Seq2rel: A sequence-to-sequence-based approach for document-level relation extraction

Anonymous ACL Rolling Review submission

Abstract

Motivated by the fact that many relations cross the sentence boundary, there has been increasing interest in document-level relation extraction (RE). Document-level RE requires integrating information within and across sentences, capturing complex interactions between mentions of interacting entities. Most document-level RE methods proposed to date are pipeline-based, requiring entities as input. However, previous work has demonstrated that jointly learning to extract entities and relations can improve performance and be more efficient due to shared parameters and training steps. In this paper, we develop a sequence-to-sequence-based approach that can learn the sub-tasks of document-level RE — entity extraction, coreference resolution and relation extraction — in an end-to-end fashion. We evaluate our approach on several datasets, in some cases exceeding the performance of existing methods. Finally, we demonstrate that, under our model, the end-to-end approach outperforms a pipeline-based approach. [1]

1 Introduction

PubMed, the largest repository of biomedical literature, contains over 30 million publications and is adding more than one paper per minute (Church, 2017). Accurate, automated text mining and natural language processing (NLP) methods are needed to maximize discovery and unlock structured information from this massive volume of text. An important step in this process is relation extraction (RE), the task of identifying groups of entities within some text that participate in a semantic relationship. In the domain of biomedicine, relations of interest include chemical-induced disease, protein-protein interactions, and gene-disease associations.

Many methods have been proposed for RE, ranging from rule-based to machine learning-based (Zhou et al., 2014). Most of this work has focused on *intra*-sentence binary RE, where pairs of entities within a sentence are classified as belonging to a particular relation (or none). These methods often ignore commonly occurring complexities like nested or discontinuous entities, coreferent mentions (words or phrases in the text that refer to the same entity), inter-sentence and n-ary relations (see Table 1 for examples). The decision not to model these phenomena is a strong assumption. In GENIA (Kim et al., 2003), a corpus of PubMed articles labelled with around 100,000 biomedical entities, ~17% of all entities are nested within another entity. Discontinuous entities are particularly common in clinical text, where ~10% of mentions in popular benchmark corpora are discontinuous (Wang et al., 2021). In the CDR corpus (Li et al., 2016b), which comprises 1500 PubMed articles annotated for chemical-induced disease relations, ~30% of all relations cross sentence boundaries. In Peng et al. (2017), including inter-sentence relations when deploying an RE system on PubMed for large-scale extraction tripled the yield. Some relations, like drug-gene-mutation interactions, are difficult to model with binary RE (Zhou et al., 2014).

In response to some of these shortcomings, there has been a growing interest in *document*-level RE. Document-level RE aims to model *inter*-sentence relations between (potentially coreferent) mentions of entities in a document. A popular approach involves graph-based methods, which have the advantage of naturally modelling inter-sentence relations (Peng et al., 2017; Song et al., 2018; Christopoulou et al., 2019; Nan et al., 2020; Minh Tran et al., 2020). However, like all pipeline-based approaches, these methods assume that the entities within the text are known. As previous work has demonstrated—and as we show in §5.2—jointly learning to extract entities and relations can improve performance (Miwa and Sasaki, 2014;

[1] Our code and models will be made publicly available.

Table 1: Examples of complexities in entity and relation extraction and the proposed linearization schema to model them. `CID`: chemical-induced disease. `GDA`: gene-disease association. `DGM`: drug-gene-mutation.

Complexities	Example	Comment
Discontinuous mentions	Induction by **paracetamol**$_{\text{DRUG}}$ of **[bladder** and **liver tumours]**$_{\text{DISEASE}}$.	Discontinuous mention of **bladder tumours**.
	`paracetamol @DRUG@ bladder tumours @DISEASE@ @CID@` `paracetamol @DRUG@ liver tumours @DISEASE@ @CID@`	
Coreferent mentions	Proto-oncogene **HER2**$_{\text{GENE}}$ (also known as **erbB-2**$_{\text{GENE}}$ or **neu**$_{\text{GENE}}$) plays an important role in the carcinogenesis and the prognosis of **breast cancer**$_{\text{DISEASE}}$.	Two coreferent mentions of **HER2**.
	`her2 ; erbb-2 ; neu @GENE@ breast cancer @DISEASE@ @GDA@`	
n-ary, inter-sentence	The deletion mutation on exon-19 of **EGFR**$_{\text{GENE}}$ gene was present in 16 patients, while the **L858E**$_{\text{MUTATION}}$ point mutation on exon-21 was noted in 10. All patients were treated with **gefitinib**$_{\text{DRUG}}$ and showed a partial response.	Ternary **DGM** relationship crosses a sentence boundary.
	`gefitinib @DRUG@ egfr @GENE@ l858e @MUTATION@ @DGM@`	

Miwa and Bansal, 2016; Gupta et al., 2016; Li et al., 2016a, 2017; Nguyen and Verspoor, 2019a; Yu et al., 2020) and may be more efficient due to shared parameters and training steps. Ideally, a document-level RE system would be capable of modelling the complexities we have discussed without strictly requiring entities to be known. End-to-end methods typically combine task-specific components for entity detection, coreference resolution, and relation extraction that are trained jointly. Most approaches are restricted to intra-sentence RE (Bekoulis et al., 2018; Luan et al., 2018; Nguyen and Verspoor, 2019b; Wadden et al., 2019) and have only recently been extended to document-level RE (Eberts and Ulges, 2021). However, they still focus on binary relations. A less popular, end-to-end approach is to frame RE as a sequence-to-sequence (seq2seq) task (Sutskever et al., 2014). If the information to extract is appropriately linearized to a string, seq2seq-based methods are flexible enough to model all the complexities discussed thus far. However, existing work stops short, focusing on intra-sentence binary relations (Zeng et al., 2018; Zhang et al., 2020; Nayak and Ng, 2020; Zeng et al., 2020). In this paper, we extend the work on seq2seq methods to document-level RE with several important contributions:

- We propose a novel linearization schema that can handle complexities overlooked by previous seq2seq-based approaches, like coreferent mentions and n-ary relations (§3.1).
- Using a strategy we call "entity hinting", we demonstrate that our model can be used to perform document-level RE in a pipeline-like setup when entities are known (§3.3).
- When given only the raw text, we demonstrate that our model is able to learn the sub-tasks of document-level RE — entity extraction, coreference resolution and relation extraction — jointly, sharing all parameters across the tasks.
- We evaluate our model on several datasets, in some cases exceeding the performance of existing document-level RE systems (§5.1).

