HacRED: A Large-Scale Relation Extraction Dataset Toward Hard Cases in Practical Applications

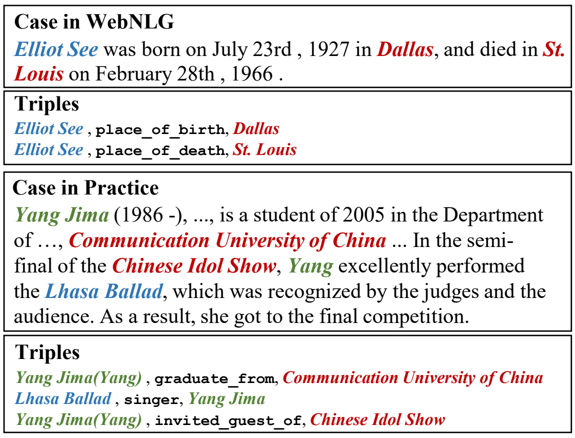
Qiao Cheng[1*], **Juntao Liu**[1*], **Xiaoye Qu**[2], **Jin Zhao**[1], **Jiaqing Liang**[1],
Zhefeng Wang[2], **Baoxing Huai**[2], **Nicholas Jing Yuan**[2], **Yanghua Xiao**[1,3†]
[1]Shanghai Key Laboratory of Data Science,
School of Computer Science, Fudan University, China
[2]Huawei Cloud, China
[3]Fudan-Aishu Cognitive Intelligence Joint Research Center, Shanghai, China
{cq.qiaojim, l.j.q.light}@gmail.com, {jtliu19, shawyh, jinzhao20}@fudan.edu.cn,
{quxiaoye, wangzhefeng, huaibaoxing, nicholas.yuan}@huawei.com

Abstract

Relation extraction (RE) is an essential topic in natural language processing and has attracted extensive attention. Current RE approaches achieve fantastic results on common datasets, while they still struggle on practical applications. In this paper, we analyze the above performance gap, the underlying reason of which is that practical applications intrinsically have more hard cases. To make RE models more robust on such practical hard cases, we propose a case-oriented construction framework to build a **H**ard **C**ase **R**elation **E**xtraction **D**ataset (**HacRED**). The proposed HacRED consists of 65,225 relational facts annotated from 9,231 documents with sufficient and diverse hard cases. Notably, HacRED is one of the largest Chinese document-level RE datasets and achieves a high 96% F1 score on data quality. Furthermore, we apply the state-of-the-art RE models on this dataset and conduct a thorough evaluation. The results show that the performance of these models is far lower than humans, and RE applying on practical hard cases still requires further efforts. HacRED is publicly available at `https://github.com/qiaojiim/HacRED`.

1 Introduction

Relation extraction (RE) is one of the core NLP tasks and plays an increasingly important role in knowledge graph completion (Bordes et al., 2013) and question answering (Dong et al., 2015). RE aims to extract structured relational facts, i.e., triples such as (*Bill Gates*, `founder_of`, *Microsoft*) from plain texts. Recently, various models (Zeng et al., 2018; Takanobu et al., 2019; Fu et al., 2019; Wei et al., 2020) have been proposed to identify the relational facts and achieved state-of-the-art (SOTA) performance, among which the latest method CasRel achieves notable 91.8% F1 score on WebNLG (Gardent et al., 2017) and 89.6% on NYT (Riedel et al., 2010).



Case in WebNLG
Elliot See was born on July 23rd , 1927 in ***Dallas***, and died in ***St. Louis*** on February 28th , 1966 .

Triples
Elliot See , `place_of_birth`, ***Dallas***
Elliot See , `place_of_death`, ***St. Louis***

Case in Practice
Yang Jima (1986 -), ..., is a student of 2005 in the Department of ..., ***Communication University of China*** ... In the semi-final of the ***Chinese Idol Show***, ***Yang*** excellently performed the ***Lhasa Ballad***, which was recognized by the judges and the audience. As a result, she got to the final competition.

Triples
Yang Jima(Yang) , `graduate_from`, ***Communication University of China***
Lhasa Ballad , `singer`, *Yang Jima*
Yang Jima(Yang) , `invited_guest_of`, ***Chinese Idol Show***

Figure 1: Cases and corresponding triples in WebNLG and practical applications.

However, can these seemingly fantastic results prove that the current RE models are powerful enough to perform well in practical applications. To answer the question, we employ CasRel on 300 randomly selected samples of WebNLG and the same number of data from practical DuIE[1]. The F1 scores under these scenarios drop significantly from 89.3% to 62.8%. As illustrated in Figure 1, CasRel extracts correct triples (*Elliot See*, `place_of_birth`, *Dallas*) and (*Elliot See*, `place_of_death`, *St. Louis*) in WebNLG where keywords such as *born* and *died* explicitly express the relation information. In contrast, CasRel fails to extract triples such as (*Yang Jima*, `graduate_from`, *Communication University of China*) where no keywords like *graduate* are mentioned. The most significant reason why CasRel performs well on WebNLG but struggles on practical data is that more challenging instances which

[*] Equally contributed.
[†] Corresponding author.

[1] http://lic2019.ccf.org.cn/kg

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2819–2831
August 1–6, 2021. ©2021 Association for Computational Linguistics

we refer to as hard cases exist in the practical applications. Moreover, according to the statistics of entity description documents in CN-DBpedia (Xu et al., 2017), at least 40.1% relational facts can only be extracted from hard cases. Therefore, relation extraction from hard cases can not be neglected and demands more attention.

Although many datasets (Li et al., 2016; Yao et al., 2019) have been proposed for RE, they rarely analyze the performance gap and focus on the hard cases. In order to make models robust on hard cases and more fit practical scenarios, in this paper, we aim to build a RE dataset with sufficient hard cases. To this end, we propose a case-oriented construction framework based on the challenging instances and build a **Ha**rd **C**ase **R**elation **E**xtraction **D**ataset (**HacRED**). Specifically, we first obtain general, massive, and various contexts as well as relational facts from CN-DBpedia to construct a distantly supervised dataset. The crucial part is to distinguish hard cases from abundant data. Therefore, we formulate nine indicators through systematic analysis of hard cases to quantify them. Then, we conduct feature engineering based on the valid indicators. Afterwards, a classifier is trained for distinguishing the desired hard cases. Finally, we develop a crowdsourcing platform with a novel three-stage annotation strategy and effective aggregation method CrowdTruth2.0 (Dumitrache et al., 2018) to guarantee the data size and quality.

