A Dataset for Hyper-Relational Extraction and a Cube-Filling Approach

Yew Ken Chia[*1] **Lidong Bing**[†1] **Sharifah Mahani Aljunied**[1]
Luo Si[1] **Soujanya Poria**

[1]DAMO Academy, Alibaba Group Singapore University of Technology and Design
{yewken.chia, l.bing, mahani.aljunied, luo.si}@alibaba-inc.com
{yewken_chia, sporia}@sutd.edu.sg

Abstract

Relation extraction has the potential for large-scale knowledge graph construction, but current methods do not consider the qualifier attributes for each relation triplet, such as time, quantity or location. The qualifiers form hyper-relational facts which better capture the rich and complex knowledge graph structure. For example, the relation triplet (Leonard Parker, Educated At, Harvard University) can be factually enriched by including the qualifier (End Time, 1967). Hence, we propose the task of hyper-relational extraction to extract more specific and complete facts from text. To support the task, we construct HyperRED, a large-scale and general-purpose dataset. Existing models cannot perform hyper-relational extraction as it requires a model to consider the interaction between three entities. Hence, we propose CubeRE, a cube-filling model inspired by table-filling approaches and explicitly considers the interaction between relation triplets and qualifiers. To improve model scalability and reduce negative class imbalance, we further propose a cube-pruning method. Our experiments show that CubeRE outperforms strong baselines and reveal possible directions for future research. Our code and data are available at github.com/declare-lab/HyperRED.

1 Introduction

Knowledge acquisition is an open challenge in artificial intelligence research (Lenat, 1995). The standard form of representing the acquired knowledge is a knowledge graph (Hovy et al., 2013), which has broad applications such as question answering (Yih and Ma, 2016; Chia et al., 2020) and search engines (Xiong et al., 2017). Relation extraction (RE) is a task that has the potential for large-scale and automated knowledge graph construction by extracting facts from natural language text. Most

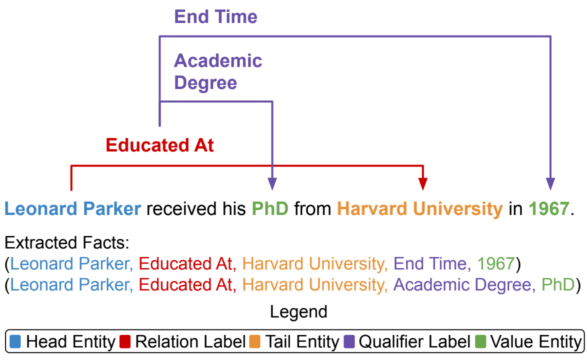

Figure 1: A sample from our HyperRED dataset for the proposed task of hyper-relational extraction.

relation extraction methods focus on binary relations (Bach and Badaskar, 2007) which consider the relationship between two entities, forming a relation triplet consisting of the head entity, relation and tail entity respectively.

However, knowledge graphs commonly contain hyper-relational facts (Guan et al., 2019) which have qualifier attributes for each relational triplet, such as time, quantity, or location. For instance, Wen et al. (2016) found that the Freebase knowledge graph contains hyper-relational facts for 30% of entities. Hence, extracting relation triplets may be an oversimplification of the rich and complex knowledge graph structure. As shown in Figure 1, a relation triplet can be attributed to one or more qualifiers, where a qualifier is composed of a qualifier label and value entity. For example, the relation triplet (`Leonard Parker`, *Educated At*, `Harvard University`) can be factually enriched by specifying the qualifier of (*End Time*, `1967`), forming the hyper-relational fact (`Leonard Parker`, *Educated At*, `Harvard University`, *End Time*, `1967`).

Hyper-relational facts generally cannot be simplified into the relation triplet format as the qualifiers are attributed to the triplet as a whole and not targeted at a specific entity in the triplet. Furthermore, attempting to decompose the hyper-relational structure to an n-ary format would lose

[*]Yew Ken is a student under the Joint PhD Program between Alibaba and SUTD.
[†]Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 10114–10133
December 7-11, 2022

the original triplet information and be incompatible with the knowledge graph schema (Rosso et al., 2020). On the other hand, hyper-relational facts have practical benefits such as improved fact verification (Thorne et al., 2018) and representation learning for knowledge graphs (Galkin et al., 2020). Thus, it is necessary to extract relation triplets together with qualifiers to form hyper-relational facts.

In this work, we propose the task of hyper-relational extraction to jointly extract relation triplets with qualifiers from natural language sentences. To support the task, we contribute a general-purpose and large-scale hyper-relational extraction dataset (HyperRED) which is constructed through distant supervision (Mintz et al., 2009) and partially refined through human annotation. Our dataset differs from previous datasets in two distinct ways: (1) Compared to existing datasets for binary relation extraction (Zhang et al., 2017; Han et al., 2018), HyperRED enables richer information extraction as it contains qualifiers for each relation triplet in the sentence. (2) While datasets for n-ary relation extraction (Jia et al., 2019) are restricted to the biomedical domain, HyperRED covers multiple domains and has a hyper-relational fact structure that is compatible with the knowledge graph schema.

Unfortunately, to the best of our knowledge, there are no existing models for hyper-relational extraction. Currently, a popular end-to-end method for binary relation extraction is to cast it as a table-filling problem (Miwa and Sasaki, 2014). Generally, a two-dimensional table is used to represent the interaction between any two individual words in a sentence. However, hyper-relational extraction requires the model to consider the interactions between two entities in the relation triplet, as well as the value entity for the qualifier. Thus, we extend the table-filling approach to a third dimension, casting it as a cube-filling problem. On the other hand, a naive cube-filling approach faces two issues: (1) Computing the full cube representation is computationally expensive and does not scale well to longer sequence lengths. (2) The full cube will be sparsely labeled with a vast majority of entries as negative samples, causing the model to be biased in learning (Li et al., 2020) and hence underperform.

To tackle these two issues, we propose a simple yet effective cube-pruning technique that filters the cube entries based on words that are more likely to constitute valid entities. Our experiments show that cube-pruning significantly improves the computational efficiency and simultaneously improves the extraction performance by reducing the negative class imbalance. In addition to our cube-filling model which we refer to as CubeRE, we also introduce two strong baseline models which include a two-stage pipeline and a generative sequence-to-sequence (Sutskever et al., 2014) model.

In summary, our main contributions include: (1) We propose the task of hyper-relational extraction to extract richer and more complete facts by jointly extracting each relation triplet with the corresponding qualifiers; (2) To support the task, we provide a large-scale and general-purpose dataset known as HyperRED. (3) As there is no existing model for hyper-relational extraction, we propose a cube-filling model known as CubeRE, which consistently outperforms baseline extraction methods.

2 HyperRED: A Hyper-Relational Extraction Dataset

Our goal is to construct a large-scale and general-purpose dataset for extracting hyper-relational facts from natural language text. However, it is seldom practical to assume to have an ample amount of high-quality labeled samples in real applications, especially for complex tasks such as information extraction. Hence, we propose a weakly supervised (Craven and Kumlien, 1999) data setting which enables us to collect a larger and more diverse training set than would be otherwise possible. To minimize the effect of noisy samples in evaluation, we then perform human annotation for a portion of the collected data and allocate it as the held-out set. In the following sections, we first introduce the process of collecting the distantly supervised data, followed by the human-annotated data portion.

2.1 Distantly Supervised Data Collection

To collect a large and diverse dataset of sentences with hyper-relational facts, we employ distant supervision which falls under the weakly supervised setting. Distant supervision automatically collects a dataset of relational facts by aligning a text corpus with facts from an existing knowledge graph. Similar to Elsahar et al. (2018), we first extract and link entities from the corpus to an existing knowledge graph, and resolve any coreference cases to the previously linked entities. To align hyper-relational facts from the knowledge graph to the text corpus, we detect if the entities that comprise each fact are also present in each sentence. Each sentence with

Type	Proportion	Example Sentence	Hyper-Relational Facts
Time	48%	Tennyson was an ASCAP member from 1950.	(Tennyson, member of, ASCAP, start time, 1950)
Quantity	19%	Szewczyk played 37 times for Poland, scoring 3 goals.	(Szewczyk, member of sports team, Poland, number of matches played, 37) (Szewczyk, member of sports team, Poland, number of points, 3)
Role	12%	John Sculley is a former Apple CEO.	(John Sculley, employer, Apple, position held, CEO)
Part-Whole	11%	The Ohio Senate is the upper house of the Ohio General Assembly, the Ohio state legislature.	(Ohio, legislative body, Ohio General Assembly, has part, Ohio Senate)
Location	9%	Donner was elected at the 1931 election as Conservative MP for Islington West.	(Donner, candidacy in election, 1931 election, electoral district, Islington West)

Table 1: General typology and distribution of frequent qualifier labels for the HyperRED dataset, shown with example sentences and the corresponding hyper-relational facts.

aligned facts is collected as part of the distantly supervised dataset. To ensure that the large-scale text corpus can be well-aligned with the knowledge graph, we perform distant supervision between English Wikipedia and Wikidata (Erxleben et al., 2014), which is the central knowledge graph for Wikipedia. Following Elsahar et al. (2018), we use the introduction sections of Wikipedia articles as the text corpus as they generally contain the most important information.