2 Task definition: document-level relation extraction

Given a document of S tokens[2], a model must extract all tuples corresponding to a relation, R, expressed between the entities, E in the document, $(E_1, ..., E_n, R)$ where n is the arity, or the number of participating entities, in the relation. Each entity E_i is represented as the set of its coreferent mentions $\{e_j^i\}$ in the document, which are often expressed as aliases, abbreviations or acronyms. All entities appearing in a tuple have at least one mention in the document. The mentions that express a given relation are not necessarily contained within the same sentence. Commonly, E is assumed to be known and provided as input to a model. We will refer these methods as "pipeline-based". In this paper, we are primarily concerned with the situation where E is *not* given, and must be predicted by a model, which we will refer to as "end-to-end".

3 Our approach: seq2seq learning

3.1 Linearization

To use seq2seq learning for RE, the information to be extracted must be linearized to a target string.

[2] S stands for *source* tokens, to distinguish them from *target* tokens, T. See §3.2.

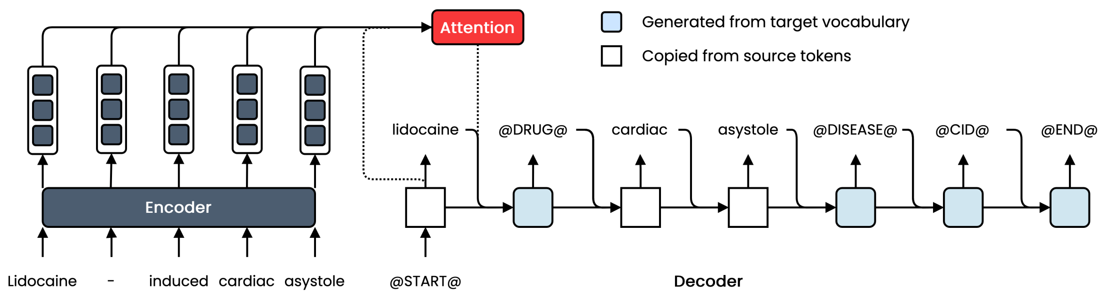

Figure 1: A seq2seq model for document-level relation extraction. Special tokens are generated by the decoder. Entity mentions are copied from the input via a copy mechanism. Decoding is initiated by a `@START@` token and terminated when the model generates the `@END@` token. Attention connections shown only for the second timestep to reduce clutter. `CID`: chemical-induced disease.

There are several desiderata for this linearization. It should be expressive enough to model the complexities of entity and relation extraction without being overly verbose. We propose the following schema, illustrated with an example:

X: Variants in the **estrogen receptor alpha** (**ESR1**) gene and its mRNA contribute to risk for **schizophrenia**.
Y: `estrogen receptor alpha ; ESR1 @GENE@ schizophrenia @DISEASE@ @GDA@`

The input text, X, expresses a gene-disease association (GDA) between **ESR1** and **schizophrenia**. In the corresponding target string Y, each relation begins with its constituent entities. A semicolon separates coreferent mentions (`;`), and entities are terminated with a special token denoting their type (e.g. `@GENE@`). Similarly, relations are terminated with a special token denoting their type (e.g. `@GDA@`). Two or more entities can be included before the special relation token to support n-ary extraction. Entities can be ordered if they serve specific roles as head or tail of a relation. For each document, multiple relations can be included in the target string. Entities may be nested or discontinuous in the input text. In Table 1, we provide examples of how this schema can be used to model various complexities, like coreferent entity mentions and n-ary relations.

3.2 Model

The model follows a canonical seq2seq setup. An encoder maps each token in the input to a contextual embedding. An autoregressive decoder generates an output, token-by-token, attending to the outputs of the encoder at each timestep (Figure 1). Formally, X is the *source* sequence of length S, which is some text we would like to extract relations from. Y is the corresponding *target* sequence of length T, a linearization of the relations contained in the source. We seek to model

$$p(Y|X) = \prod_{t=1}^{T} p(y_t|X, y_{<t}) \tag{1}$$

During training, we optimize over the model parameters θ the sequence cross-entropy loss

$$\ell(\theta) = -\sum_{t=1}^{T} \log p(y_t|X, y_{<t}; \theta) \tag{2}$$

maximizing the log-likelihood of the training data.[3] The main problems with this setup for RE are: 1) The model might "hallucinate" by generating entity mentions that do not appear in the source text. 2) It may generate a target string that does not follow the linearization schema, and therefore cannot be parsed. 3) The loss function is permutation-sensitive, enforcing an unnecessary decoding order.

To address 1) we use two modifications: a restricted target vocabulary (§3.2.1) and a copy mechanism (§3.2.2). To address 2) we experiment with several constraints applied during decoding (§3.2.3). Finally, to address 3) we sort relations according to their first appearance in the text (§3.2.4).

3.2.1 Restricted target vocabulary

To prevent the model from "hallucinating" (i.e. generating entity mentions that do not appear in the source text) the target vocabulary is restricted to the set of special tokens needed to model entities and relations (e.g. `;` and `@DRUG@`). All other tokens must be copied from the input using a copy mechanism (see §3.2.2). The embeddings of these special tokens are initialized randomly and learned jointly with the rest of the models parameters.

[3] See §4.3 for details about the encoder and decoder.

