2)$$

Micro scores are useful for measuring the overall performance of a model but they are less informative for imbalanced datasets (e.g., when some entities or relations are disproportionately more present in both training and test sets). Indeed, micro scores assign equal weight to every sample while macro scores assign equal weight to every class. Thus, we also measure macro scores by aggregating per relation type. If we denote $P_d^{(r)}$ and $G_d^{(r)}$ as the predicted and gold set only containing the relation $r \in \mathscr{R}$ of a document d, then macro-precision is defined as:

$$\frac{1}{\mathscr{R}} \sum_{r\in\mathscr{R}} \left(\sum_{d\in\mathscr{D}} |P_d^{(r)} \cap G_d^{(r)}| \Big/ \sum_{d\in\mathscr{D}} |P_d^{(r)}| \right) \,, \quad (3)$$

and macro-recall as:

$$\frac{1}{\mathscr{R}} \sum_{r\in\mathscr{R}} \left(\sum_{d\in\mathscr{D}} |P_d^{(r)} \cap G_d^{(r)}| \Big/ \sum_{d\in\mathscr{D}} |G_d^{(r)}| \right) \,. \quad (4)$$

D Note on End-to-End Baselines

We invested a considerable amount of time trying to use a strong end-to-end baseline to compare GenIE with. Unfortunately, most works do not have available or directly usable code. In particular, we first concentrated on SetGenNet (Sui et al., 2021) as, to the best of our knowledge, it is the strongest model on the task of cIE. However, the authors do not report a link to the code[6] in neither the arXiv nor the ACL Antology version of the paper. We could not find any related repository on GitHub either. For these reasons we were unable to use their method as a baseline for REBEL.

[6] As of October 2021.

We then focused on the work most similar to SetGenNet, that is the system proposed by Trisedya et al. (2019). They released code and we were able to run it. However, the code was incomplete: they included code for training only a part of their system. They start with pre-trained word, entity and relation embeddings, but did not release code for pre-training them. The closest solution we found was using Wikipedia2Vec (Yamada et al., 2020), which does not include relation embeddings. Besides, the pre-trained word embeddings on the official Wikipedia2Vec website[7] do not match the dimensionality used by Trisedya et al. (2019). Finally, the authors did not include code to train the "triple classifier" of their model. The classifier is instead directly loaded in their code. For these reasons we were unable to use their method as a baseline for REBEL.

E Implementation Details

Data. The train, test and validation splits are either inherited from the original dataset (see Appendix B for details) or sampled at random. To facilitate reproducibility, we release the exact splits employed in our experiments.

Additionally, we release the curated entity and relation catalogs for both the large and the small schema, in which the redirects have been resolved and each of the QID/PID is paired with a unique, semantically meaningful textual identifier. We hope that this will allow for a fair comparison of future work in which the same evaluation setup can be maintained.

E.1 GenIE

Infrastructure. For training we used a single machine with 24 Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz processor cores and 441 GB of RAM, equipped with 4 Tesla V100-PCIE-16GB GPUs.

Training. The models were trained using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 3e-5, 0.1 gradient clipping and a varying weight decay (cf. Table 4). The learning rate is updated using a polynomial decay schedule with an end value of 0. While most of the parameters were left at their default values for BART, the rest were tuned on the respective datasets' validation set, and their corresponding optimal values are given in Table 4.

Inference. At test time, we use Constrained Beam Search with 10 beams. We restrict the input and the output sequence to be at most 256 tokens, cutting from the right side if the input is too long. We normalize the log-probabilities by sequence length, and allow for any number of n-gram repetition. The other parameters are kept to their default values for inference with BART.

E.2 SotA Pipeline

We described our SotA pipeline system baseline in Sec. 4.3. We release code to both train and run inference with the proposed pipeline. The named entity recognition and the entity disambiguation components were not trained. The relation classification module is a linear layer on top of RoBERTa (Liu et al., 2019). We trained it learning rate 3e-4 using the Adam optimizer (Kingma and Ba, 2015). We trained for a maximum number of steps using early stopping on the validation sets. We restrict the input sequence to be at most 128 tokens cutting from the right side if the input is too long. All other hyperparameters are reported in Table 4. The triple classification module is also a linear layer on top of RoBERTa (Liu et al., 2019) with the same hyperparameters of the relation classification module but we trained for less steps.