Entity Extraction and Linking The distant supervision process relies on matching entities in a sentence with facts from the knowledge graph. To detect and identify the named entities in the articles, we use the DBpedia Spotlight (Mendes et al., 2011) entity linker. For the extraction of temporal and numerical entities, we use the spaCy [1] tool.

Coreference Resolution As Wikipedia articles often use pronouns to refer to entities across sentences, it is necessary to resolve such references. We employ the Stanford CoreNLP tool (Manning et al., 2014) for this task.

Hyper-Relational Alignment To extend the distant supervision paradigm to hyper-relational facts, we jointly match based on the entities that comprise each hyper-relational fact. Formally, let $f = (e_{head}, r, e_{tail}, q, e_{value})$ be a possible hyper-relation fact consisting of the head entity, relation, tail entity, qualifier label and value entity, respectively. Given a corpus of text articles, each article contains a set of sentences $\{s_i, ..., s_n\}$, where each sentence s_i has E_i entities that are linked to the knowledge graph. For each hyper-relational fact f in the knowledge graph, it is aligned to the sentence s_i if the head entity e_{head}, tail entity e_{tail} and value entity e_{value} are all linked in the sentence. Hence, we obtain a set of aligned facts for each sentence: $\{(s_i, f) \mid e_{head} \in E_i, e_{tail} \in E_i, e_{value} \in E_i\}$. Following Riedel et al. (2010), we remove any sentence that does not contain aligned facts.

[1] https://spacy.io

2.2 Human-Annotated Data Collection

Although distant supervision can align a large amount of hyper-relational facts, the process can introduce noise in the dataset due to possible spurious alignments and incompleteness of the knowledge graph (Nickel et al., 2016). However, it is not feasible to completely eliminate such noise from the dataset due to the annotation time and budget constraints. Hence, we select a portion of the distantly supervised data to be manually labeled by human annotators. To provide a solid evaluation setting for future research works, the human-annotated data will be used as the development and testing set. We include the development set in the annotated portion as it is necessary for hyperparameter tuning and model selection.

The goal of the human annotation stage is to identify correct alignments and remove invalid alignments. During the process, the annotators are tasked to review the correctness of each aligned fact, where an aligned fact consists of the sentence s_i and hyper-relational fact f. The alignment may be invalid if the relation triplet of the fact is not semantically expressed in the sentence, based on the Wikidata relation meaning. For instance, given the sentence “Prince Koreyasu was the son of Prince Munetaka who was the sixth shogun.”, the relation triplet (Prince Koreyasu, Occupation, shogun) is considered invalid as the sentence did not explicitly state if “Prince Koreyasu” became a shogun. Similarly, the alignment may be invalid if the qualifier of the fact is not semantically expressed in the sentence, based on the Wikidata definition of the qualifier label. For example, given the sentence “Robin Johns left Northamptonshire at the end of the 1971 season.”, the hyper-relational fact (Robin Johns, member of sports team, Northamptonshire, Start Time, 1971) has an invalid qualifier as the

Dataset	#Train	#Dev	#Test	#Facts	$\|R\|$	$\|Q\|$
TACRED	37,311	10,233	6,277	68,586	41	0
NYT24	56,196	5,000	5,000	17,624	24	0
NYT29	63,306	7,033	4,006	18,479	29	0
HyperRED	39,840	1,000	4,000	44,372	62	44

Table 2: Comparison of existing sentence-level datasets with HyperRED. "#Fact" denotes the unique facts, $|R|$ and $|Q|$ denote the unique relation labels and qualifier labels, respectively. To our knowledge, HyperRED is the first RE dataset to include hyper-relational facts.

label should be changed to "End Time". Hence, the annotation is posed as a multi-class classification over each alignment with three classes: "correct", "invalid triplet" or "invalid qualifier". Appendix A has the annotation guide and data samples.

Each alignment sample is annotated by two professional annotators working independently. There are 6780 sentences annotated in total and the inter-annotator agreement is measured using Cohen's kappa with a value of 0.56. The kappa value is comparable with previous relation extraction datasets (Zhang et al., 2017), demonstrating that the annotations are of reasonably high quality. For each sample with disagreement, a third annotator is brought to judge the final result. We observe that 76% of samples are annotated as "correct", which indicates a reasonable level of accuracy in the distantly supervised data. To reduce the long-tailed class imbalance (Zhang et al., 2019), we use a filter to ensure that all relation and qualifier labels have at least ten occurrences in the dataset. Although it can be more realistic to include challenging samples such as long-tailed class samples or negative samples in the dataset, we aim to address such challenges in a future dataset version release.

2.3 Data Analysis

To provide a better understanding of the HyperRED dataset, we analyze several aspects of the dataset.

Qualifier Typology The qualifiers of the hyper-relational facts can be grouped into several broad categories as shown in Table 1. Notably, the majority of the qualifiers fall under the "Time" category, as it can be considered a fundamental attribute of many facts. The remaining qualifiers are distributed among the "Quantity", "Role", "Part-Whole" and "Location" categories. Hence, the HyperRED dataset is able to support a diverse typology of hyper-relational facts.

Size and Coverage The statistics of HyperRED are shown in Table 2. We find that in terms of size and number of relation types, HyperRED is comparable to existing sentence-level datasets, such as TACRED (Zhang et al., 2017), NYT24 and NYT29 (Nayak and Ng, 2020). Table 1 also demonstrates that HyperRED can serve as a general-purpose dataset, covering several domains such as business, sports and politics. Appendix C has more details.

3 CubeRE: A Cube-Filling Approach

3.1 Task Formulation

Hyper-Relational Extraction Given an input sentence of n words $s = \{x_1, x_2, ..., x_n\}$, an entity e is a consecutive span of words where $e = \{x_i, x_{i+1}, ..., x_j\}, i, j \in \{1, ..., n\}$. For each sentence s, the output of a hyper-relational extraction model is a set of facts where each fact consists of a relation triplet with an attributed qualifier. A relation triplet consists of the relation $r \in R$ between head entity e_{head} and tail entity e_{tail} where R is the predefined set of relation labels. The qualifier is an attribute of the relation triplet and is composed of the qualifier label $q \in Q$ and the value entity e_{value}, where Q is the predefined set of qualifier labels. Hence, a hyper-relational fact has five components: $(e_{head}, r, e_{tail}, q, e_{value})$.

Cube-Filling Inspired by table-filling approaches which can naturally perform binary relation extraction in an end-to-end fashion, we cast hyper-relational extraction as a cube-filling problem, as shown in Figure 2. The cube contains multiple planes where the front-most plane is a two-dimensional table containing the entity and relation label information, while the following planes contain the corresponding qualifier information. Each entry on the table diagonal represents a possible entity, while each entry outside the table diagonal represents a possible relation triplet. For example, the entry "Educated At" represents a relation between the head entity "Parker" and the tail entity "Harvard". Each table entry y^t_{ij} can contain the null label $\perp$, an entity or relation label, i.e., $y^t_{ij} \in Y^t = \{\perp, \text{Entity}\} \cup R$.

The following planes in the cube represent the qualifier dimension, where each entry represents a possible qualifier label and value entity word for the corresponding relation triplet. For instance, the entry "Academic Degree" in the qualifier plane for "PhD" corresponds to the relation triplet (Parker, Educated At, Harvard), hence forming the hyper-relational fact (Parker, Educated At, Harvard, Academic Degree, PhD). Each qualifier entry y^q_{ijk} can contain the null label $\perp$ or a qualifier label, i.e., $y^q_{ijk} \in Y^q = \{\perp\} \cup Q$. Note that the cube-filling formulation also supports hyper-relational facts that share the same relation triplet, as the different qualifiers can occupy separate planes in the qualifier dimension and still correspond to the same relation triplet entry.