3.2.2 Copy mechanism

To enable copying of input tokens during decoding, we use a copying mechanism (Gu et al., 2016). The mechanism works by effectively extending the target vocabulary with the tokens in the source sequence X, allowing the model to "copy" these tokens into the output sequence, Y. Our use of the copy mechanism is similar to previous seq2seq-based approaches for RE (Zeng et al., 2018, 2020)

3.2.3 Constrained decoding

We experimented with several constraints applied to the decoder during inference to reduce the likelihood of generating syntactically invalid target strings (i.e. strings that do not follow the proposed linearization schema). These constraints are applied by setting the predicted probabilities of invalid tokens to a tiny value at each timestep. The full set of constraints is depicted in Appendix A. In practice, we found that a trained model rarely generates invalid target strings, so these constraints have little effect on final performance. We elected not to apply them in the rest of our experiments.

3.2.4 Sorting relations

The relations to extract from a given document are inherently unordered. However, the sequence cross-entropy loss (Equation 2) is permutation-sensitive with respect to the predicted tokens. During training, this enforces an unnecessary decoding order and may make the model prone to overfit frequent token combinations in the training set (Vinyals et al., 2016; Yang et al., 2019). To partially mitigate this, we sort relations within the target strings according to their order of first appearance in the source text, providing the model with a consistent decoding order. The order of a relation is determined by the sum of the end character offsets of each of its entities. When an entity has more than one mention, we take the end character offset of the mention that appears first in the text.

3.3 Entity hinting

Although the proposed model can extract entities and relations from unannotated text, it is interesting to consider the case where entities are known (e.g. as the predictions of an existing system) and provided to the model as input. To handle this case, we use a simple strategy that we refer to as "entity hinting". This involves prepending entities to the source text as they appear in the target string. Taking the example from §3.1, entity hints would be added as follows:

X: `estrogen receptor alpha ; ESR1 @GENE@ schizophrenia @DISEASE@ @HINTS@` Variants in the **estrogen receptor alpha** (**ESR1**) gene and its mRNA contribute to risk for **schizophrenia**.

where the special `@HINTS@` token demarcates the end of the entity hint. In our experiments, we use entity hinting when comparing to existing document-level RE methods that provide entities as input to the model (§5.1.1). In §5.2, we make use of entity hinting to compare a pipeline-based approach to an end-to-end approach.

4 Experimental setup

4.1 Datasets

We evaluate our approach on several document-level RE datasets. In Appendix B, we list relevant details about their annotations.

CDR (Li et al., 2016b) The BioCreative V CDR task corpus is manually annotated for chemicals, diseases and chemical-induced disease (CID) relations. It contains the titles and abstracts of 1500 PubMed articles and is split into equally sized train, validation and test sets. Given the relatively small size of the training set (500 examples), we follow Christopoulou et al. (2019) and others by first tuning the model on the validation set and then training on the combination of the train and validation sets before evaluating on the test set.

GDA (Wu et al., 2019) The gene-disease association corpus contains 30,192 titles and abstracts from PubMed articles that have been automatically labelled for genes, diseases and gene-disease associations via distant supervision. The test set is comprised of 1000 of these examples. Following Christopoulou et al. (2019) and others, we hold out a random 20% of the remaining abstracts as a validation set and use the rest for training.

DGM (Jia et al., 2019) The drug-gene-mutation corpus contains 4606 PubMed articles that have been automatically labelled for drugs, genes, mutations and ternary drug-gene-mutation relationships via distant supervision. The dataset is available in three variants of sentence-, paragraph-, and document-length text. We train and evaluate our model on the paragraph-length inputs. Since the test set does not contain relation annotations on the paragraph-level, we report results on the validation set. We hold out a random 15% of training examples to form a new validation set for tuning.

DocRED (Yao et al., 2019) DocRED includes over 5000 human-annotated documents from Wikipedia. There are 6 entity and 96 relation types, with ∼40% of relations crossing the sentence boundary, making this one of the most challenging document-level RE benchmarks to date. We use the same split as previous work on end-to-end document-level RE (Eberts and Ulges, 2021), which has 3,008 documents in the training set, 300 in the validation set and 700 in the test set[4].

4.2 Evaluation

We evaluate our model using the micro F1-score by extracting relations from the decoders output. Similar to prior work, we use a "strict" criteria. A predicted relation is considered correct if the relation type and its entities match a ground truth relation. An entity is considered correct if the entity type and its mentions match a ground truth entity. However, since the aim of document-level RE is to extract relations at the *entity*-level (as opposed to the *mention*-level), we also report performance using a relaxed criteria (denoted "relaxed" from here on), where predicted entities are considered correct if more than 50% of their mentions match a ground truth entity (see Appendix G).

Existing methods that evaluate on the CDR, GDA and DGM use the ground truth entity annotations as input. This makes it difficult to directly compare with our end-to-end approach, which takes only the raw text as input. To make the comparison fairer, we use entity hinting (§3.3) so that our model has access to the ground truth entity annotations. We also report the performance of our method in the end-to-end setting on these corpora to facilitate future comparison. To compare to existing end-to-end approaches, we use DocRED.

4.3 Implementation, training and hyperparameters

Implementation We implemented our model in PyTorch (Paszke et al., 2017) using AllenNLP (Gardner et al., 2018). As encoder, we use a pretrained transformer, implemented in the Transformers library (Wolf et al., 2020), which is fine-tuned during training. When training and evaluating on biomedical corpora, we use PubMedBERT

[4] `https://github.com/lavis-nlp/jerex`

Table 2: Comparison to existing pipeline-based methods. Performance reported as micro-precision, recall and F1-scores (%) on the CDR and GDA test sets. Results below the horizontal line are not comparable to existing methods. Bold: best scores.

