Corpus-Based Task-Specific Relation Discovery

Karthik Venkat Ramanan
University of Illinois Urbana-Champaign
kv16@illinois.edu

Abstract

Relation extraction is a crucial language processing task for various downstream applications, including knowledge base completion, question answering, and summarization. Traditional relation-extraction techniques, however, rely on a predefined set of relations and model the extraction as a classification task. Consequently, such closed-world extraction methods are insufficient for inducing novel relations from a corpus. Unsupervised techniques like OpenIE, which extract `<head, relation, tail>` triples, generate relations that are too general for practical information extraction applications. In this work, we contribute the following: 1) We motivate and introduce a new task, corpus-based task-specific relation discovery. 2) We adapt existing data sources to create Wiki-Art, a novel dataset for task-specific relation discovery. 3) We develop a novel framework for relation discovery using zero-shot entity linking, prompting, and type-specific clustering. Our approach effectively connects unstructured text spans to their shared underlying relations, bridging the data-representation gap and significantly outperforming baselines on both quantitative and qualitative metrics. Our code and data are available in our GitHub repository.[1]

1 Introduction

Relation extraction (RE) aims to identify semantic relationships between entities in text in order to obtain triples of the form `<head, relation, tail>`, for instance, `<Washington D.C., capital_of, USA>`. RE is an important Information Extraction (IE) technique primarily used to complete knowledge bases (such as YAGO[2] and NELL[3]) and construct semantic graphs (Vashishth et al., 2018). Knowledge bases and semantic graphs see wide application in tasks such as question-answering (Saxena et al., 2020; Das et al., 2017), recommendation (Zhang et al., 2016), and natural language inference (Peters et al., 2019).

Traditional relation extraction techniques approach the problem as a multi-class classification problem, and hence assume a predefined set of relations. Open Information Extraction (OpenIE) approaches (Angeli et al., 2015; Mausam et al., 2012) seek to remedy this problem by extracting relations from text without a predefined schema. But recent work (Schneider et al., 2017) has shown that in the absence of a schema, OpenIE results tend to be uninformative or redundant. Moreover, OpenIE systems are tuned for high recall and hence extract a very general set of tuples and defer the problem of sifting through the generated triples to find meaningful ones to subsequent analysis.

We now introduce the problem of task-specific relation discovery. Discovering unseen relations from a corpus serves two functions. Firstly, it serves as a starting point to fine-tune relation extraction models on novel relations and unseen domains. Secondly, it is intuitively appealing as a data mining task to gather actionable insights from a large unstructured corpus. Say, for instance, we are presented with a collection of recent documents reporting on the COVID-19 pandemic. Discovering all the relationships between *viral variants* and *cities in Ohio* in the corpus would allow us to detect relations such as "declining in", "spreading in" and "endemic in". This would allow us to automatically identify if there are areas of concern where viral spread is increasing. We define a task as a pair of semantic types (such as Humans, Geographic Locations, Sports Teams, etc.) between which we are interested in relations. A few concrete examples of this formulation can be seen in figure 1.

In this work, we propose a novel solution to this problem in three steps. First, we identify candidate spans for relation discovery using zero-shot entity

[1]https://github.com/karthik63/relation-discovery
[2]https://yago-knowledge.org/
[3]http://rtw.ml.cmu.edu/rtw/kbbrowser/

Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023), pages 45–57
July 13, 2023 ©2023 Association for Computational Linguistics

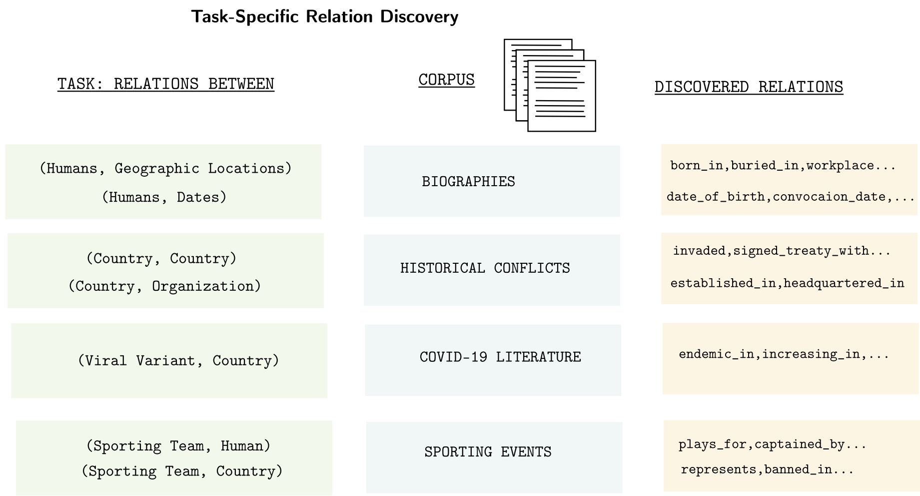

Figure 1: The formulation of corpus-based task-specific relation discovery.

linking and typing. Then we use a self-supervised prompting technique using an encoder-decoder transformer architecture (Lewis et al., 2019) to discover relation phrases that describe the relationship between the two entities. Finally, we cluster these discovered phrases while keeping the head and tail semantic types of our relations in mind.