3.2 Model Architecture

Our model known as CubeRE first encodes each input sentence using a language model encoder to obtain the contextualized sequence representation. We then capture the interaction between each possible head and tail entity as a pair representation for predicting the entity-relation label scores. To reduce the computational cost, each sentence is pruned to retain only words that have higher entity scores. Finally, we capture the interaction between each possible relation triplet and qualifier to predict the qualifier label scores and decode the outputs.

3.2.1 Sentence Encoding

To encode a contextualized representation for each word in a sentence s, we use the pre-trained BERT (Devlin et al., 2019) language model:

$$\{h_1, h_2, ..., h_n\} = \text{BERT}(\{x_1, x_2, ..., x_n\}) \quad (1)$$

where h_i denotes the contextualized representation of the i-th word in the sentence.

3.2.2 Entity-Relation Representation

To capture the interaction between head and tail entities, we concatenate each possible pair of word representations and project with a dimension-reducing feed-forward network (FFN):

$$g_{ij} = \text{FFN}_{pair}(h_i \oplus h_j) \quad (2)$$

Thus, we construct the table of categorical probabilities over entity and relation labels by applying an FFN and softmax over the pair representation:

$$P(\hat{y}^t_{ij}) = \text{Softmax}(\text{FFN}_t(g_{ij})) \quad (3)$$

where $\hat{y}^t_{ij}$ denotes the predicted table entry corresponding to the relation between the i-th possible head entity word and j-th possible tail entity word. Note that we use the concatenation operation in Equation 2 instead of the averaging operation or other representation methods (Baldini Soares et al., 2019) as the concatenation operation is simple and shown to be effective in recent RE works (Wang et al., 2021a; Wang and Lu, 2020).

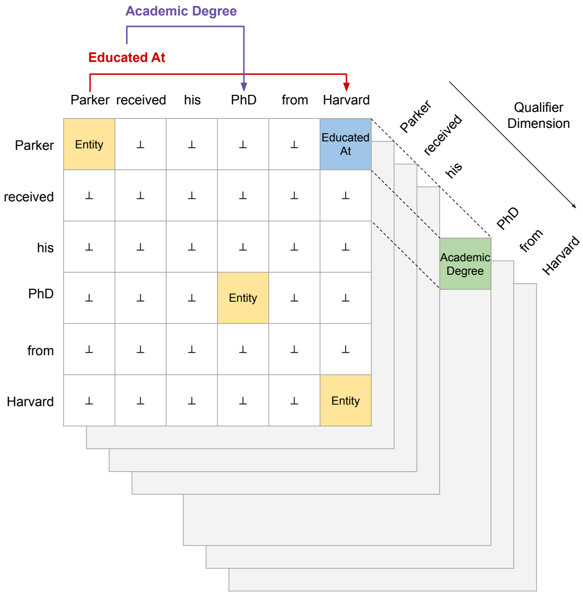

Figure 2: An example of cube-filling for hyper-relational extraction. The front-most plane is a two-dimensional table that contains entity and relation information. It extends to the third dimension where each plane represents a possible qualifier label and value entity word that corresponds to the relation triplet entry.

3.2.3 Cube-Pruning

To predict the qualifier of a hyper-relational fact, the model needs to consider the interaction between each possible relation triplet and value entity, where the relation triplet contains a head entity and a tail entity. For a sentence with n words, there are n^3 interactions that do not scale well for longer input sequences. Hence, we propose a cube-pruning method to consider only interactions between the top m words in terms of entity score. Consequently, the model will only consider the interaction between the top-m most probable words of the potential head entities, tail entities, and value entities respectively. This reduces the number of interactions to m^3 where m is a fixed hyperparameter. The cube-pruning method also has the benefit of alleviating the negative class imbalance by reducing the proportion of entries with the null label, and we analyze this effect in Section 5.1. To detect the most probable entity words, we obtain the respective entity scores from the diagonal of the table $\hat{y}^t$ containing the entity and relation scores (i.e., the front-most plane in Figure 2):

$$\Phi^{entity}_i = P(\hat{y}^t_{ii}), i \in \{1, ..., n\} \quad (4)$$

The entity scores are then ranked to obtain the pruned indices $\{1, ..., m\}$ which will be applied to each dimension of the cube representation.

To capture the hyper-relational structure between relation triplets and qualifier attributes, we use a bilinear interaction layer between each possible pair representation and word representation. The categorical probability distribution over qualifier labels for each possible relation triplet and value entity is then computed as:

$$P(\hat{y}^{q}_{i'j'k'}) = \text{Softmax}(g^{\intercal}_{i'j'}\ U\ h_{k'}) \tag{5}$$

where $i', j', k' \in \{1, ..., m\}$ are the pruned indices and U is a trainable bilinear weight matrix.

3.2.4 Training Objective

The training objective for the entity-relation table is computed using the negative log-likelihood as:

$$\mathcal{L}_t = -\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\log P(\hat{y}^{t}_{ij}) \tag{6}$$

The training objective for the qualifier dimension is computed using the negative log-likelihood as:

$$\mathcal{L}_q = -\frac{1}{m^3}\sum_{i'=1}^{m}\sum_{j'=1}^{m}\sum_{k'=1}^{m}\log P(\hat{y}^{q}_{i'j'k'}) \tag{7}$$

To enable end-to-end training, the overall cube-filling objective is aggregated as the sum of losses:

$$\mathcal{L} = \mathcal{L}_t + \mathcal{L}_q \tag{8}$$

3.2.5 Decoding

To decode the hyper-relational facts from the predicted scores, we implement a simple and efficient method and provide the pseudocode in Appendix D. As it is intractable to consider all possible solutions, a slight drop in decoding accuracy is acceptable. A key intuition is that if a valid qualifier exists, this indicates that a corresponding relation triplet also exists. Hence, we first decode the qualifier scores (Equation 5) to determine the span positions of the head entity, tail entity and value entity in each hyper-relational fact. Consequently, we can determine the relation and qualifier label from the corresponding entries in the relation scores (Equation 3) and qualifier scores respectively.

To handle entities that may contain multiple words, we consider adjacent non-null qualifier entries to correspond to the same head entity, tail entity, and value entity, hence belonging to the same hyper-relational fact. This assumption holds true for 97.14% of facts in the dataset. To find and merge the adjacent non-null entries, we use the nonzero operation which is more computationally efficient compared to nested for-loops. For each group of adjacent entries that correspond to the same hyper-relational fact, we determine the relation label by averaging the corresponding relation scores. Similarly, we determine the qualifier label by averaging the corresponding qualifier scores. When using cube-pruning, we map the pruned indices back to the original indices before decoding. Appendix E has the model speed comparison.

4 Experiments

4.1 Experimental Settings

Evaluation Similar to other information extraction tasks, we use the Micro F_1 metric for evaluation on the development and test set. For a predicted hyper-relational fact to be considered correct, the whole fact $f = (e_{head}, r, e_{tail}, q, e_{value})$ must match the ground-truth fact in terms of relation label, qualifier label and entity bounds.

Hyperparameters For the encoding module, we use the BERT language model, specifically the uncased base and large versions. We train for 30 epochs with a linear warmup for 20% of training steps and a maximum learning rate of 5e-5. We employ AdamW as the optimizer and use a batch size of 32. For model selection and hyperparameter selection, we evaluate based on the F_1 on the development set. We use $m = 20$ for cube-pruning and Appendix B has more experimental details.

4.2 Baseline Methods

As there are no existing models for hyper-relational extraction, we introduce two strong baselines that leverage pretrained language models. The pipeline baseline is based on a competitive table-filling model for joint entity and relation extraction, while the generative baseline is extended from a state-of-the-art approach for end-to-end relation extraction.