	CDR			GDA		
Method	P	R	F1	P	R	F1
Christopoulou et al. (2019)	62.1	65.2	63.6	–	–	81.5
Nan et al. (2020)	–	–	64.8	–	–	82.2
Lai and Lu (2021)	64.9	67.1	66.0	–	–	82.8
Minh Tran et al. (2020)	–	–	66.1	–	–	82.8
Xu et al. (2021)	–	–	68.7	–	–	83.7
Zhou et al. (2021)	–	–	**69.4**	–	–	83.9
seq2rel (entity hinting)	65.3	66.2	65.8	83.6	85.0	**84.3**
seq2rel (entity hinting, relaxed)	64.6	65.3	64.9	83.7	85.1	84.4
seq2rel (end-to-end)	39.8	35.6	37.6	54.8	55.2	55.0
seq2rel (end-to-end, relaxed)	52.5	46.9	49.6	70.0	70.5	70.2

(Gu et al., 2020), and $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019) otherwise. As decoder, we use a single-layer LSTM with randomly initialized weights. We use multi-head attention (Vaswani et al., 2017) as the encoder-decoder attention mechanism.

Training All parameters of the model are trained jointly using the AdamW optimizer (Loshchilov and Hutter, 2019). The learning rate is linearly increased for the first 10% of training steps and linearly decayed to zero afterward. Gradients are scaled to a vector norm of 1.0 before backpropagating. The hidden state of the decoder is initialized with the `[CLS]` token representation output by the encoder. As is common, we use teacher forcing, feeding previous ground truth inputs to the decoder when predicting the next token in the sequence. During inference, we generate the output using beam search decoding (Graves, 2012). Beams are ranked by mean token log probability. All models were trained and evaluated on a single NVIDIA Tesla V100. See Appendix C for hyperparameters.

5 Results

5.1 Comparison to existing methods

In the following sections, we compare our model to existing document-level RE methods on several benchmark corpora. We include existing pipeline-based methods (§5.1.1), n-ary methods, (§5.1.2), and end-to-end methods (§5.1.3). Details about these methods are provided in Appendix D.

5.1.1 Existing pipeline-based methods

In Table 2 we list our results on the GDA corpus. Although our method is designed for end-to-end RE, we find that it outperforms existing pipeline-

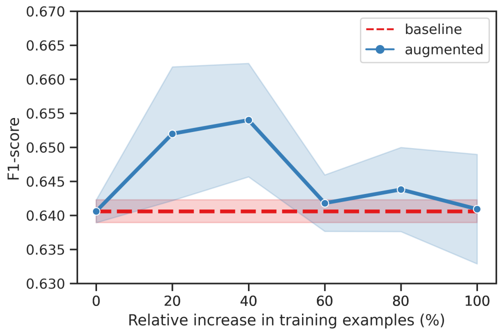

Figure 2: Data augmentation by concatenation. Performance over five runs reported as micro F1-score on the CDR test set with entity-hinting. At the beginning of each epoch, we randomly concatenate pairs of existing training examples and add them to the original train set. 100% corresponds to doubling train set size. Standard deviation is displayed as a band.

based methods when using entity hinting. We also report end-to-end performance, which is not comparable to existing pipeline-based methods but will facilitate future comparisons. The large performance improvement when using entity hinting (+29%) confirms that the model benefits from the hints. The fact that relaxed entity matching makes a large difference in the end-to-end setting (+15%), suggests that a significant portion of the model's mistakes occur during coreference resolution.

Unlike GDA, our method underperforms existing methods on CDR (Table 2). Given that GDA is 46X larger, we speculated that our method might be underperforming in the low-data regime. To determine if this is a contributing factor, we artificially reduce the size of the GDA and CDR training sets and plot the performance as a curve (Appendix E). On both corpora, performance increases monotonically with dataset size. There is no obvious plateau on CDR even when using all 500 training examples. Performance only begins to plateau on GDA after training on $\sim$14,000 examples. To improve performance in the low-data regime, we adapted a data augmentation technique from neural machine translation (Kondo et al., 2021; Nguyen et al., 2021). This technique creates additional training examples simply by concatenating pairs of existing training examples (see Appendix F). We re-train on the CDR corpus, increasing the number of training examples via augmentation, and plot the performance as a curve (Figure 2). We find that a small amount of augmentation can boost in performance by as much as 1%, but too much can hurt. Together, these results suggest that our seq2seq based approach can outperform existing pipeline-based methods when there are sufficient training examples but underperforms relative to existing methods in the low-data regime. However, this can be partially mitigated using a simple data augmentation technique.

Table 3: Comparison to existing n-ary methods. Performance reported as micro-precision, recall and F1-scores (%) on the DGM validation set. Results below the horizontal line are not comparable to existing methods. Bold: best scores. † Jia et al. 2019 do not report results on the validation set, so we re-run their paragraph-level model.

Method	P	R	F1
Jia et al. (2019) †	68.4	70.6	69.5
seq2rel (entity hinting)	**84.5**	**77.3**	**80.7**
seq2rel (entity hinting, relaxed)	84.8	77.5	81.0
seq2rel (end-to-end)	63.4	56.4	59.7
seq2rel (end-to-end, relaxed)	72.5	64.4	68.2

5.1.2 n-ary relation extraction

In Table 3 we compare against existing n-ary document-level RE methods on the DGM corpus. With entity hinting, our method outperforms existing methods. This result suggests that our linearization schema effectively models n-ary relations without requiring any changes to the model architecture or training procedure.

5.1.3 End-to-end methods

In Table 4 we compare against existing end-to-end approaches on DocRED. To the best of our knowledge, Eberts and Ulges (2021) is the only method to evaluate an end-to-end approach on DocRED. To make the comparison fair, we use the same pretrained encoder ($\text{BERT}_{\text{BASE}}$). We find that our model underperforms JEREX, mainly due to recall. We speculate that this is due to the large number of relations per document, which leads to longer target strings and, therefore, more decoding steps. The median length of the target strings in DocRED, using our linearization, is 205, whereas the next largest is 21 in GDA. We speculate that improving the decoder's ability to process long sequences (e.g. by switching the LSTM for a Transformer) or modifying the linearization schema to produce shorter target strings, may improve recall and close the gap with existing methods.