In total, HacRED consists of 9,231 instances with 26 predefined relations and 9 types of entities. To the best of our knowledge, it is one of the largest document-level RE benchmark. Moreover, HacRED contains sufficient and diverse hard cases in line with practice. We conduct extensive experiments and systematic error analysis of SOTA models on HacRED. A sharp performance drop on HacRED compared to the existing benchmarks proves that RE in practical applications remains an open problem and still requires further research.

To recap, our main contributions are three-fold:

- We first analyze the performance gap between popular datasets and practical applications, and therefore construct one of the largest Chinese document-level RE dataset which contains sufficient and diverse hard cases to improve the evaluation for complex RE tasks.

- We propose a case-oriented construction framework to build RE dataset toward special cases. Meanwhile, we design a novel three-stage annotation method applicable for crowdsourcing of complex RE.

- We systematically evaluate the current mainstream RE models on HacRED and justify its effectiveness in depth.

2 Related Work

2.1 Datasets for Relation Extraction

A series of datasets have been built for RE as of late, which have extraordinarily advanced the improvement of RE systems. RE datasets such as SemEval-2010 Task 8 (Hendrickx et al., 2009) and ACE05 are constructed through human annotation with relatively limited relation types and size. A large-scale dataset TACRED (Zhang et al., 2017) is obtained via crowdsourcing to satisfy the training of data-hungry models.

As RE applications differ much in various scenarios, constructing datasets aimed at specific targets is a popular trend in RE. DocRED (Yao et al., 2019) is constructed to accelerate the research on document-level RE. To meet the challenges of few-shot RE, FewRel (Han et al., 2018) as well as FewRel 2.0 (Gao et al., 2019) have been presented. RELX (Koksal and Ozgur, 2020) is a benchmark for cross-lingual RE. Jia et al. (2020) propose the task of interpersonal RE in dyadic dialogues and further construct a corresponding dataset called DDRel.

Compared with previous RE datasets, HacRED is derived from the analysis of the performance gap between popular datasets and practical applications. It targets towards promoting the RE models to extract information from the complex contexts.

2.2 Models for Relation Extraction

Recently, many exciting works have been proposed to solve the RE tasks. **(1)Joint Model:** NovelTagging (Zheng et al., 2017) first formulates the task as a sequence labeling problem and presents a novel tagging schema to jointly extract entities and relations. CopyRE (Zeng et al., 2018) extracts triples based on a sequence-to-sequence structure and integrates the copy mechanism for entity generation. GraphRel (Fu et al., 2019) uses graph convolutional network (GCN) to capture features of words and text. CasRel (Wei et al., 2020) is different from the past and is able to extract more triples by learning relation-specific entity taggers. **(2)Pipeline Model:** PURE

(Zhong and Chen, 2020) is a simple pipelined approach which learns an entity model and a relation model independently. DGCNN-BERT is a powerful pipeline method that first identifies multiple relations and then labels the head and tail entities given a relation. It achieves 89.3 F1 scores and has won the champion in the Competition of DuIE held by Baidu Inc. **(3)Document-level Relation Classification Models:** LSR (Nan et al., 2020) is a model that empowers the relational reasoning across sentences by automatically inducing the latent document-level graph. GAIN (Zeng et al., 2020) introduces a path reasoning mechanism based on a heterogeneous mention-level graph and an entity-level graph. ATLOP (Zhou et al., 2020) proposes two techniques, adaptive thresholding and localized context pooling. SSAN (Xu et al., 2021) designs several transformations to incorporate mention structural dependencies for document-level relation classification (DocRC).

3 Easy Cases vs. Hard Cases

To analyze where models struggle in practical instances and distinguish the hard cases, we conduct a manual exploratory analysis on the error-prone instances of SOTA models (CGCN, CasRel, DGCNN-BERT) on NYT, DuIE and industry data. Then we formulate the potential causes of the errors with nine indicators illustrated as follows:

Text Length. We notice that models tend to fail on instances with longer text. The experiments of Alt et al. (2020) also reflect that RE models get a relatively higher error rate with the length of sentence greater than 30 in TACRED.

Argument Distance. We observe that the performance of the models declines when the arguments (i.e., head and tail entity mentions) are far away, especially in inter-sentence RE.

Distractors. Extracting triples in contexts with linguistic distractors is tough for current models. For example, *drop out* will contribute to wrong relation `graduate_from` between entity mentions with `PERSON` and `SCHOOL` type.

Reasoning. Reasoning is needed to extract the relation mentioned implicitly in the text. Recent work suggests that future researchers consider incorporating common sense knowledge or improved causal modules in RE tasks (Han et al., 2018).

Homogeneous Entities. The context contains multiple homogeneous entity mentions with identical types. We observe the high error rate in relations like `children` and `parents` when the text mentions different entities with type `PERSON`.

Text 1: "..." said *Joseph Bastianich*, who owns Del Posto with his mother, *Lidia Bastianich*, and the chef, *Mario Batali*. **Annotation**: `NA` **Prediction**: `children_of` **Indicators**: Distractor, Homogeneous Entities **Interpretation**: Three entity mentions with the same type of `PERSON` are mentioned in the text and the word *mother* may lead to wrong prediction `children`.
Text 2: ... Lieberman, who was defeated by the political upstart Ned *Lamont* in *Connecticut*'s Democratic primary earlier this month. **Annotation**: `place_lived` **Prediction**: `place_of_birth` **Indicators**: Similar Relations **Interpretation**: The relation `place_lived` and `place_of_birth` are similar in semantics.
Text 3: One of the most brutal tyrants of recent history, Saddam *Hussein* unleashed devastating regional wars and reduced oil-rich *Iraq* to a claustrophobic police state. **Annotation**: `nationality` **Prediction**: `place_of_death` **Indicators**: Reasoning **Interpretation**: Reasoning is required to get the relation `nationality` based on the context that *Hussein* is the *tyrants* of *Iraq*.