Pipeline Baseline As pipeline methods can serve as strong baselines for information extraction tasks (Zhong and Chen, 2021), we implement a pipeline method for hyper-relational extraction. Concretely, we first train a competitive relation extraction model architecture UniRE (Wang et al., 2021a) to extract relation triplets from each input sentence. Separately, we train a span extraction model based on BERT-Tagger (Devlin et al., 2019) that is conditioned on the input sentence and a relation triplet

Model	Parameters	Dev			Test		
		Precision	Recall	F_1	Precision	Recall	F_1
Generative Baseline (Base)	140M	63.79 ± 0.27	59.94 ± 0.68	61.80 ± 0.37	64.60 ± 0.47	59.67 ± 0.35	62.03 ± 0.21
Pipeline Baseline (Base)	132M	$\mathbf{69.23} \pm 0.30$	58.21 ± 0.57	63.24 ± 0.44	$\mathbf{69.00} \pm 0.48$	57.55 ± 0.19	62.75 ± 0.29
CubeRE (Base)	115M	66.14 ± 0.88	$\mathbf{64.39} \pm 1.23$	$\mathbf{65.24} \pm 0.82$	65.82 ± 0.84	$\mathbf{64.28} \pm 0.25$	$\mathbf{65.04} \pm 0.29$
Generative Baseline (Large)	400M	67.08 ± 0.49	65.73 ± 0.78	66.40 ± 0.47	$\mathbf{67.17} \pm 0.40$	64.56 ± 0.58	65.84 ± 0.25
CubeRE (Large)	343M	$\mathbf{68.75} \pm 0.82$	$\mathbf{68.88} \pm 1.03$	$\mathbf{68.81} \pm 0.46$	66.39 ± 0.96	$\mathbf{67.12} \pm 0.69$	$\mathbf{66.75} \pm 0.65$

Table 3: Evaluation results for hyper-relational extraction on the HyperRED dataset.

to extract the value entities and corresponding qualifier label. However, as both stages fine-tune a pretrained language model, the pipeline method doubles the number of trainable parameters compared to an end-to-end method which only fine-tunes one pretrained language model. To avoid an unfair comparison as larger models are more sample-efficient (Kaplan et al., 2020), we use DistilBERT (Sanh et al., 2019) in both stages of the pipeline.

Generative Baseline Inspired by the flexibility of language models for complex tasks such as information extraction and controllable structure generation (Shen et al., 2022), we propose a generative method for hyper-relational extraction. Compared to a pipeline method, a generative method can perform hyper-relational extraction in an end-to-end fashion without task-specific modules (Paolini et al., 2021). Similar to existing generative methods for relation extraction (Huguet Cabot and Navigli, 2021; Chia et al., 2022), we use BART (Lewis et al., 2020) which takes the sentence as input and outputs a structured text sequence that is then decoded to form the extracted facts. For instance, given the sentence "Parker received his PhD from Harvard.", the sequence-to-sequence model is trained to generate "Head Entity: Parker, Relation: educated at, Tail Entity: Harvard, Qualifier: academic degree, Value: PhD." The generated text is then decoded through simple text processing to form the hyper-relational fact (Parker, Educated At, Harvard, Academic Degree, PhD).

4.3 Main Results

We compare CubeRE with the baseline models and report the precision, recall, and F_1 scores with standard deviation in Table 3. The results demonstrate the general effectiveness of our model as CubeRE has consistently higher F_1 scores on both the base and large model settings. While the pipeline baseline relies on a two-stage approach that is prone to error propagation, CubeRE can perform hyper-relational extraction in an end-to-end fashion. Hence, CubeRE is able to detect more valid hyper-relational facts, which is demonstrated by the higher recall and F_1 scores. Compared to the generative baseline, our cube-filling approach is able to explicitly consider the interaction between relation triplets and qualifiers to better extract hyper-relational facts. Furthermore, we argue that CubeRE is more interpretable than the generative baseline as it can compute the score for each possible relation triplet and qualifier. Hence, CubeRE can also be more controllable as it is possible to control the number of predicted facts by applying a threshold to the triplet and qualifier scores.

Model	Precision	Recall	F_1
Generative Baseline	69.96 ± 0.31	64.56 ± 0.21	67.15 ± 0.09
Pipeline Baseline	75.94 ± 0.66	66.41 ± 0.72	70.85 ± 0.13
CubeRE	72.45 ± 0.66	69.64 ± 0.53	71.01 ± 0.16

Table 4: Evaluation results on HyperRED considering only the triplet component of hyper-relational facts.

4.4 Triplet-Based Evaluation

To further investigate the differences in model performance, we also report the results when considering only the triplet component of hyper-relational facts in Table 4. The results show that CubeRE has comparable performance to the pipeline baseline when considering only relation triplets. Hence, this suggests that the performance improvement in hyper-relational extraction is most likely due to more accurate qualifier extraction. Compared to the pipeline baseline which has two separate encoders for triplet extraction and conditional qualifier extraction, CubeRE learns a shared representation of the input sentence that is guided by both the triplet and qualifier losses facilitating the interaction between relation triplets and qualifiers. The triplet-qualifier interaction is important as most qualifier labels are relatively relation-specific[2]. This allows CubeRE to extract the qualifiers more accurately, resulting in better overall performance.

[2]Please refer to Appendix C for the qualifier analysis.

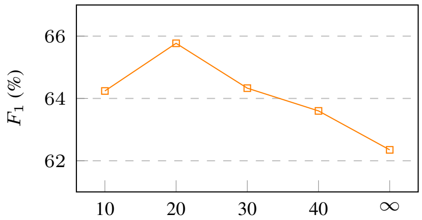

Figure 3: The effect of pruning threshold m on Dev F_1. The model without pruning is indicated as $m = \infty$.

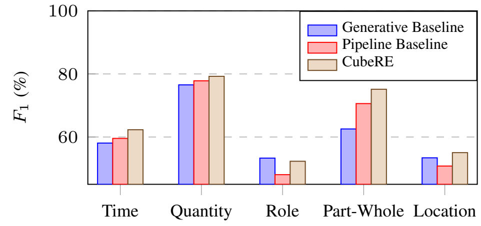

Figure 4: Model performance breakdown based on the general categories of qualifiers as shown in Table 1.

5 Analysis

In this section, we study the effect of cube-pruning and identify directions for future research. Further analysis is shown in Appendix F.

5.1 Effect of Pruning

In addition to improving the computational efficiency of CubeRE as discussed in Section 3.2.3, our cube pruning method may also improve the extraction performance of the model. During training, the cube-filling approach faces the issue of having mostly null entries, thus biasing the learning process with negative class imbalance (Li et al., 2020). By pruning the cube to consider only the entries associated with higher entity scores, the proportion of null entries is reduced, hence alleviating the class imbalance issue. This is supported by the trend in Figure 3, as relaxing the pruning threshold m leads to reduced F_1 scores. On the other hand, overly strict pruning will reduce the recall, negatively affecting the overall performance.

5.2 Model Performance Breakdown

To identify directions for future research in hyper-relational extraction, we analyze the model performance separately for each general qualifier category. As shown in Table 4, there is a variance in model performance across qualifier categories that cannot be fully explained by their proportion in the dataset. For instance, although the "Time" category comprises a majority of the qualifiers, it does not have the highest performance. This suggests that future research may focus on areas such as temporal reasoning, which is an open challenge for language models (Vashishtha et al., 2020; Dhingra et al., 2022). In addition, CubeRE demonstrates strong performance across all categories which suggests that it can serve as a general extraction model for different qualifiers.

6 Related Work

Knowledge Graph Construction In addition to extraction from natural language text, the underlying facts for knowledge graphs can also be extracted from semi-structured websites (Lockard et al., 2018), tables (Dong et al., 2020) or link prediction (Wang et al., 2017). However, textual extraction may be a more pressing challenge due to the vast amount of unstructured textual data on the web (Lockard et al., 2020). Hence, this work focuses on extracting facts from unstructured text.

Relation Extraction Although relation extraction is a well-established task, most methods only consider the relation between two entities. There have been several directions to extract more complex facts, such as n-ary relation extraction or document-level relation extraction (Yao et al., 2019). However, n-ary relation extraction (Jia et al., 2019; Akimoto et al., 2019) has a limited scope as the available datasets address the biomedical domain. On the other hand, document-level (Tan et al., 2022a) and cross-document relation extraction (Yao et al., 2021) are fundamentally limited by the binary relation structure which does not consider hyper-relational information. Although dialogue-level relation extraction (Chen et al., 2020) may have a more complex structure consisting of utterances and speaker information, current datasets (Welleck et al., 2019) focus on the binary relation format. Hence, we propose to fill the gap by contributing HyperRED, a general-purpose and large-scale dataset for hyper-relational extraction that is not limited to any specific domain.