5.2 Pipeline vs. End-to-end

In §5.1.1 and §5.1.2, we provide gold-standard entity annotations from each corpus as input to our

Table 4: Comparison to existing end-to-end methods. Performance reported as micro-precision, recall and F1-scores (%) on the DocRED test set. Results below the horizontal line are not comparable to existing methods. Bold: best scores.

Method	P	R	F1
JEREX (Eberts and Ulges, 2021)	42.8	**38.2**	**40.4**
seq2rel (end-to-end)	**43.8**	32.0	37.0
seq2rel (end-to-end, relaxed)	53.6	39.2	45.3

Table 5: Comparison of pipeline and end-to-end approach. Gold hints use gold-standard entity annotations to insert entity hints in the source text. Silver hints use the entity annotations provided by PubTator. Pipeline is identical to silver entity hints, except that we filter out entity mentions predicted by our model that PubTator does not predict. The end-to-end model only has access to the unannotated source text as input. Performance reported as micro-precision, recall and F1-scores (%) on the CDR test set, with strict and relaxed entity matching criteria. Bold: best scores.

	Strict			Relaxed		
	P	R	F1	P	R	F1
Gold hints	64.4	65.1	64.7	64.6	65.3	64.9
Silver hints	41.6	35.9	**38.5**	53.6	46.3	49.7
Pipeline	**42.4**	29.9	35.0	**55.5**	39.0	45.7
End-to-end	40.8	**36.2**	38.4	53.0	**47.0**	**49.9**

model (via entity hinting, referred to as "gold" hints from here on), allowing us to compare to existing methods that also provide these annotations as input. However, gold-standard entity annotations are (almost) never available in real-world settings, such as large-scale extraction on PubMed. In this setting, there are two strategies: pipeline-based approaches, where independent systems perform entity and relation extraction, and end-to-end approaches, where a single model performs both tasks. To compare these approaches under our model, we perform evaluations where an existing entity extraction system is used to determine entity hints ("silver" hints) and when no entity hints are provided (end-to-end).[5] However, this alone does not create a true pipeline, as our model can recover from false negatives in the entity extraction step. To mimic error propagation in the pipeline setting, we filter any entity mention predicted by our model that does *not* appear in the hints. In Table 6, we present the results of all four settings (gold and silver entity hints, pipeline and end-to-end) on CDR.

[5]Specifically, we use PubTator (Wei et al., 2013). PubTator provides up-to-date entity annotations for PubMed using state-of-the-art machine learning systems.

Table 6: Ablation study results. Performance reported as micro-precision, recall and F1-scores (%) on the CDR validation set, with and without entity hinting. Δ: difference to the full models F1-score. Bold: best scores.

	Entity hinting				End-to-end			
	P	R	F1	Δ	P	R	F1	Δ
Full model	**64.9**	**63.6**	**64.2**	–	38.6	**33.7**	36.0	–
- pretraining	39.4	26.5	31.7	-32.5	11.0	7.9	9.2	-26.8
- fine-tuning	45.9	38.3	41.7	-22.5	25.6	20.8	22.9	-13.1
- sorting relations	62.7	56.2	59.3	-5.0	37.5	30.0	33.3	-2.7
- vocab restriction	62.8	59.9	61.3	-2.9	**40.1**	33.3	**36.4**	+0.4

First, we find that using gold entity hints significantly outperforms all other settings. This is expected, as the gold-standard entity annotations are high-quality labels produced by domain experts. Using silver hints significantly drops performance, likely due to a combination of false positive and false negatives from the entity extraction step. In the pipeline setting, where there is no recovery from false negatives in the entity extraction step, performance falls by over 3%. Under our model, the end-to-end setting significantly outperforms the pipeline setting (due to a large boost in recall) and performs comparably to using silver entity hints. Together, our results suggest that performance reported using gold-standard entity annotations can be overly optimistic and corroborate previous work demonstrating the benefits of jointly learning entity and relation extraction (Miwa and Sasaki, 2014; Miwa and Bansal, 2016; Gupta et al., 2016; Li et al., 2016a, 2017; Nguyen and Verspoor, 2019a; Yu et al., 2020).

5.3 Ablation

In Table 6, we present the results of an ablation study on the CDR corpus. We perform the analysis twice, once with entity hinting (see §3.3) and once without. Unsurprisingly, we find that fine-tuning a pretrained encoder has a large impact on performance. Training the same encoder from scratch reduces performance by 26.8-32.5% (depending on whether entity hints are used or not). Using the pretrained weights without fine-tuning drops performance by 13.1-22.5%. Deliberately ordering the relations within each target string has a large positive impact, boosting performance by 2.7%-5.0%. This is likely because the sequence cross-entropy is permutation-sensitive; sorting relations removes ambiguity as to the order they should be decoded (see §3.2.4). Lastly, we find that restricting the target vocabulary (see §3.2.1) improves performance when entity hints are used but slightly reduces performance in the end-to-end setting. The motivation for restricting the vocabulary was to prevent hallucination, as it forces the model to copy entity mentions from the source text. The results suggest that, in the end-to-end setting, hallucination is less of a problem than initially assumed.

6 Discussion

6.1 Related work

Seq2seq learning for RE has been explored in prior work. CopyRE (Zeng et al., 2018) uses an encoder-decoder architecture with a copy mechanism, similar to our approach, but is restricted to intra-sentence relations. Additionally, because CopyRE's decoding proceeds for exactly three timesteps per relation, the model is limited to generating binary relations between single token entities. The ability to decode multi-token entities was addressed in follow-up work, CopyMTL (Zeng et al., 2020). A similar approach was published concurrently but was again limited to intra-sentence binary relations (Nayak and Ng, 2020). None of these approaches deal with the complexities of document-level RE, where many relations cross the sentence boundary, and coreference resolution is critical.