Table 1: Examples of hard cases in NYT. The *head* and *tail* mentions are colored accordingly.

Similar Relations. Models struggle to identify the correct relation among those semantically similar ones concurrently mentioned in context. A sharp decrease is also found in few-shot RE when selecting N similar relations on N-way K-shot settings (Han et al., 2020).

Long-tail Relations. Only a handful instances are available for long-tail relations in common datasets. Current data-hungry models struggle to learn the semantic patterns on these relations.

Multiple Triples. Models always get a poor performance on the instances with numerous triples.

Overlapping Triples. Different triples involve the identical entity mentions. Many existing models can not well handle the *EntityPairOverlap* and *SingleEntityOverlap* (Zeng et al., 2018) instances.

Table 1 provides various examples from NYT and corresponding hard case indicators. In Table 2, the proportion growing on the error instances reflects the gap between existing datasets and practical data, which also proves the effectiveness of these indicators.

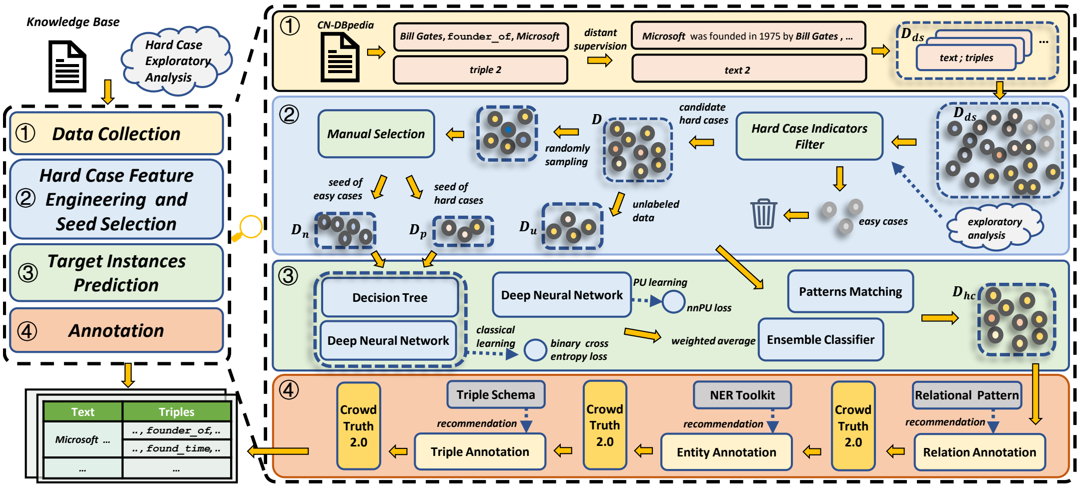


Figure 2: The case-oriented construction framework of building HacRED which consists of four stages. The right part correspondingly describes each stage. Through the construction, the texts and triples are established.

Indicator	WebNLG		DuIE	
	original	error	original	error
Text Length	18	39	3	32
Argument Distance	12	30	5	17
Distractors	1	5	4	13
Reasoning	-	3	1	9
Homogeneous Ent.	2	34	19	21
Similar Rel.	9	54	27	17
Long-tail Rel.	1	5	-	2
Multiple Triples	17	59	8	93
Overlapping Triples	25	64	16	33

Table 2: The proportion of indicators in randomly selected samples of original test set and error-prone instances. Note that one case may fit multiple indicators.

4 HacRED Dataset Construction

The overall architecture of the proposed case-oriented construction framework is illustrated in Figure 2. Different from previous works (Zhang et al., 2017, Zaporojets et al., 2020) which start crowdsourcing annotation straight after the data collection stage, we introduce additional stages of hard case feature engineering and target instance prediction. Moreover, we design a novel three-stage annotation method and employ CrowdTruth2.0.

4.1 Data Collection

To avoid data bias to high-frequency entities and relations, we first obtain about 5 million plain texts and 800 thousand triples from CN-DBpedia. The abundant texts and triples contribute to a more reasonable distribution. We use fine-grained named entity recognition (NER) toolkit TexSmart (Zhang et al., 2020) and entity linking (Chen et al., 2018) to align mentioned entities in texts to those in triples. Finally, we construct a distantly supervised dataset D_{ds} with 1.6 million instances, where we select challenging instances in the following steps.

4.2 Hard Case Feature Engineering and Seed Selection

To build a dataset toward practical hard cases, we systematically formulate the nine indicators of hard cases (refer to Section 3) and introduce measurements to quantify them. For example, we calculate the *Argument Distance* as the number of tokens between the head and tail entity mentions in the text. More details of feature engineering are described in Appendix A. After hard case oriented feature engineering, we discard the instances in D_{ds} without any indicator of hard cases. The remaining part forms a hard case candidate dataset D with about 108 thousand instances.

We randomly sample 3,500 instances from D and ask experts to select the hard cases given the context and features. Specifically, if an instance with multiple hard case indicators or with only one indicator but selected by all three experts based on their expertise, it is regarded as a hard case. To further evaluate the quality of selected hard cases, we utilize DGCNN-BERT to test the selected and unselected data. If the F1 score drops δ=10% on the hard cases, we reserve the data to constitute the high quality seeds of hard case D_p. The remaining data is easy case D_n. In total, we obtain 1,431 seeds of hard cases.

4.3 Classifier Training and Hard Case Prediction

It is impossible to manually select all instances to construct a large-scale dataset. So we utilize a classifier to recall more hard cases similar to the seed samples selected by experts. The classifiers consist of three categories: (1) Decision tree (Quinlan, 1986); (2) Deep classifiers by positive negative (PN) learning (Rakhlin, 2016); (3) Deep classifiers by positive unlabeled (PU) learning (Kiryo et al., 2017; du Plessis et al., 2015). First of all, we adopt the decision tree to make the classifier aware of the indicators explicitly. Then, we form the representation vector as recommended in Baldini Soares et al. (2019) and utilize classical PN learning on D_p and D_n to train the basic classifiers. Since the easy cases are extremely diverse and D_n can not represent the entire distribution of easy cases, we leverage the massive unlabeled data in D_{ds} by introducing PU learning to improve the generalization of hard cases classification. Besides, we train deep models based on different neural network structures, including CNN (LeCun et al., 1998) and BiLSTM (Hochreiter and Schmidhuber, 1997), to capture the context information. More training details can be found in Appendix B.