Information Extraction In this work, we focus on relation extraction which falls under the broad scope of information extraction (Bing et al., 2015). Hence, a possible future direction is to adapt CubeRE for extracting other types of information such as attributes (Bing et al., 2013), events (Wang et al., 2021b), arguments (Cheng et al., 2020, 2022), aspect-based sentiment (Xu et al., 2021; Yu Bai Jian et al., 2021), commonsense knowledge (Ghosal et al., 2021), or visual scene relations (Andrews et al., 2019). Additionally, as HyperRED relies on distant supervision for dataset construction, it is necessary to further explore how to mitigate the noise in distantly supervised datasets for information extraction tasks (Nayak et al., 2021).

Table-Filling Table-Filling is a popular approach for entity and relation extraction tasks (Miwa and Sasaki, 2014; Gupta et al., 2016; Zhang et al., 2017). It has several advantages including interpretability and an end-to-end formulation. Hence, table-filling approaches are able to avoid the cascading error propagation faced by pipeline models, despite a compact parameter set. Inspired by the benefits of table-filling, we extend the approach to cube-filling to extract hyper-relational facts by considering qualifiers for each relation triplet. To our knowledge, our proposed model is the first cube-filling approach for information extraction tasks.

7 Conclusions

In this work, we propose the hyper-relational extraction task for extracting richer and more complete facts from natural text. To support the task, we introduce HyperRED, a large-scale and general-purpose dataset that is not restricted to any specific domain. As there is no available model for hyper-relational extraction, we propose an end-to-end cube-filling approach inspired by table-filling methods for relation extraction. We further propose a cube-pruning method to reduce computational cost and alleviate negative class imbalance during training. Experiments on HyperRED demonstrate the effectiveness of CubeRE compared to strong baselines, setting the benchmark for future work.

Limitations

Model Limitations Regarding the CubeRE model, we propose a cube-pruning method to improve the computational efficiency and reduce the negative class imbalance. The cube-pruning threshold is fixed, although the input can have different sentence lengths. Hence, it may result in overly strict pruning if the sentence is extremely long. However, the pruning threshold is similar to the maximum sequence length in most transformer-based models and may need to be tuned according to the specific dataset or application scenario. The optimal cube-pruning threshold is selected based on the analysis in Section 5.1. CubeRE may not work well for overlapping or nested entity spans, which affects 2.11% of the sentences. This can be considered a general limitation of table-filling methods for relation extraction, and future work may need to consider a span-based approach (Xu et al., 2021) to address this issue.

Data Limitations Regarding the HyperRED dataset, the distant supervision method of data collection may not align all valid facts present in the text articles. This is due to the possible incompleteness of the knowledge graph which is an open research challenge (Nickel et al., 2016). On the other hand, it is not feasible to manually annotate all possible facts due to constraints in annotation time and cost. Furthermore, there are a large number of relation and qualifier labels to consider, resulting in a challenging task for human annotators. A promising and practical method to address the challenges in distant supervision is to adopt a human-in-the-loop annotation scheme for RE (Tan et al., 2022b). The annotation scheme can increase the number of facts in a dataset by training a RE model to predict more candidate facts for each text article, which are then reviewed and filtered by humans. However, this model-assisted annotation approach is not applicable to the construction of HyperRED as it relies on existing strong RE models, whereas there are no suitable models for hyper-relational extraction existing prior to this work.

Ethics Statement

Model Ethics Regarding the model generalization, we expect that the models introduced should perform similarly for factual text articles such as news articles from various domains, similar to the proposed dataset. However, it may not perform well for more casual text formats such as chat discussions or opinion pieces. On the other hand, we note that the models extract hyper-relational facts from the input sentences and do not guarantee the factual correctness of the extracted facts. This is an ethical consideration of RE models in general and further fact verification (Nie et al., 2019) modules are necessary before the facts can be integrated into knowledge graphs or downstream applications.

Data Ethics For the dataset construction, we collect texts and facts from Wikipedia and Wikidata respectively, which is a common practice for distantly supervised datasets. Wikidata facts are under the public domain[3] while Wikipedia texts are

[3]https://www.wikidata.org/wiki/Wikidata:Licensing

licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License[4]. Hence, we are free to adapt the texts to construct our dataset, which will also be released under the same license. For the human data annotation stage, we employ two professional data annotators, and they have been fairly compensated. The compensation is negotiated based on the task complexity and assessment of the reasonable annotation speed. Based on the agreed annotation scheme, each annotation batch is required to undergo quality checking where a portion of samples are manually checked. If any batch does not meet the acceptance criteria of 95% accuracy, the annotators are required to fix the errors before the batch can be accepted. The overall quality of the dataset is evaluated in Section 2.1 and Section 2.2, and we analyze the dataset characteristics in Section 2.3, with further analysis in Section F.

[4]https://en.wikipedia.org/wiki/Wikipedia:Copyrights

References

Kosuke Akimoto, Takuya Hiraoka, Kunihiko Sadamasa, and Mathias Niepert. 2019. Cross-sentence n-ary relation extraction using lower-arity universal schemas. 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 6225–6231, Hong Kong, China. Association for Computational Linguistics.

Martin Andrews, Yew Ken Chia, and Sam Witteveen. 2019. Scene graph parsing by attention graph. *In Proceedings of the Second Workshop on Visually Grounded Interaction and Language (ViGIL) at NeurIPS 2018*.

Nguyen Bach and Sameer Badaskar. 2007. A review of relation extraction. *Literature review for Language and Statistics II*, 2:1–15.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Lidong Bing, Sneha Chaudhari, Richard Wang, and William Cohen. 2015. Improving distant supervision for information extraction using label propagation through lists. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 524–529, Lisbon, Portugal. Association for Computational Linguistics.

Lidong Bing, Wai Lam, and Tak-Lam Wong. 2013. Wikipedia entity expansion and attribute extraction from the web using semi-supervised learning. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, page 567–576, New York, NY, USA. Association for Computing Machinery.

Hui Chen, Pengfei Hong, Wei Han, Navonil Majumder, and Soujanya Poria. 2020. Dialogue relation extraction with document-level heterogeneous graph attention networks. *CoRR*, abs/2009.05092.

Liying Cheng, Lidong Bing, Ruidan He, Qian Yu, Yan Zhang, and Luo Si. 2022. IAM: A comprehensive and large-scale dataset for integrated argument mining tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2277–2287, Dublin, Ireland. Association for Computational Linguistics.

Liying Cheng, Lidong Bing, Qian Yu, Wei Lu, and Luo Si. 2020. APE: Argument pair extraction from peer review and rebuttal via multi-task learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7000–7011, Online. Association for Computational Linguistics.

Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. RelationPrompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 45–57, Dublin, Ireland. Association for Computational Linguistics.

Yew Ken Chia, Sam Witteveen, and Martin Andrews. 2020. Red dragon AI at TextGraphs 2020 shared task : LIT : LSTM-interleaved transformer for multi-hop explanation ranking. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 115–120, Barcelona, Spain (Online). Association for Computational Linguistics.

Mark Craven and Johan Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, page 77–86. AAAI Press.

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.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Xin Luna Dong, Hannaneh Hajishirzi, Colin Lockard, and Prashant Shiralkar. 2020. Multi-modal information extraction from text, semi-structured, and tabular data on the web. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 23–26, Online. 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).

Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandecic. 2014. Introducing wikidata to the linked data web. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, volume 8796 of *Lecture Notes in Computer Science*, pages 50–65. Springer.

Mikhail Galkin, Priyansh Trivedi, Gaurav Maheshwari, Ricardo Usbeck, and Jens Lehmann. 2020. Message passing for hyper-relational knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7346–7359, Online. Association for Computational Linguistics.

Deepanway Ghosal, Pengfei Hong, Siqi Shen, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2021. CIDER: Commonsense inference for dialogue explanation and reasoning. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 301–313, Singapore and Online. Association for Computational Linguistics.

Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2019. Link prediction on n-ary relational data. In *The World Wide Web Conference*, WWW '19, page 583–593, New York, NY, USA. Association for Computing Machinery.

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.

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.

Eduard Hovy, Roberto Navigli, and Simone Paolo Ponzetto. 2013. Collaboratively built semi-structured content and artificial intelligence: The story so far. *Artificial Intelligence*.

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.

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.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Douglas B. Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Commun. ACM*.

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.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Colin Lockard, Xin Luna Dong, Arash Einolghozati, and Prashant Shiralkar. 2018. Ceres: Distantly supervised relation extraction from the semi-structured web. *Proc. VLDB Endow.*, 11(10):1084–1096.

Colin Lockard, Prashant Shiralkar, Xin Luna Dong, and Hannaneh Hajishirzi. 2020. *Web-Scale Knowledge Collection*, page 888–889. Association for Computing Machinery, New York, NY, USA.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, page 1–8, New York, NY, USA. Association for Computing Machinery.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. 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.