More generally, our paper is related to a recently proposed "text-to-text" framework (Raffel et al., 2020). In this framework, a task is formulated so that the inputs and outputs are both text strings, enabling the use of the same model, loss function and even hyperparameters across many seq2seq, classification and regression tasks. This framework has recently been applied to biomedical literature to perform named entity recognition, relation extraction (binary, intra-sentence), natural language inference, and question answering (Phan et al., 2021). Our work can be seen as an attempt to formulate the task of document-level RE within this framework.

6.2 Limitations and future work

Permutation-sensitive loss Our approach adopts the sequence cross-entropy loss (Equation 2), which is sensitive to the order of predicted tokens, enforcing an unnecessary decoding order on the inherently unordered relations. To partially mitigate this problem, we order relations within the target string according to order of first appearance in the source text, providing the model with a consistent decoding order that can be learned (see §3.2.4, §5.3). Previous work has addressed this issue with various strategies, including reinforcement learning (Zeng et al., 2019), unordered-multi-tree decoders (Zhang et al., 2020), and non-autoregressive decoders (Sui et al., 2020). However, these works are limited to binary intra-sentence relation extraction, and their suitability for document-level RE has not been explored. An exciting future direction would be to modify our approach such that the arbitrary order of relations is not enforced during training.

Input length restriction Due to the pretrained encoder's input size limit (512 tokens), our experiments are conducted on paragraph-length text. Our model could be extended to full documents by swapping its encoder with any of the recently proposed "efficient transformers" (Tay et al., 2021). Future work could evaluate such a model's ability to extract relations from full scientific papers.

Pretraining the decoder In our model, the encoder is pretrained, while the decoder is trained from scratch. Several recent works, such as T5 (Raffel et al., 2020) and BART (Lewis et al., 2020), have proposed pretraining strategies for entire encoder-decoder architectures, which can be fine-tuned on downstream tasks. An interesting future direction would be to fine-tune such a model on document-level RE using our linearization schema.

7 Conclusion

In this paper, we extend seq2seq methods for relation extraction to document-level RE. We propose a novel linearization schema for entities and relations that is capable of modelling coreferent mentions and inter-sentence relations (prerequisites for document-level RE) and n-ary relations. We also propose a simple strategy for providing the model with entity annotations as input that we call entity hinting. We include comparisons to existing pipeline-based and end-to-end methods on several benchmark corpora, in some cases exceeding their performance. In future work, we hope to develop strategies to improve performance in the low-data regime, and cases where there are a large number of relations per document.

References

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

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. 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 4925–4936, Hong Kong, China. Association for Computational Linguistics.

Kenneth Church. 2017. Emerging trends: Inflation. *Natural Language Engineering*, 23(5):807–812.

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.

Markus Eberts and Adrian Ulges. 2021. An end-to-end model for entity-level relation extraction using multi-instance learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3650–3660, Online. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *ArXiv preprint*, abs/1211.3711.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing. *ArXiv preprint*, abs/2007.15779.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. SciREX: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7506–7516, Online. Association for Computational Linguistics.

Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level n-ary relation extraction with multi-scale representation learning. 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 3693–3704, Minneapolis, Minnesota. Association for Computational Linguistics.

Jin-Dong Kim, T. Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19 Suppl 1:i180–2.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Seiichiro Kondo, Kengo Hotate, Tosho Hirasawa, Masahiro Kaneko, and Mamoru Komachi. 2021. Sentence concatenation approach to data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 143–149, Online. Association for Computational Linguistics.

Po-Ting Lai and Zhiyong Lu. 2021. Bert-gt: Cross-sentence n-ary relation extraction with bert and graph transformer. *Bioinformatics*.

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.

Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. 2017. A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, 18(1):198.

Fei Li, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016a. Joint models for extracting adverse drug events from biomedical text. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA,* *9-15 July 2016*, pages 2838–2844. IJCAI/AAAI Press.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016b. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *ArXiv preprint*, abs/d.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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

Hieu Minh Tran, Minh Trung Nguyen, and Thien Huu Nguyen. 2020. The dots have their values: Exploiting the node-edge connections in graph-based neural models for document-level relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4561–4567, Online. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557, Online. Association for Computational Linguistics.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8528–8535. AAAI Press.

Dat Quoc Nguyen and Karin Verspoor. 2019a. End-to-end neural relation extraction using deep biaffine attention. In *Advances in Information Retrieval*, pages 729–738, Cham. Springer International Publishing.

Dat Quoc Nguyen and Karin Verspoor. 2019b. End-to-end neural relation extraction using deep biaffine attention. In *European Conference on Information Retrieval*, pages 729–738. Springer.

Toan Q. Nguyen, Kenton Murray, and David Chiang. 2021. Data augmentation by concatenation for low-resource translation: A mystery and a solution. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 287–293, Bangkok, Thailand (online). Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph LSTMs. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Long N Phan, James T Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. Scifive: a text-to-text transformer model for biomedical literature. *ArXiv preprint*, abs/2106.03598.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph-state LSTM. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2226–2235, Brussels, Belgium. Association for Computational Linguistics.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, Xiangrong Zeng, and Shengping Liu. 2020. Joint entity and relation extraction with set prediction networks. *ArXiv preprint*, abs/2011.01675.

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.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021.

Long range arena : A benchmark for efficient transformers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order matters: Sequence to sequence for sets. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Yucheng Wang, Bowen Yu, Hongsong Zhu, Tingwen Liu, Nan Yu, and Limin Sun. 2021. Discontinuous named entity recognition as maximal clique discovery. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 764–774, Online. Association for Computational Linguistics.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2013. Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic acids research*, 41(W1):W518–W522.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ye Wu, Ruibang Luo, Henry C. M. Leung, Hing-Fung Ting, and Tak-Wah Lam. 2019. Renet: A deep learning approach for extracting gene-disease associations from literature. In *Research in Computational Molecular Biology*, pages 272–284, Cham. Springer International Publishing.