We ensemble multiple classifiers by weighted average and distinguish hard cases with high confidence in the original massive unlabeled dataset. Besides, we directly select instances by implicit semantic patterns to explore more hard cases fitting the indicator of *Reasoning* which is not well quantified by the auxiliary features. Finally, we obtain the dataset D_{hc} ready for annotation.

4.4 Crowdsourcing

To make instances in D_{hc} fully and accurately labeled, we develop a novel three-stage RE annotation platform taking the following two aspects into consideration: (1) Heavy workload of annotating all information at once results in growing negative feedback as the task goes on; (2) Aggregated method, such as majority vote (Dumitrache et al., 2018), is insufficient for complicated and open-ended tasks. To relieve the pressure of workers, we divide the whole task into three partitions consisting of *Relation Annotation*, *Entity Annotation*, and *Triple Annotation*. Moreover, we utilize patterns and toolkits to provide high-quality recommendations in each stage for higher recall. To capture the label disagreement more thoroughly among workers, we employ CrowdTruth2.0 (Dumitrache et al., 2018), which models the quality of workers, documents, and annotations.

In short, in the *Relation Annotation*, workers select the missed relations or delete wrong recommended ones. When all relations are annotated, NER toolkit recommends multiple entity mentions with the corresponding type based on schema information. Workers also need to append new entity mentions or delete incorrect ones in the *Entity Annotation*. As for *Triple Annotation*, workers verify the correctness of a candidate triples automatically generated by permutation of entity arguments and relations based on schema. Note that every input data in the three stage is assigned to three different annotators and aggregated by CrowdTruth2.0. Detailed annotation process is in Appendix D.

5 Experiments

In this section, we first compare our HacRED with existing datasets. Then we re-evaluate the SOTA RE models on HacRED and systematically analyze their abilities on different experiment settings. At last, we demonstrate the effectiveness of HacRED via a case study.

5.1 Data Analysis

In this section, we analyze various aspects of common RE datasets and HacRED.

Data Size. As shown in Table 3, HacRED has a greater average number of words, entities, and triples in each text than all of the sentence-level datasets. Thus we regard HacRED as a document-level RE dataset. Compared with the document-level datasets, DocRED aims at common document-level RE but not consider performance gaps and various hard cases in practical scenarios. BC5CDR is specially designed for biomedical domain. By contrast, we are the first to analyze the performance gap between popular datasets and practical applications, and propose HacRED which focuses on different kinds of hard cases in general domain. Besides, HacRED is larger in scale and contains much more various relational facts than BC5CDR and DocRED but with lower duplicated triples ratio.

Data Distribution. We calculate three global statistic metrics about data distribution of common datasets and HacRED. Table 4 show the results. Specifically, 84.29% of the triples in NYT and 91.20% in WebNLG are duplicate, which results in a bias to high-frequency triples of same entity pairs (known as *semantic bias* for models). For example, (*Beijing*, `capital_of`, *China*) occurs frequently in corpus and models still extract this triple from *Beijing is a historic city in China*. Mean-

Dataset	# Text	# Relation	# Triple	# Fact	Avg. Sent.	Avg. Word‡	Avg. Ent.	Avg. Triple
sentence-level dataset								
SemEval10	13,434	10	13,434	10,251	1.0	17.4	2.0	1.0
NYT	66,194	24	104,339	16,387	2.1	37.8	2.2	1.6
WebNLG	6,222	171	14,485	1,275	2.5	24.0	3.15	2.3
TACRED	106,264	41	21,773	5,976	1.0	33.2	2.0	1.0
document-level dataset								
BC5CDR	1,500	1	3,116	2,434	7.4	188.0	19.5	2.1
DocRED	5,053	96	63,427	56,354	8.0	198.3	26.2	12.5
HacRED	**9,231**	26	**67,047**	**65,225**	5.0	126.6	10.8	7.4

Table 3: Statistics of common RE datasets and HacRED. Note that the *Avg.Word* is computed at word-level vocabulary, which means “中国” (China), two characters in Chinese, is regarded as one word. The average length of documents at character-level is 204.2 in HacRED.

Dataset	Duplicated Triples	Biased Relations	Top 20% Relation Triples
sentence-level dataset			
SemEval10	23.69%	0.00%	44.92%
NYT	84.29%	58.33%	98.93%
WebNLG	91.20%	94.74%	77.57%
TACRED	72.55%	9.52%	91.33%
document-level dataset			
BC5CDR	21.89%	-	-
DocRED	11.15%	12.50%	71.46%
HacRED	**2.72%**	**0.00%**	49.96%

Table 4: Data distributions of common RE datasets and HacRED. The ratio of duplicate triples, biased relations, and top 20% relation triples is calculated as $1-\frac{\#Facts}{\#Triples}$, $\frac{\#Biased\ Rel}{\#Rel}$, $\frac{\#Triples\ of\ top20\%\ Rel}{\#Triples}$, respectively. If the highest-frequency mention is involved in more than 10% triples of the given relation, we regard it as a biased relation.

Dataset	Relation Example	Highest-frequency Mention (Ratio)
WebNLG	`county_seat`	*Texas* (72.73%)
NYT	`person_profession`	*Bavetta* (50.00%)
DocRED	`sister_city`	*Chipilo* (35.29%)
HacRED	`dynasty`	*Tang* (4.20%)

Table 5: Example of relations which could lead to *selection bias* in WebNLG, NYT, and DocRED. In HacRED, the ratio of the highest-frequency mention in all relations is only 4.20%.