F Additional Experiments

F.1 Analysis of Performance as a Function of the Number of Relations

Previous works focus on small schemas meaning that few relations were considered. Indeed, classification problems on a large set of possible classes become particularly difficult under large class imbalances, which is the case here as shown by Fig. 2. However, scaling up to larger schemas with more relations is crucial for the models to be useful in downstream tasks. To measure the scaling ability of GenIE, we create different setups with variable numbers of relations. To create such setups, we start with the REBEL dataset and schema (857 relations) and choose subsets of relations with their associated training data. In Fig. 4, we report GenIE and the pipeline baseline F1 for schemas with 100, 400, and 857 relations. To choose a subset of *n* relations, we take the *n* most frequent relations to mimic the strategies used by previous works to reduce the schemas (Sui et al., 2021).

[7] https://wikipedia2vec.github.io/wikipedia2vec/pretrained

	Max steps	Warm-up steps	Batch size	Dropout	Weight decay	Training time
GenIE (W)	60,000	1,000	32	0.1	0.01	0.5 GPU days
GenIE (R)	100,000	5,000	384	0.1	0.01	18.5 GPU days
GenIE (R + W)	100,000	5,000	384	0.1	0.01	20.5 GPU days
GenIE - Genre	50,000	3,000	2,048	0.3	0.50	11 GPU days
GenIE - PLM	50,000	3,000	2,048	0.3	0.50	17 GPU days
SoTA Rel-class (W)	20,000	500	128	0.1	0.01	0.2 GPU days
SoTA Rel-class (R)	250,000	500	128	0.1	0.01	2.5 GPU days
SoTA Rel-class (R + W)	250,000	500	128	0.1	0.01	2.5 GPU days
SoTA Tri-class (R)	50,000	500	128	0.1	0.01	0.3 GPU days
SoTA Tri-class (W)	5,000	500	128	0.1	0.01	0.1 GPU days
SoTA Tri-class (R + W)	50,000	500	128	0.1	0.01	0.3 GPU days

Table 4: **Hyperparameters for the different models.**

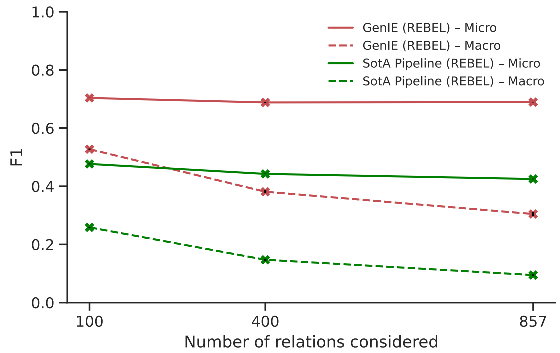

Figure 4: **Impact of the number of relations in the schema on REBEL.** Micro and macro F1 of both GenIE and the pipeline of SotA components for 3 schema sizes: 100, 400, and 857 relations. The schema is constrained at both training and testing time. Full results (i.e., precision and recall) are reported in Table 6 in Appendix G.

We first observe that GenIE is always largely better than the baseline. The baseline suffers from the same difficulty as previous works; classifying among a large set of relations is hard with large imbalances. GenIE and the baseline have similar absolute decrease in performance when the number of relations increases, corresponding to a more considerable relative decrease for the baseline. More concretely, GenIE's micro F1-score goes from 70.36 % for the top 100 relations, to 68.82 % and 68.93 % for the top 400 and 857 relation setups, respectively. This translates to a relative decrease of 2 % only in the first step. For the baseline, the absolute score of 47.67 % first falls to 44.25 % and subsequently to 42.5 % as the number of relations grows. This in turn, is an overall relative drop of almost 11 %.

Notably, when looking at precision and recall separately (cf. Table 6 in Appendix G), GenIE has a slight proportional decrease of 1-2 absolute points, both in precision and recall, which reflects the increased difficulty of the task due to larger number of relations. The baseline exhibits a similar drop in precision, but a much more significant drop in the recall of almost 10 absolute or 16 point relative. This suggests that the baseline simply ignores most of the relations with lower occurrence counts, which is consistent with the results in Sec. 2, and the hypothesis that the relation classification task is a bottleneck for effectively scaling the baseline system to a large number of relations.

We already have to deploy several techniques to help the baseline better deal with these issues (see Sec. 4.3), while GenIE, thanks to its generative autoregressive formulation, can effectively scale and manage the inherent imbalances of the task much more naturally.