Tapas Nayak, Navonil Majumder, and Soujanya Poria. 2021. Improving distantly supervised relation extraction with self-ensemble noise filtering. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1031–1039, Held Online. INCOMA Ltd.

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.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6859–6866.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, RISHITA ANUBHAI, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *International Conference on Learning Representations*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.

Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. *Beyond Triplets: Hyper-Relational Knowledge Graph Embedding for Link Prediction*, page 1885–1896. Association for Computing Machinery, New York, NY, USA.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Chenhui Shen, Liying Cheng, Ran Zhou, Lidong Bing, Yang You, and Luo Si. 2022. MReD: A meta-review dataset for structure-controllable text generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2521–2535, Dublin, Ireland. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Qingyu Tan, Ruidan He, Lidong Bing, and Hwee Tou Ng. 2022a. Document-level relation extraction with adaptive focal loss and knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1672–1681, Dublin, Ireland. Association for Computational Linguistics.

Qingyu Tan, Lu Xu, Lidong Bing, and Hwee Tou Ng. 2022b. Revisiting docred - addressing the overlooked false negative problem in relation extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4070–4078, Online. Association for Computational Linguistics.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021a. UniRE: A unified label space for entity relation extraction. 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 220–231, Online. Association for Computational Linguistics.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021b. CLEVE: Contrastive Pre-training for Event Extraction. 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 6283–6297, Online. Association for Computational Linguistics.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. On the representation and embedding of knowledge bases beyond binary relations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 1300–1307. AAAI Press.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1271–1279, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Lu Xu, Yew Ken Chia, and Lidong Bing. 2021. Learning span-level interactions for aspect sentiment triplet extraction. 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 4755–4766, Online. Association for Computational Linguistics.

Yuan Yao, Jiaju Du, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. 2021. CodRED: A cross-document relation extraction dataset for acquiring knowledge in the wild. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4452–4472, Online and Punta Cana, Dominican Republic. 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.

Wen-tau Yih and Hao Ma. 2016. Question answering with knowledge base, web and beyond. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, pages 8–10, San Diego, California. Association for Computational Linguistics.

Samson Yu Bai Jian, Tapas Nayak, Navonil Majumder, and Soujanya Poria. 2021. Aspect sentiment triplet extraction using reinforcement learning. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM '21, page 3603–3607, New York, NY, USA. Association for Computing Machinery.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. 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 3016–3025, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

A Annotation Guide

This section explains the guideline for human annotators. The task is a classification of whether each hyper-relational fact can be reasonably extracted from a piece of text. Each annotation sample contains one sentence and one corresponding fact for judgment. The annotator should classify each sample as “Correct” or “Invalid Triplet” or “Invalid Qualifier”. Each hyper-relational fact has five components with the format (head entity, relation label, tail entity, qualifier label, value entity). The head entity is the main subject entity of the relationship. The relation label is the category of relationship that is expressed between the head and tail entity. The tail entity is the object entity of the relationship that is paired with the head entity. The qualifier label is the category of the qualifier information. The value entity is the corresponding value of the qualifier that is applied to the relation triplet (head, relation, tail).

The value entity can contain a date, quantity, or short piece of text which is the mentioned name of the entity. For the annotation objective, we want to know whether this piece of information is clearly expressed by the given text. All the entities, relations, and qualifiers exist in the Wikidata database, so annotators can refer to the relation or qualifier definition at https://www.wikidata.org for clarification. The annotation steps are as follows:

1. Read and understand the text sample which is a continuous sequence of words. Then, consider the corresponding hyper-relational fact.
2. First check the triplet (head, relation, tail) of the fact. If the head and tail entity mentioned in the text do not clearly express the relation's meaning, then the whole fact should be marked as “Invalid Triplet”.
3. Check the (qualifier, value) components. If the value mentioned in the text does not clearly express the qualifier meaning or is not directly

Data Setting	Annotation Type	Sentences	Facts	Entities	Average Sentence Length	Average Entity Length
Train	Distant-Supervised	39,840	39,978	32,539	31.91 words	1.67 words
Dev	Human Annotated	1,000	1,220	1,912	30.30 words	1.71 words
Test	Human Annotated	4,000	4,796	5,842	30.06 words	1.69 words

Table 5: Detailed statistics for the HyperRED dataset.

related to the triplet, then the fact should be marked as "Invalid Qualifier".

4. If there is no error in the fact, then it can be marked as "Correct".

For example, given the sentence "The film's story earned Leonard Spigelgass a nomination as Best Story for the 23rd Academy Awards.", the fact (Leonard Spigelgass, nominated for, Best Story, statement is subject of, 23rd Academy Awards) is correct as Leonard was nominated and the main topic is the Academy Awards. However, given the sentence "Prince Koreyasu was the son of Prince Munetaka who was the sixth shogun.", the fact (Prince Koreyasu, occupation, shogun, replaces, Prince Munetaka) has an invalid triplet as we don't know if Koreyasu became a shogun. On the other hand, given the sentence "Robin Johns left Northamptonshire at the end of the 1971 season.", the fact (Robin Johns, member of sports team, Northamptonshire, Start Time, 1971) has an invalid qualifier as the qualifier label should be "End Time" instead of "Start Time".

B Experiment Details

Hyperparameters Table 8 shows the details of our experimental setup and model hyperparameters. For the analysis experiments in Section 5, we use the BERT-Base version of CubeRE and report the F_1 metric score on the development set of HyperRED unless otherwise stated in the specific subsection.

Pipeline Baseline Details For the pipeline baseline, we use DistilBERT as the language model encoder for both the triplet extraction and conditional qualifier extraction stages. Both stages of the pipeline are fine-tuned separately on the gold labels. At inference time, the triplet extraction stage takes the sentence as input and outputs the predicted relation triplets. For each predicted relation triplet, the conditional qualifier extractor takes the sentence and the relation triplet as input to predict the possible qualifiers where each qualifier consists of the qualifier label and value entity. The input of the qualifier extraction model is the concatenated sentence and relation triplet. For example, the sentence "Leonard Parker received his PhD from Harvard University in 1967." and relation triplet (Leonard Parker, Educated At, Harvard University), will be concatenated to become 'Leonard Parker received his PhD from Harvard University in 1967. Leonard Parker | Educated At | Harvard University". The outputs of both stages are then merged to form the predicted hyper-relational facts. Following the BERT-Tagger, the conditional qualifier extraction model is trained using the crossentropy loss for sequence labeling. To encode the qualifier information as sequence labels, we use the BIO tagging scheme where the sequence label corresponds to the possible qualifier label for each entity word. For both stages which are trained separately, we use the same epochs, learning rate and batch size as the CubeRE model for fairness.

Generative Baseline Details The generative baseline model can predict hyper-relational facts by learning to generate a text sequence with a special structured format as demonstrated in Section 4.2. Note that if the sentence contains multiple hyper-relational facts, the desired output sequence is simply the concatenated text sequence of the structured text for each fact. The multiple facts can be easily decoded from the structured text format with simple text processing such as regex. As the input and output of the model are text sequences which do not violate the model vocabulary, the generative baseline can be trained using a standard sequence-to-sequence modeling objective. For training, we use the same epochs, learning rate and batch size as the CubeRE model for fairness.