Indicators		Ratio
Text Length & Argument Distance		25.40%
Distractors & Reasoning		21.20%
Homo. Entities & Similar Relations		9.67%
Long-tail Relations		13.66%
Multiple Triples	1-3	38.87%
	4-9	36.67%
	10-15	14.27%
	16+	10.20%
Overlapping Triples		13.20%

Table 6: Statistics about the proportion of instances fitting different hard case indicators on HacRED.

CrowdTruth 2.0		Human (%)	
Avg. UQS ↑	0.9373	Precision	97.29
Avg. AQS ↑	0.9446	Recall	94.64
Avg. WQS ↑	0.9557	F1	95.94

Table 7: Results of different quality metrics on HacRED.

while, the top 20% relations in NYT nearly cover the entire relation triples. The numbers of top and last 20% relation triples in WebNLG, TACRED and DocRED also vary greatly. As a result, models perform well on popular relations but fail on long-tail ones. The experiments in the Section 5.4 prove this and we regard it as *relation bias*. In addition, 94.74% relations in WebNLG, 58.33% in NYT and 12.5% in DocRED contribute to the *selection bias*. In WebNLG, 72.73% triples with relation `county_seat` involve the mention *Texas*, as illustrated in Table 5. Models could memorize the cooccurrence between high-frequency mentions and the relation while low-frequency mentions are neglected. All these three aspects reveal the unreasonable data distribution of common datasets.

In comparison, we observe a more reasonable data distribution in HacRED from Table 4 and Table 5. HacRED has a low ratio of duplicate triples and contains various relational facts, which addresses *semantic bias*. No biased relation existing in HacRED reduces the risk of *selection bias*. The proportion of top 20% relations promotes the alleviation of *relation bias* on HacRED. The more comparison of overall data distribution can be found in Appendix E.

Data Quality. We evaluate the quality of HacRED through both automatic metrics and human evaluation. Specifically, we first compute the average *unit quality score* (UQS), *annotation quality score* (AQS), and *worker quality score* (WQS) of the whole 9,231 instances. *UQS*, *AQS* and *WQS* are proposed by CrowdTruth2.0 (Appendix F provides more calculation details). The closer these

Model	Precision	Recall	F1
Joint	**NER‡**		
NovelTagging	46.77	35.07	40.08
CopyRE	75.04	51.38	61.00
GraphRel	**85.14**	**69.69**	**76.64**
CasRel	75.43	62.88	68.59
	End-to-end		
NovelTagging	30.51	2.91	5.31
CopyRE	13.11	9.64	11.12
GraphRel	30.13	35.62	32.65
CasRel	**55.24**	**43.78**	**48.85**
Pipeline	**NER‡**		
PURE	72.23	63.45	67.56
	End-to-end		
PURE	55.14	66.09	60.12
Doc. Level	**Relation Classification**		
LSR	69.70	67.17	68.41
GAIN	72.04	**80.62**	76.09
ATLOP	**77.89**	76.55	**77.21**
SSAN	60.01	62.03	61.00

Table 8: Model performance on HacRED test set(%). NER results are computed based on the entities involved in the gold triples of each instance.

scores are to 1, the higher quality of the crowdsourcing is. Meanwhile, we randomly sample 400 instances from HacRED and compute the precision, recall, and F1 score with annotations based on the revision of humans. The evaluation scores are reported in Table 7. From this table, our HacRED achieves a considerable annotation quality. As a comparison, NYT contains about 31% noise instances (Riedel et al., 2010) and TACRED has poor annotation quality (Alt et al., 2020).

Hard Case Types. We group the randomly sampled 400 instances into nine categories as shown in Table 6. The proportions of different kinds of instances reflect that HacRED contains a various range of hard cases, which evaluates models comprehensively for practical applications.

5.2 Model Evaluation

As DGCNN-BERT has been used in the main process of construction, we evaluate other strong RE models including joint RE models, pipeline RE models, and DocRC models on HacRED. First, we limit the relation set within 20 types both in HacRED and DuIE, and then separate a part of instances in DuIE to form the contrastive easy case dataset D_{ec}. We carry out the equivalent substitution of hard cases in HacRED for easy ones in D_{ec} in different proportions. Figure 3 shows the F1 curve of the performances w.r.t. the proportion of substitution. As the ratio of replacement increases, models generally have a growing trend in performance. The SOTA model CasRel still outperforms other joint models and achieves great F1 on 100% D_{ec}. However, the performance drops on data with more complex instances. We notice that F1 value of easy cases is generally greater than that of hard cases in different substitution ratio settings, which illustrates that RE models indeed struggle when tackling hard cases. Note that by combining HacRED with easy cases in existing datasets, it is easy to simulate diverse practical scenarios.

Model	Precision	Recall	F1
	End-to-end		
CasRel	58.76	45.43	51.24
PURE	56.52	65.15	60.53
Human	**90.21**	**84.59**	**87.31**
	Relation Classification		
ATLOP	78.33	76.70	77.51
Human	**96.21**	**93.03**	**94.59**

Table 9: Human performance (%).

In addition, we split HacRED into train, dev, and test sets with 6231, 1500, 1500 instances respectively. The precision, recall, and F1 score of the three major categories of models are shown in Table 8. The joint and pipeline learning strategies do not contribute to a great F1 on triple extraction. For the NER task, PURE has a separate entity model but results in a 30.61% F1 when all entities in a document are considered, including entities with no positive relation labels. This also reflects the challenge to obtain complete entity information in practical scenarios. On the other hand, the relation classification performances of DocRC models are far from satisfactory. The results suggest that existing models have remarkably poor performance on HacRED compared with humans (Table 9), which indicates that RE applicable for practical hard cases still requires further research.

5.3 Human Performance

We randomly select 200 contexts from test set and ask three volunteers to extract relational facts in an end-to-end manner. Schema information like entity type set as well as relation set is provided but no entity mentions. As for relation classification task, three volunteers select the relation, including NA regarded as negative, of the given entity pair. As demonstrated in Table 9, humans fulfill excellent results which indicate the possible ceiling performance on HacRED.