G Additional Results

	Wiki-NRE			Geo-NRE		
	Precision	Recall	F1	Precision	Recall	F1
Pipeline baselines						
AIDA + MinIE	36.72	48.56	41.82	35.74	39.01	37.30
NeuralEL + MinIE	35.11	39.67	37.25	36.44	38.11	37.26
AIDA + ClauseIE	36.17	47.28	40.99	35.31	39.51	37.29
NerualEL + ClauseIE	34.45	37.86	36.07	35.63	37.91	36.73
AIDA + CNN	40.35	35.03	37.50	37.15	31.65	34.18
NeuralEL + CNN	36.89	35.21	36.03	37.81	30.05	33.49
Encoder-decoder baselines						
Single Attention	45.91	38.36	41.80	40.10	39.12	39.60
Single Attention (+pre-trained)	47.25	40.53	43.63	43.14	43.11	43.12
Single Attention (+beam)	60.56	52.31	56.13	58.69	48.51	53.12
Single Attention (+triplet classifier)	**73.78**	50.13	59.70	67.04	53.01	59.21
Transformer	46.28	38.97	42.31	45.75	46.20	45.97
Transformer (+pre-trained)	47.48	40.91	43.95	48.41	48.31	48.36
Transformer (+beam)	58.29	50.25	53.97	61.81	61.61	61.71
Transformer (+triplet classifier)	73.07	48.66	58.42	**71.24**	57.61	63.70
Our pipeline baseline	67.43	**54.22**	**60.11**	64.60	**64.05**	**64.32**

Table 5: **Baselines comparison.** All results are taken from from Trisedya et al. (2019). Encoder-decoder baseline are proposed by the authors and other pipeline baseline include an NER and an ED system AIDA (Hoffart et al., 2011) or NeuralEL (Kolitsas et al., 2018) and then a relation extraction system CNN (Lin et al., 2016), MiniE (Gashteovski et al., 2017), or ClausIE (Corro and Gemulla, 2013). Best results are highlighted in **bold** and second best are underlined. Our pipeline baseline scores the best or on pair among these other methods.

	REBEL (top 100 Relations)			REBEL (top 400 Relations)			REBEL (857 Relations)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Micro									
GenIE	68.76 ± 0.12	72.05 ± 0.13	70.36 ± 0.10	67.10 ± 0.13	70.62 ± 0.15	68.82 ± 0.12	68.02 ± 0.15	69.87 ± 0.14	68.93 ± 0.12
SotA Pipeline	44.76 ± 0.17	50.99 ± 0.17	47.67 ± 0.16	38.98 ± 0.13	51.18 ± 0.12	44.25 ± 0.11	43.30 ± 0.15	41.73 ± 0.13	42.50 ± 0.13
Micro									
GenIE	52.26 ± 0.25	54.13 ± 0.27	52.75 ± 0.24	41.50 ± 0.66	38.53 ± 0.57	38.12 ± 0.51	33.90 ± 0.73	30.48 ± 0.65	30.46 ± 0.62
SotA Pipeline	27.41 ± 0.27	31.05 ± 0.18	25.87 ± 0.15	16.94 ± 0.63	19.00 ± 0.36	14.73 ± 0.37	12.20 ± 0.35	10.44 ± 0.22	9.48 ± 0.21

Table 6: **Impact of the number of relations in the schema on REBEL.** The schema is constrained at both training and testing time.

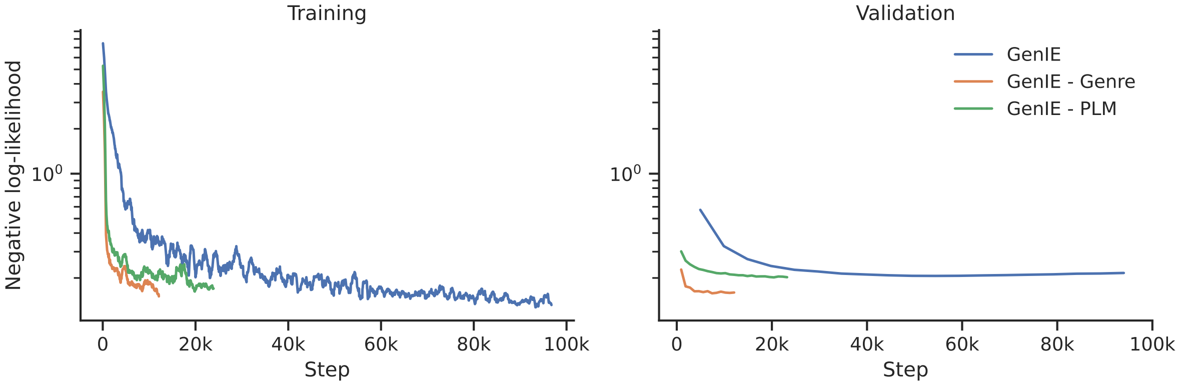

Figure 5: **Training and validation loss curves for different initialization of our model.** GenIE starts from a random initialization, GenIE – PLM fine-tunes a BART pre-trained language model, while GenIE - GENRE is initialized with a pre-trained autoregressive entity linking model by De Cao et al. (2021b).