C Dataset Details

Dataset Statistics Table 5 shows the detailed statistics of HyperRED, such as the number of unique facts and entities, as well as the average number of words in each sentence. Table 9 and Table 10 show the set of relation and qualifier labels respectively. For the construction of the

Model	Parameters	Dev			Test		
		Precision	Recall	F_1	Precision	Recall	F_1
Generative Baseline	140M	63.79 $\pm$ 0.27	59.94 $\pm$ 0.68	61.80 $\pm$ 0.37	64.60 $\pm$ 0.47	59.67 $\pm$ 0.35	62.03 $\pm$ 0.21
Pipeline Baseline	132M	**69.23** $\pm$ 0.30	58.21 $\pm$ 0.57	63.24 $\pm$ 0.44	**69.00** $\pm$ 0.48	57.55 $\pm$ 0.19	62.75 $\pm$ 0.29
CubeRE	115M	66.14 $\pm$ 0.88	**64.39** $\pm$ 1.23	**65.24** $\pm$ 0.82	65.82 $\pm$ 0.84	**64.28** $\pm$ 0.25	**65.04** $\pm$ 0.29
Pipeline Baseline (Medium)	221M	69.70 $\pm$ 1.08	62.33 $\pm$ 0.50	65.80 $\pm$ 0.54	69.38 $\pm$ 0.39	61.96 $\pm$ 0.54	65.46 $\pm$ 0.32
Generative Baseline (Large)	400M	67.08 $\pm$ 0.49	65.73 $\pm$ 0.78	66.40 $\pm$ 0.47	**67.17** $\pm$ 0.40	64.56 $\pm$ 0.58	65.84 $\pm$ 0.25
CubeRE (Large)	343M	**68.75** $\pm$ 0.82	**68.88** $\pm$ 1.03	**68.81** $\pm$ 0.46	66.39 $\pm$ 0.96	**67.12** $\pm$ 0.69	**66.75** $\pm$ 0.65
Pipeline Baseline (Large)	680M	70.58 $\pm$ 0.78	66.58 $\pm$ 0.66	68.52 $\pm$ 0.32	69.21 $\pm$ 0.55	64.27 $\pm$ 0.24	66.65 $\pm$ 0.28

Table 6: Evaluation results for hyper-relational extraction on the HyperRED dataset.

Model	Training Time	Inference Speed	Memory Usage
Generative	1.93 hrs	37 samples/s	3.9 GB
Pipeline	2.41 hrs	181 samples/s	5.5 GB
CubeRE	3.08 hrs	160 samples/s	6.6 GB

Table 7: Comparison of the computational cost for the Generative, Pipeline and CubeRE models.

	Experimental Detail
GPU Model	Nvidia V100
CUDA Version	11.3
Python Version	3.7.12
PyTorch Version	1.11.0
Wikidata Version	20170503
Long-Tailed Threshold	10
Pruning Threshold	20
Maximum Sequence Length (words)	80
FFN Hidden Size	150
Learning Rate Decay	0.9
Adam Epsilon	1e-12
Adam Weight Decay Rate	1e-5

Table 8: List of experimental details.

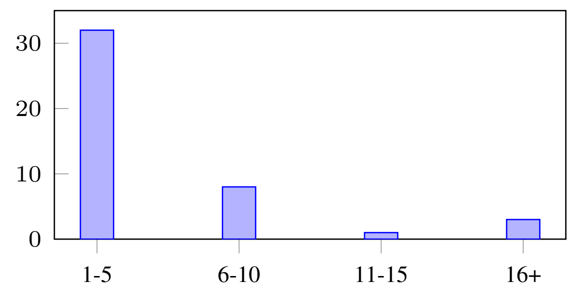

Figure 5: Histogram distribution of number of relation labels covered by each qualifier label.

dataset, we use the Wikidata which has 594,088 hyper-relational facts and introductions from English Wikipedia which has 4,650,000 articles.

Distant Supervision Example In this section, we demonstrate the distant supervision process for fact alignment with a sentence example. Given the input sentence "Leonard Parker received his PhD from Harvard University in 1967.", we first perform entity linking which detects the entity mentions and their Wikipedia IDs: {(Leonard Parker, Q3271532), (PhD, Q752297), (Harvard University, Q13371)}. As the entity linker does not consider dates or numbers, we use the spaCy tool to extract such spans: {(1967, Date)}. Hence, the set of linked entities in the sentence is {(Leonard Parker, Q3271532), (PhD, Q752297), (Harvard University, Q13371), (1967, Date)}. To address the case if the sentence contains unresolved pronouns such as "he" or "she", we use the Stanford CoreNLP tool to detect and resolve such cases to a suitable entity in the set of linked entities above. For each hyper-relational fact the in Wikidata knowledge graph, we attempt to align it to the sentence based on the entities in the fact. If the head entity, tail entity and value entity are all present in the linked entities set of the sentence, then it is a successful alignment. For example, given the fact (Leonard Parker, Educated At, Harvard University, End Time, 1967) where the head entity, tail entity and value entity is (Leonard Parker, Q3271532), (Harvard University, Q13371) and (1967, Date) respectively, the fact is successfully aligned with the sentence as the three entities are present in the set of linked entities. If any entities are missing from the set of linked entities, the alignment is unsuccessful and we do not include it in the dataset. If any sentence does not have any successfully aligned facts, we do not include it in the dataset.

Annotation Challenges The human annotation of the dataset may be imperfect due to complexity of the hyper-relational fact structure, diversity of relation and qualifier labels, and possible ambiguous facts. The hyper-relational facts require annotators to joint consider the relation triplet and qualifier which is more challenging compared to previous datasets which commonly consider the relation between two entities. On the other hand, the annotators are also required to consider the definitions of a large set of relation and qualifier labels. This may pose difficult when some relations or qualifiers are similar in meaning. Lastly, there may be ambiguous cases where multiple entities are mentioned in relation to a topic and it is not clear which entity is the main subject.

Relation-Specific Qualifiers To investigate the link between relation triplets and qualifiers, we plot a histogram distribution in Figure 5. A majority (32) of the qualifier labels are each linked to a small number of relation labels (1-5), which suggests that most qualifiers are highly relation-specific. For example, the "electoral district" qualifier label is only linked to the "candidacy in election" and "position held" relation labels. On the other hand, a few (3) qualifier labels are each linked to a large number (16+) of relation labels, and not specific to any particular relation. For example, the "end time" qualifier is linked to 35 relation labels. Hence, it is generally important to consider the interaction between relation triplets and qualifiers in extracting hyper-relational facts. However, it is not trivial to predict the qualifier only based on the relation, as some qualifier labels are relation-agnostic and it also requires the model to consider the value entity.

D Decoding Algorithm

Algorithm 1: Pseudocode of our decoding algorithm in a PyTorch-like style.

```
# y_t: Input entity-relation scores (Eq.3)
# y_q: Input qualifier scores (Eq.5)

facts = [] # Output hyper-relational facts
groups = [] # Hyper-relational span groups

# Find and merge adjacent non-null entries
for i,j,k in y_q.argmax(-1).nonzero():
    entry = (i,i+1,j,j+1,k,k+1)
    for spans in groups:
        if is_adjacent(spans, entry):
            merge(spans, entry)
            break
    else:
        groups.append(entry)

# Aggregate relation and qualifier scores
for spans in groups:
    i,i2,j,j2,k,k2 = spans
    r_scores = y_t[i:i2,j:j2]
    r_label = r_scores.mean(0,1).argmax()
    q_scores = y_q[i:i2,j:j2,k:k2]
    q_label = q_scores.mean(0,1,2).argmax()
    facts.append((spans,r_label,q_label))
```

We include the pseudocode algorithm of the proposed decoding method in Algorithm 1. Note that we can use the nonzero operation to find and merge adjacent non-null entries as it returns the entries sorted in lexicographic order. This ensures that the order of entries seen in consecutive order if they correspond to the same hyper-relational fact.

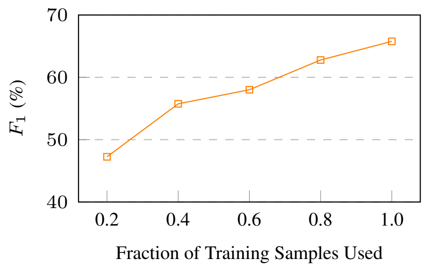

Figure 6: The effect of training data size on Dev F_1. The training set of HyperRED is distantly supervised, while the development and test set are human-annotated.

E Model Costs

Table 7 shows a comparison of total training time, inference speed in samples per second and GPU memory usage for the different models. We observe that CubeRE has a comparable computational cost with the generative and pipeline models. This result that our cube-pruning method is effective in ensuring that the model is computationally efficient and practical in real applications. Note that we compute the statistics for the two-stage pipeline model by summing the time taken and memory used by both stages.

F Further Analysis

Additional Pipeline Results For a fair comparison of main results in Section 4.3, we do not include the pipeline baseline in the large model setting as it would have 680M parameters which is much more than the other models. On the other hand, we also do not include a BERT-Base version of the pipeline baseline in the main results, as it would have 221M parameters which is not comparable to both the base and large model settings. Hence, we only include the pipeline baseline using DistilBERT in the main result discussion as it has a comparable parameter count to the base model setting. However, we include the pipeline baseline with BERT-Base in Table 6 for reference.

Effect of Pruning The main effect of cube-pruning is to reduce the sparsity of the cube entries by retaining the entries which are most likely to be valid entities. To quantify the effect on sparsity, we measure the cube without pruning to consist of 99.9900% null entries on average. When using pruning threshold $m = 20$, the cube consists of 99.9098% null entries on average. Hence, there is a roughly tenfold increase in the proportion of non-null entries when using pruning.