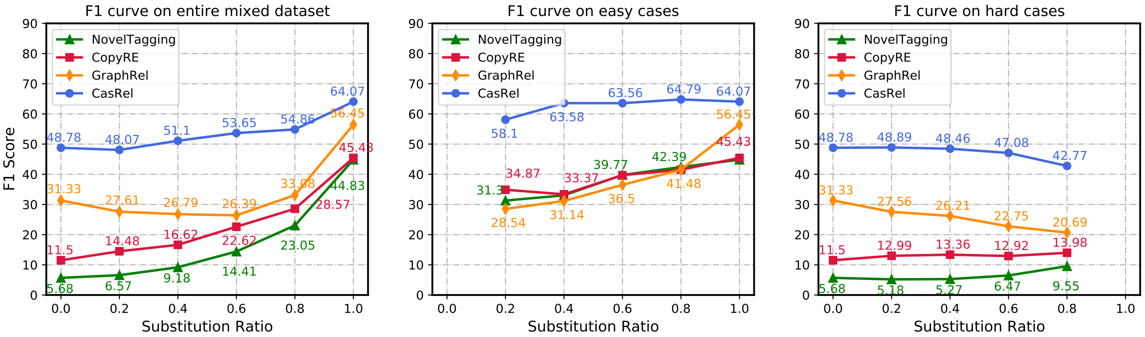


Figure 3: The F1 curve of the model performance on different mix ratios of hard and easy cases.

Model	Text Length Argument Distance	Homo. Ent. Similar Rel.	Long-tail Rel.	Overlapping Triples	Distractor Reasoning	Overall
NovelTagging	4.99	4.33	1.72	3.99	9.23	5.31
CopyRE	5.47	3.90	1.28	6.59	7.30	11.12
GraphRel	30.15	27.82	0.08	34.67	29.81	32.65
CasRel	45.34	45.60	13.54	53.34	44.00	48.85

Table 10: F1 score on HacRED instances with different indicators of hard cases (%).

5.4 Detailed Analysis

In this section, we give insight into the abilities of current mainstream joint models when tackling different kinds of hard cases and propose some research indications as well. As it is hard to obtain complete entity information in practical scenarios, we do not consider DocRC models in this section that entity information is provided as input.

Multiple Triples. Table 11 shows the F1 score of existing models when extracting from texts with different number of triples. The performance of NovelTagging and CopyRE decreases as the number of triples increases, which indicates that the novel tagging schema and multiple decoder mechanism are not able to address the challenge of *Multiple Triples*. Since GraphRel predicts relations for all word pairs and CasRel learns separate entity tagger for different relations, these two models alleviate this problem. An interesting point is that the performance of GraphRel and CasRel rises as the number of triples increases when the triples number is less than 16, indicating that these two models work well in texts with number of triples nearing the average. However, all models get F1 score below average when text mentions have more than 16 triples.

Text Length and Argument Distance. To assess the abilities of models in capturing the long-distance context, we provide the evaluation on instances with indicators of *Text Length* and *Argument Distance* in Table 10. The GCN-based models (i.e., GraphRel) outperforms the BiLSTM-based neural models like NovelTagging and CopyRE. The performance improvement on CasRel suggests the powerfulness of BERT encoder in the long-distance context.

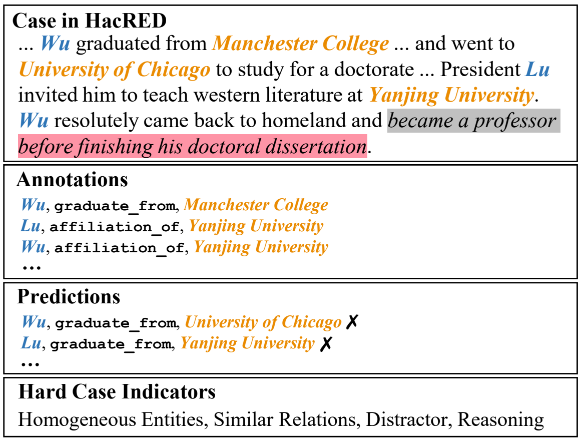
Model	Number of triples			
	1-3	4-9	10-15	16+
NovelTagging	17.92	12.18	8.60	3.29
CopyRE	12.69	10.58	8.82	3.38
GraphRel	29.49	35.23	37.04	29.24
CasRel	43.42	51.05	54.90	43.18

Table 11: F1 score on HacRED test set with different number of triples (%).

Homogeneous Entities and Similar Relations. Since the text mentions multiple homogeneous entities and semantically similar relations, models are required to distinguish the fine-grained difference of the context to extract the correct triples. The first two columns in Table 10 have similar results, which indicates that the contexts with homogeneous entities and similar relations are as challenging as the long-distance contexts.

Long-tail Relations. We observe a dramatic decrease on the instances with long-tail relational triples. As long-tail relations are common in real-world scenarios, a more efficient learning method is required to make RE models applicable for practical applications.

Overlapping Triples. CasRel achieves a better performance on extracting overlapping triples. This proves the effectiveness of cascade binary tagging strategy by first identifying the head mention and then extract the corresponding tail mention given a relation. Specifically, the F1 scores of



Case in HacRED

... *Wu* graduated from *Manchester College* ... and went to *University of Chicago* to study for a doctorate ... President *Lu* invited him to teach western literature at *Yanjing University*. *Wu* resolutely came back to homeland and *became a professor before finishing his doctoral dissertation*.

Annotations

Wu, `graduate_from`, *Manchester College*
Lu, `affiliation_of`, *Yanjing University*
Wu, `affiliation_of`, *Yanjing University*
...

Predictions

Wu, `graduate_from`, *University of Chicago* ✗
Lu, `graduate_from`, *Yanjing University* ✗
...

Hard Case Indicators

Homogeneous Entities, Similar Relations, Distractor, Reasoning

Figure 4: An example of hard cases in HacRED with multiple indicators.

overlapping head and tail mentions are 66.38% and 47.44% respectively. Similarly, results of the two above metrics in CopyRE are 13.31% and 3.57%. The relative higher performance on overlapping head mentions than tail mentions also suggests that the order of extracting arguments could have effect on the results.