Benfeng Xu, Quan Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In *AAAI*.

Pengcheng Yang, Fuli Luo, Shuming Ma, Junyang Lin, and Xu Sun. 2019. A deep reinforced sequence-to-set model for multi-label classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5252–5258, Florence, Italy. Association for Computational Linguistics.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Tingwen Liu, Yubin Wang, Bin Wang, and Sujian Li. 2020. Joint extraction of entities and relations based on a novel decomposition strategy. In *ECAI 2020*, pages 2282–2289. IOS Press.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9507–9514. AAAI Press.

Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. 2019. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. 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 367–377, Hong Kong, China. Association for Computational Linguistics.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

Ranran Haoran Zhang, Qianying Liu, Aysa Xuemo Fan, Heng Ji, Daojian Zeng, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020. Minimize exposure bias of Seq2Seq models in joint entity and relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 236–246, Online. Association for Computational Linguistics.

Deyu Zhou, Dayou Zhong, and Yulan He. 2014. Biomedical relation extraction: from binary to complex. *Computational and mathematical methods in medicine*, 2014.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14612–14620.

A Constrained decoding

In Figure 3, we illustrate the rules used to constrain decoding. At each timestep t, given the prediction of the previous timestep $t-1$, the predicted class probabilities of tokens that would generate a syntactically invalid target string are set to a tiny value. In practice, we found that a trained model rarely generates invalid target strings, so these constraints have little effect on final performance. Therefore, we elected not to apply them in our experiments.

B Details about dataset annotations

In Table 7, we list which modelling complexities (e.g. nested and discontinuous mentions) are contained within each corpora used in our evaluations.

C Hyperparameters

In Table 8, we list the hyperparameter values used during evaluation on each corpus.

D Baselines

This section contains detailed descriptions of all methods we compare to in the main paper.

D.1 Pipeline-based methods

These methods are pipeline-based, assuming the entities are provided as input. Many of them construct a graph with dependency parsing, heuristics, or structured attention, and then performance inference with graph neural networks (Kipf and Welling, 2017).

- Christopoulou et al. (2019) propose EoG, an edge-orientated graph neural model. The nodes of the graph are constructed from mentions, entities, and sentences. Edges between nodes are initially constructed using heuristics. An iterative algorithm is then used to generate edges between nodes in the graph. Finally, a classification layer takes the representation of entity-to-entity edges as input to determine whether those entities express a relation or not. We compare to EoG in the pipeline-based setting on the CDR and GDA corpora.

- Nan et al. (2020) propose LSR (Latent Structure Refinement). A "node constructor" encodes each sentence of an input document and outputs contextual representations. Representations that correspond to mentions and tokens on the shortest dependency path in a sentence are extracted as nodes. A "dynamic reasoner" is then applied to induce a document-level graph based on the extracted nodes. The classifier uses the final representations of nodes for relation classification. We compare to LSR in the pipeline-based setting on the CDR and GDA corpora.

- Lai and Lu (2021) propose BERT-GT, which combines BERT with a graph transformer. Both BERT and the graph transformer accept the document text as input, but the graph transformer requires the neighbouring positions for each token, and the self-attention mechanism is replaced with a neighbour–attention mechanism. The hidden states of the two transformers are aggregated before classification. We compare to BERT-GT in the pipeline-based setting on the CDR and GDA corpora.

- Minh Tran et al. (2020) propose EoGANE (EoG model Augmented with Node Representations), which extends the edge-orientated model proposed by Christopoulou et al. (2019) to include explicit node representations which are used during relation classification. We compare to EoGANE in the pipeline-based setting on the CDR and GDA corpora.

- SSAN (Xu et al., 2021) propose SSAN (Structured Self-Attention Network) which inherits the architecture of the transformer encoder (Vaswani et al., 2017), but adds a novel structured self-attention mechanism to model the coreference and co-occurrence dependencies between an entities mentions. We compare to SSAN in the pipeline-based setting on the CDR and GDA corpora.

- Zhou et al. (2021) propose ALTOP (Adaptive Thresholding and Localized cOntext Pooling) which extends extends BERT with two modifications. Adaptive thresholding, which learns an optimal threshold to apply to the relation classifier. Localized context pooling, which uses the pretrained self-attention layers of BERT to create an entity embedding from its mentions and their context. We compare to ALTOP in the pipeline-based setting on the CDR and GDA corpora.

D.2 n-ary relation extraction

These methods are explicitly designed for the extraction of n-ary relations, where $n > 2$.

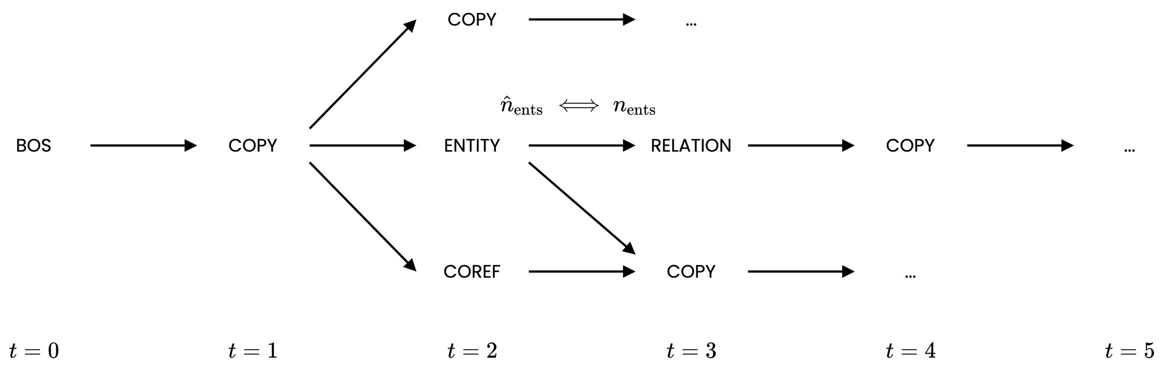

Figure 3: A diagram depicting the syntactically valid predictions during decoding at each timestep t. The class log probabilities of all other possible predictions are set to a tiny value to prevent the model from producing a syntactically invalid target string. `BOS` is the special beginning-of-sequence token, `COPY` denotes any token copied from the source text, and `COREF` is the special token used to separate coreferent mentions (i.e. `;`). `ENTITY` is any special entity token (e.g. `@GENE@`) and `RELATION` any special relation token (e.g. `@GDA@` for gene-disease association). $\hat{n}_{\text{ents}}$ denotes the number of entities predicted by the current timestep and n_{ents} the expected arity of the relation. The special end-of-sequence token, `EOS` (not shown) is always considered syntactically valid, and therefore its class log probability is never modified.