Effect of Training Data Size The HyperRED training set consists of distantly supervised data which enables large-scale and diverse model training. However, there may be noisy samples that affect the model performance. Hence, we aim to study whether the quantity of data can overcome noise in the training set. As shown in Figure 6, we observe a strictly increasing trend when the size of the training set is increased from 20% of the original size to 100% of the original size. Thus, the results suggest that the quantity of data is still a beneficial factor for model performance despite some noise in the distantly supervised training set.

Wiki ID	Label	Description
P6	head of government	head of the executive power of this town, city, municipality, state, country, or other governmental body
P17	country	sovereign state of this item (not to be used for human beings)
P19	place of birth	most specific known (e.g. city instead of country, or hospital instead of city) birth location of a person, animal or fictional cha racter
P26	spouse	the subject has the object as their spouse (husband, wife, partner, etc.). Use "unmarried partner" (P451) for non-married companions
P27	country of citizenship	the object is a country that recognizes the subject as its citizen
P31	instance of	that class of which this subject is a particular example and member
P35	head of state	official with the highest formal authority in a country/state
P39	position held	subject currently or formerly holds the object position or public office
P40	child	subject has object as child. Do not use for stepchildren
P47	shares border with	countries or administrative subdivisions, of equal level, that this item borders, either by land or water. A single common point is enough.
P54	member of sports team	sports teams or clubs that the subject represents or represented
P69	educated at	educational institution attended by subject
P81	connecting line	railway line(s) subject is directly connected to
P97	noble title	titles held by the person
P102	member of political party	the political party of which a person is or has been a member or otherwise affiliated
P106	occupation	occupation of a person; see also "field of work" (Property:P101), "position held" (Property:P39)
P108	employer	person or organization for which the subject works or worked
P115	home venue	home stadium or venue of a sports team or applicable performing arts organization
P118	league	league in which team or player plays or has played in
P127	owned by	owner of the subject
P131	located in the administrative territorial entity	the item is located on the territory of the following administrative entity.
P137	operator	person, profession, or organization that operates the equipment, facility, or service
P156	followed by	immediately following item in a series of which the subject is a part
P159	headquarters location	city, where an organization's headquarters is or has been situated. Use P276 qualifier for specific building
P161	cast member	actor in the subject production
P166	award received	award or recognition received by a person, organisation or creative work
P175	performer	actor, musician, band or other performer associated with this role or musical work
P176	manufacturer	manufacturer or producer of this product
P179	part of the series	series which contains the subject
P194	legislative body	legislative body governing this entity; political institution with elected representatives, such as a parliament/legislature or council
P197	adjacent station	the stations next to this station, sharing the same line(s)
P241	military branch	branch to which this military unit, award, office, or person belongs, e.g. Royal Navy
P276	location	location of the object, structure or event. In the case of an administrative entity as containing item use P131.
P279	subclass of	next higher class or type; all instances of these items are instances of those items; this item is a class (subset) of that item.
P361	part of	object of which the subject is a part
P414	stock exchange	exchange on which this company is traded
P449	original broadcaster	network(s) or service(s) that originally broadcasted a radio or television program
P463	member of	organization, club or musical group to which the subject belongs. Do not use for membership in ethnic or social groups
P466	occupant	person or organization occupying property
P488	chairperson	presiding member of an organization, group or body
P551	residence	the place where the person is or has been, resident
P641	sport	sport that the subject participates or participated in or is associated with
P669	located on street	street, road, or square, where the item is located.
P710	participant	person, group of people or organization (object) that actively takes/took part in an event or process (subject).
P725	voice actor	performer of a spoken role in a creative work such as animation, video game, radio drama, or dubbing over
P749	parent organization	parent organization of an organization, opposite of subsidiaries (P355)
P793	significant event	significant or notable events associated with the subject
P800	notable work	notable scientific, artistic or literary work, or other work of significance among subject's works
P1037	director / manager	person who manages any kind of group
P1327	partner in business or sport	professional collaborator
P1346	winner	winner of a competition or similar event, not to be used for awards
P1365	replaces	person, state or item replaced. Use "structure replaces" (P1398) for structures.
P1376	capital of	country, state, department, canton or other administrative division of which the municipality is the governmental seat
P1411	nominated for	award nomination received by a person, organisation or creative work (inspired from "award received" (Property:P166))
P1441	present in work	this (fictional or fictionalized) entity or person appears in that work as part of the narration
P1535	used by	item or concept that makes use of the subject (use sub-properties when appropriate)
P1923	participating team	like 'Participant' (P710) but for teams. For an event like a cycle race or a football match you can use this property to list the teams
P3450	sports season of league or competition	property that shows the competition of which the item is a season. Use P5138 for "season of club or team" .
P3602	candidacy in election	election where the subject is a candidate
P3701	incarnation of	incarnation of another religious or supernatural being
P5800	narrative role	narrative role of this character (should be used as a qualifier with P674 or restricted to a certain work using P642)
P6087	coach of sports team	sports club or team for which this person is or was on-field manager or coach

Table 9: List of relation labels in HyperRED.

Wiki ID	Label	Description
P17	country	sovereign state of this item (not to be used for human beings)
P25	mother	female parent of the subject. For stepmother, use "stepparent" (P3448)
P31	instance of	that class of which this subject is a particular example and member
P39	position held	subject currently or formerly holds the object position or public office
P81	connecting line	railway line(s) subject is directly connected to
P102	member of political party	the political party of which a person is or has been a member or otherwise affiliated
P131	located in the administrative territorial entity	the item is located on the territory of the following administrative entity.
P155	follows	immediately prior item in a series of which the subject is a part, preferably use as qualifier of P179
P175	performer	actor, musician, band or other performer associated with this role or musical work
P197	adjacent station	the stations next to this station, sharing the same line(s)
P249	ticker symbol	identifier for a publicly traded share of a particular stock on a particular stock market or that of a cryptocurrency
P276	location	location of the object, structure or event. In the case of an administrative entity as containing item use P131.
P413	position played on team / speciality	position or specialism of a player on a team
P453	character role	specific role played or filled by subject – use only as qualifier of "cast member" (P161), "voice actor" (P725)
P512	academic degree	academic degree that the person holds
P518	applies to part	part, aspect, or form of the item to which the claim applies
P527	has part	part of this subject; inverse property of "part of" (P361). See also "has parts of the class" (P2670).
P577	publication date	date or point in time when a work was first published or released
P580	start time	time an event starts, an item begins to exist, or a statement becomes valid
P582	end time	time an item ceases to exist or a statement stops being valid
P585	point in time	time and date something took place, existed or a statement was true
P642	of	qualifier stating that a statement applies within the scope of a particular item
P670	street number	number in the street address. To be used as a qualifier of Property:P669 "located on street"
P708	diocese	administrative division of the church to which the element belongs
P768	electoral district	electoral district this person is representing, or of the office that is being contested.
P805	statement is subject of	(qualifying) item that describes the relation identified in this statement
P812	academic major	major someone studied at college/university
P1114	quantity	number of instances of this subject
P1129	national team appearances	total number of games officially played by a sportsman for national team
P1310	statement disputed by	entity that disputes a given statement
P1346	winner	winner of a competition or similar event, not to be used for awards
P1350	number of matches played/races/starts	matches or games a player or a team played during an event.
P1352	ranking	subject's numbered position within a competition or group of performers
P1365	replaces	person, state or item replaced. Use "structure replaces" (P1398) for structures.
P1416	affiliation	organization that a person or organization is affiliated with (not necessarily member of or employed by)
P1545	series ordinal	position of an item in its parent series (most frequently a 1-based index), generally to be used as a qualifier
P1686	for work	qualifier of award received (P166) to specify the work that an award was given to the creator for
P1706	together with	qualifier to specify the item that this property is shared with
P2453	nominee	qualifier used with «nominated for» to specify which person or organization was nominated
P2868	subject has role	role/generic identity of the item ("subject"), also in the context of a statement.
P3831	object has role	(qualifier) role or generic identity of the value of a statement ("object") in the context of that statement
P3983	sports league level	the level of the sport league in the sport league system
P5051	towards	qualifier for "adjacent station" (P197) to indicate the terminal station(s) of a transportation line or service in that direction

Table 10: List of qualifier labels in HyperRED.