Distractor and Reasoning. We manually select instances with *Distractor* and *Reasoning* indicators in HacRED because they cooccur frequently in corpus. As illustrated in Table 10, we observe a drop of the F1. This suggests that models are vulnerable to this kind of instances. However, there are lots of texts with distractions or implicit expression, which needs reasoning, and even common sense. The model design should take the reasoning mechanism into consideration in the future work.

5.5 Case Study

As shown in Figure 4, the text mentions multiple organization entities and similar relations including `graduate_from` and `affiliation_of`. The incorrect triple (*Lu*, `graduate_from`, *Yanjing University*) extracted by CasRel represents that models struggle to capture fine-grained semantic information. The distractive phrases *study for a doctorate* could result in the incorrect extraction (*Wu*, `graduate_from`, *University of Chicago*), which can be rectified by comprehending the context of *before finishing his doctoral dissertation*. Reasoning is needed to extract the triple (*Wu*, `affiliation_of`, *Yanjing University*) since he worked as a professor in the organization.

6 Conclusion

In order to effectively evaluate the RE models and accelerate the research of practical RE, we first analyze the performance gap between popular datasets and practical applications. Therefore, we construct a large-scale and high-quality HacRED with reasonable data distribution and sufficient hard cases. To focus on the practical challenging cases, we propose a case-oriented construction framework. We also design a novel annotation method to guarantee the quality of HacRED. Finally, we conduct extensive experiments and analyze the abilities of SOTA models from various aspects, which provides a deeper understanding of RE models and inspiration for further improvement.

Acknowledgements

We would like to thank the anonymous reviewers for their thoughtful and constructive comments. This research is supported by National Key Research and Development Project (No. 2020AAA0109302), Shanghai Science and Technology Innovation Action Plan (No.19511120400), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103), and Language and Speech Innovation Lab of Huawei Cloud.

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online. Association for Computational Linguistics.

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.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, J. Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*.

Lihan Chen, Jiaqing Liang, Chenhao Xie, and Y. Xiao. 2018. Short text entity linking with fine-grained topics. *Proceedings of the 27th ACM International*

Conference on Information and Knowledge Management.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question answering over Freebase with multi-column convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 260–269, Beijing, China. Association for Computational Linguistics.

Anca Dumitrache, Oana Inel, Lora Aroyo, Benjamin Timmermans, and Chris Welty. 2018. Crowdtruth 2.0: quality metrics for crowdsourcing with disagreement. *arXiv preprint arXiv:1808.06080*.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, Florence, Italy. Association for Computational Linguistics.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. 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 6250–6255, Hong Kong, China. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 745–758, Suzhou, China. Association for Computational Linguistics.

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

Iris Hendrickx, S. Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, M. Pennacchiotti, Lorenza Romano, and S. Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *SemEval@ACL*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Qi Jia, Hongru Huang, and K. Q. Zhu. 2020. Ddrel: A new dataset for interpersonal relation classification in dyadic dialogues. *ArXiv*, abs/2012.02553.

Ryuichi Kiryo, Gang Niu, Marthinus Christoffel du Plessis, and Masashi Sugiyama. 2017. Positive-unlabeled learning with non-negative risk estimator. In *NIPS*.

Abdullatif Koksal and A. Ozgur. 2020. The relx dataset and matching the multilingual blanks for cross-lingual relation classification.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

J. Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, A. P. Davis, C. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database: The Journal of Biological Databases and Curation*, 2016.

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.

Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. 2015. Convex formulation for learning from positive and unlabeled data. In *ICML*.

J. Ross Quinlan. 1986. Induction of decision trees. *Machine learning*, 1(1):81–106.

A Rakhlin. 2016. Convolutional neural networks for sentence classification. *GitHub*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML/PKDD*.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. *ArXiv*, abs/1811.03925.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging framework for relational triple extraction. In *ACL*.

Benfeng Xu, Qiangwen Wang, Yajuan Lyu, Yong Zhu, and Zhendong Mao. 2021. Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. *ArXiv*, abs/2102.10249.

Bo Xu, Yong Xu, Jiaqing Liang, Chenhao Xie, Bin Liang, Wanyun Cui, and Y. Xiao. 2017. Cn-dbpedia: A never-ending chinese knowledge extraction system. In *IEA/AIE*.

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.

Klim Zaporojets, J. Deleu, Chris Develder, and Thomas Demeester. 2020. Dwie: an entity-centric dataset for multi-task document-level information extraction. *ArXiv*, abs/2009.12626.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640, Online. 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.

Haisong Zhang, Lemao Liu, Haiyun Jiang, Yangming Li, Enbo Zhao, Kun Xu, Linfeng Song, Suncong Zheng, Botong Zhou, Jianchen Zhu, Xiao Feng, Tao Chen, Tao Yang, Dong Yu, Feng Zhang, Zhanhui Kang, and Shuming Shi. 2020. Texsmart: A text understanding system for fine-grained ner and enhanced semantic analysis. *arXiv preprint arXiv:2012.15639*.

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.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. *ArXiv*, abs/1706.05075.

Zexuan Zhong and Danqi Chen. 2020. A frustratingly easy approach for joint entity and relation extraction. *ArXiv*, abs/2010.12812.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and J. Huang. 2020. Document-level relation extraction with adaptive thresholding and localized context pooling. *ArXiv*, abs/2010.11304.

A More Examples of Hard Cases in NYT

In Table 12, we provide additional error-prone examples in NYT that fit other indicators of practical hard cases including *Text Length*, *Argument Distance*, *Multiple Triples*, and *Overlapping Triples*. We have illustrated the instances with other indicators in Section 3.

Text 1: *Sixten Ehrling*, ..., and directed the conducting programs at the *Juilliard School* and ...
Annotation: `affiliation of`
Prediction: `major shareholder of`
Indicators: Text Length, Argument Distance
Interpretation: The text contains many words and the distance between head and tail mention is much far. There is no indicating phrases such as *work in* directly revealing the relation `affiliation of`.

Text 2: Though officials in Addis *Ababa* , *Ethiopia*'s capital , ...
Annotation: `administrative divisions, contains, capital`
Prediction: `capital of`
Indicators: Multiple Triples, Overlapping Triples
Interpretation: The text mentions multiple triples and entities such as *Ethiopia* are involved in different triples.