Table 7: Evaluation datasets used in this paper with details about their annotations.

Corpus	Nested Mentions?	Discontinuous Mentions?	Coreferent mentions?	Inter-sentence relations?	n-ary relations?
CDR (Li et al., 2016b)	✓	✓	✓	✓	✗
GDA (Wu et al., 2019)	✓	✗	✓	✓	✗
DGM (Jia et al., 2019)	✗	✗	✓	✓	✓
DocRED (Yao et al., 2019)	✗	✗	✓	✓	✗

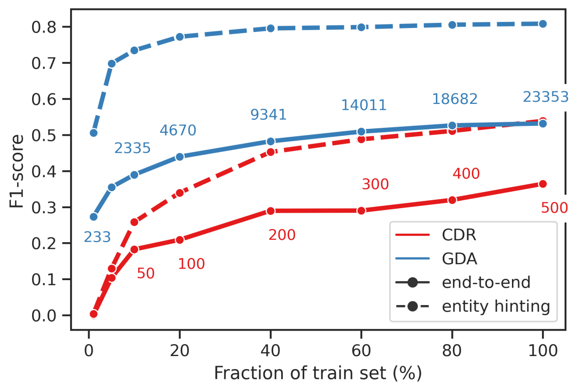

Figure 4: Effect of training set size on performance. Performance reported as micro F1-scores obtained on the CDR and GDA validation sets, with and without entity hinting. The absolute number of training examples are displayed for each corpus. Some labels are excluded to reduce clutter.

- Jia et al. (2019) propose a multiscale neural architecture, which combines representations learned over text spans of varying scales and for various sub-relations. We compare to Jia et al. (2019) in the pipeline-based setting on the n-ary DGM corpus.

D.3 End-to-end methods

These methods are capable of performing the subtasks of document-level RE in an end-to-end fashion with only the document text as input.

- Eberts and Ulges (2021) propose JEREX, which extends BERT with four task-specific components that use BERTs outputs to perform entity mention localization, coreference resolution, entity classification, and relation classification. They present two versions of their relation classifier, denoted "global relation classifier" (GRC) and "multi-instance relation classifier" (MRC). We compare to JEREX-MRC in the end-to-end setting on the DocRED corpus.

E Effect of training set size

In Figure 4 we artificially reduce the size of the training set and plot the resulting performance on the validation set as a curve. We perform this analysis for the CDR and GDA corpus, with and without entity hinting.

Table 8: Hyperparameter values used for each corpus. Hyperparameters values when using entity hinting, if they differ from the values used without entity hinting, are shown in parentheses.

Hyperparameter	CDR	GDA	DGM	DocRED
Batch size	4 (1)	8	6	4
Epochs	50 (30)	20 (15)	20	40
Encoder LR	2e-5	5e-5 (2e-5)	2e-5	2e-5
Decoder LR	3e-4 (5e-4)	5e-4 (2e-4)	2e-4	1e-4
Target embedding size	256	256	256	256
No. heads in encoder-decoder multi-head attention	6	6	6	6
Beam size	2 (6)	2	2	8
Length penalty	1.5 (10.0)	1.0	1.0	5.0
Max decoding steps	128	96	72	400

```python
import random

# Load (original) train data
train_data = load_train_data()
# Shuffle the data
random.shuffle(train_data)

n = len(train_data)

# Accumulate tuples of concatenated
# source (X) and target (Y) strings
aug_data = []
for i, j in zip(range(n - 1), range(1, n)):
    x_i, y_i = train_data[i]
    x_j, y_j = train_data[j]
    aug_data.append((x_i + x_j, y_i + y_j))

# Add the augmented data to the original data
train_data = train_data + aug_data
```

Listing 1: Pseudocode for the augmentation by concatenation technique in a Python-like style.

F Augmentation by concatenation

To improve performance in the low-data regime, we adopt a simple data augmentation technique from low-resource machine translation (Kondo et al., 2021; Nguyen et al., 2021). This technique creates additional training examples by concatenating pairs of existing examples together. In 1, we provide Python pseudocode depicting the method. In practice, we randomly sample some fraction of the original dataset (e.g. 25%) at the beginning of each epoch to create the augmented data from. The examples created via augmentation are added to the original training set. We found that creating new augmented data in each epoch outperformed creating the augmented data once before training began.

G Relaxed entity matching

The aim of document-level RE is to extract relations at the *entity*-level. However, it is common to evaluate these methods with a "strict" matching criteria, where a predicted entity $\mathcal{P}$ is considered correct if and only if all its *mentions* exactly match a corresponding gold entities mentions, i.e. $\mathcal{P} = \mathcal{G}$. This penalizes model predictions that miss even a single coreferent mention, but are otherwise correct. A relaxed criteria, proposed in prior work (Jain et al., 2020) considers $\mathcal{P}$ to match $\mathcal{G}$ if more than 50% of $\mathcal{P}$'s mentions belong to $\mathcal{G}$, that is

$$\frac{|\mathcal{P} \cap \mathcal{G}|}{|\mathcal{P}|} > 0.5$$

In the main paper, alongside the strict criteria, we report performance using this relaxed entity matching strategy (denoted "relaxed").