Table 12: Examples of hard cases in NYT. The *head* and *tail* mentions are colored accordingly.

B Details of Feature Engineering

We calculate the *Text Length* and *Argument Distance* as the number of tokens in the text and between the head and tail entity mentions. *Homogeneous Entities* are measured by the NER results of TexSmart and equal to number of entities with same NER tag. The measurements of *Distractors*, *Similar Relations* are based on pre-defined schemas and auxiliary information, part of which is shown in Table 13. *Multiple Triples* and *Overlapping Triples* are computed by the triples from DS. As reasoning can not be implicitly quantified, we suppose the deep neural models to capture the features of context.

C Details of Classifier Training

A decision tree is learned by the auxilliary features calculated in stage 2. For deep models, we concatenate multiple embeddings and auxilliary features to make up the input. We add special tokens to mark the border of each entity and generate the representation vector as recommended in Baldini Soares et al. (2019). We assign a label 1 to

Relation	Type of Arguments	Similar Relations	Explicit Phrase	Distractors
`graduate_from`	PERSON, ORG	`affiliation _of`, `founder_of`	graduate, receive a degree	drop out college, visit
`spouse`	PERSON, PERSON	`parent`, `children`	marry, wife, tie the knot with	ex-wife
`director`	PERSON, FILM / TV SERIES	`cast_member`, `scriptwriter_of`	directed	watch
`anchorperson_of`	PERSON, VARIETY SHOW	`invited_guest_of`	emcee, host	
...	...	...	...	...

Table 13: Examples of pre-defined schemas and simple auxiliary informations to measured the *indicator_distractor* and *similar_rels*. Experts define some implicit expressions such as *receive a degree* reveals the relation *graduated_from* and distractive phrases like *ex-wife* for *spouse*.

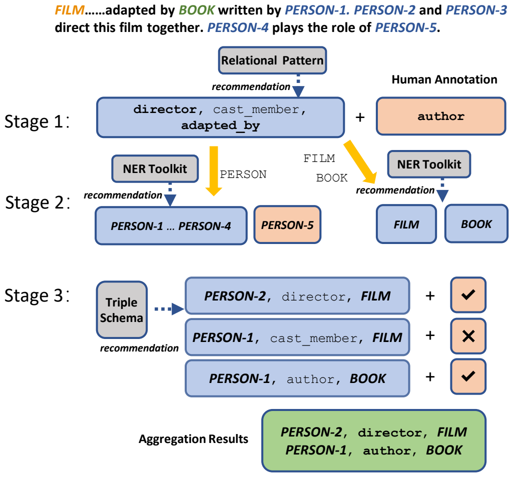


Figure 5: The illustration of three-stage annotation method.

each instance in D_p and -1 in D_n. The deep models output the probability of the instance belonging to hard cases and are optimized with the binary cross entropy loss objective. To start PU learning, we sample from D to form a unlabeled dataset D_u and set the hyperparameter $\pi_p = 0.41$ estimated by the proportion of hard cases selected by experts. We implement nnPU (Kiryo et al., 2017) which is efficient for massive data and deep learning and use J_{nnpu} as the optimized objective,

$$J_{nnpu} = \pi_p \cdot E_{p(x|y=1)}[l(g(x))] + max\{0, E_{p(x)}[l(-g(x))] - \pi_p \cdot E_{p(x|y=1)}[l(-g(x))]\} \quad (1)$$

where $\pi_p = p(y = 1)$, g is decision function, l is surrogate loss function. We choose the double hinge loss $l = max(-z, max(0, \frac{1}{2} - \frac{1}{2}z))$ proposed by (du Plessis et al., 2015).

D Three-stage Annotation Method

We illustrate the three-stage annotation method. Given the context in Figure 5, `director`, `cast_member`, and `adapted_by` is appended to the annotation of Stage 1 by relational pattern. Crowdsourcing workers select the missing relation such as `author`. When all relation mentions are annotated, NER toolkit recommend multiple entity mentions with the corresponding type. Workers need to select the highlighted words that are not covered by entity recommendation in the Stage 2. After stage 2, all mentions in context with specific type are obtained. As the example shown in Figure 5, given the target entity type of `PERSON`, platform recommends the candidates including *PERSON-1* to *PERSON-4*. Workers select highlighted words *PERSON-5* which is missed. In the final stage, we generate the candidate triples automatically by permutation of arguments and relations based on triple schema. Due to the relation `director` connects arguments with entity type `PERSON` and `FILM`, we generate the triple (*PERSON-2*, `director`, *FILM*) and ask annotator to verify the correctness. Note that we employ the powerful quality control method crowdtruth2.0 in every stages to prevent error propagation. As a result, all triples marked as valid are saved.

E Calculation of the UQS, AQS, and WQS Metrics in CrowdTruth2.0

We give the details of the calculation in data quality evaluation. We calculate the three metric unit quality score (UQS), annotation quality score (AQS), and worker quality score (WQS) by CrowdTruth2.0 (Dumitrache et al., 2018) on the whole 9,231 instances in HacRED proposed as follows, where W_1, W_2 is the weight of the iteration method and is initialized as one, u is the unit for annotation,a is one annotation given a unit, i, j denotes the different workers. We straightforward report the average of these metrics in Section 5.1.

$$UQS(u) = \frac{\sum_{i,j} W_1(i,j,u) WQS(i) WQS(j)}{\sum_{i,j} WQS(i) WQS(j)} \tag{2}$$

$$AQS(a) = \frac{\sum_{i,j} WQS(i) WQS(j) P_a(i|j)}{\sum_{i,j} WQS(i) WQS(j)} \tag{3}$$

$$\begin{aligned} WQS(i) &= WUA(i) WWA(i) \\ WUA(i) &= \frac{\sum_u W_2(u,i) UQS(u)}{\sum_u UQS(u)} \\ WWA(i) &= \frac{\sum_{j,u} W_1(i,j,u) WQS(j) UQS(u)}{\sum_{j,u} WQS(j) UQS(u)} \end{aligned} \tag{4}$$