We briefly summarize our contributions below

- We introduce a new task of corpus-based task-specific relation discovery and modify existing data sources to make available Wiki-Art, a new dataset for the same
- We propose a novel approach to extract candidate sentences, then discover and cluster unseen relations that significantly outperforms our baselines on both qualitative and quantitative metrics.

2 Related Work

The three lines of work most relevant to our approach are relation extraction, open information extraction, and prompting.

2.1 Relation Extraction

Most traditional relation extraction approaches model RE as a sequence classification task with specific accommodations for challenges arising from distant supervision. Models using piecewise CNNs (Zeng et al., 2015), reinforcement learning (Feng et al., 2018), and relationship side information have been proposed to mitigate the noise from sentences where ground-truth relations are not expressed.

2.2 Open Information Extraction

OpenIE seeks to produce domain-agnostic, unsupervised `<head, relation, tail>` extractions from a text span. Traditional approaches (Angeli et al., 2015; Mausam et al., 2012) to OpenIE use a combination of automatically mined and hand-crafted templates for relation extraction from the syntactic features and surface-forms of a sentence. These patterns are often mined using bootstrapping (Kolluru et al., 2020) where the triple extractions from multiple OpenIE approaches are aggregated to form a supervised training set using statistical rules.

2.3 Prompting

Prompting approaches (Liu et al., 2021) solve language processing tasks by eliciting natural language responses from language models rather than by training a classification layer. Relevant to our work, prompting has been successfully adapted to solve challenges such as few-shot event detection (Li et al., 2022) and event argument extraction (Ma et al., 2022; Li et al., 2021). Prompting can also be used to probe the inherent relational knowledge of pretrained language models by aggregating the masked language model generations from multiple hand-crafted prompts (Jiang et al., 2021).

3 Problem Definition

In this section, we formally define corpus-based task-specific relation discovery.

A corpus is a collection of documents from any domain. For simplicity, let our corpus $\mathcal{S} = [S_1, S_2, ..S_{|\mathcal{S}|}]$ be a sequence of sentences

S_i. A sentence $S_i = [w_{i,1}, w_{i,2}, ...w_{i,|S_i|}]$ is a sequence of words $w_i \in \mathcal{V}$. Our task $\mathcal{P} = \{(\mathcal{H}_1, \mathcal{T}_1), (\mathcal{H}_2, \mathcal{T}_2), ..(\mathcal{H}_{|\mathcal{P}|}, \mathcal{T}_{|\mathcal{P}|})\}$ is a set of $\mathcal{H}_i$ **head**, $\mathcal{T}_i$ **tail** semantic type tuples. $\mathcal{T} = \{(h_1, r_1, t_1), ...(h_{|\mathcal{T}|}, r_{|\mathcal{T}|}, t_{|\mathcal{T}|})\}$ represents the set of ground-truth head-relation-tail **triples** expressed in $\mathcal{S}$ such that $(h_i, t_i) \in \{(x_j, y_j)|(x_j, y_j) \in (\mathcal{H}_1, \mathcal{T}_1) \cup ... \cup (\mathcal{H}_{|\mathcal{P}|}, \mathcal{T}_{|\mathcal{P}|})\}$. That is, they only correspond to the task outlined using the head and tail semantic types.

Our end task is to discover the set of relations $\mathcal{R} = \{r_1, r_2, ...r_{|\mathcal{R}|}\}$ that occur in $\mathcal{T}$, without assuming any prior knowledge about the relations in $\mathcal{T}$ other than the head and tail semantic types.

4 Methodology

Our overall architecture is illustrated in figure 2. Our procedure for relation discovery comprises three steps that we detail below- 1. We identify relevant entities and extract candidate spans of sentences to perform discovery on. 2. We discover the relation phrases that explain the relation between the head and tail entities. 3. We cluster the extracted relation phrases.

4.1 Extracting Candidate Spans for Discovery

Extracting candidate spans for relation discovery requires that we identify when semantic types of interest, outlined by our task $\mathcal{P}$, co-occur in a paragraph. We require this entity typing process to have both high precision in order to avoid incorrect relation discoveries and high recall, so we don't miss infrequent relations in our corpus. We propose the following procedure (illustrated in Figure 3) that meets both of these requirements. We utilize the BLINK(Ott et al., 2019) framework for both named entity recognition and zero-shot entity linking to Wikipedia. The advantage of using a zero-shot entity linker is that we can swap the Wikipedia index out for a more recent one, if needed, in order to handle newer entities. We then make use of the Wikipedia API[4] in order to identify the Wikidata[5] ID of the linked entity. With an entity's Wikidata ID we first identify its type using the P31 *"instance of"* edge in the Wikidata KB. We then establish if it belongs to one of our pre-specified semantic classes by traversing the Wikidata taxonomy through the P279 *"subclass of"* edge until we reach the root node. If the concept node corresponding to one of our semantic classes (e.g. human, country, film) is an ancestor of the type node, we identify the entity as such. If entities from both the head and tail semantic classes co-occur in a paragraph, it is selected as a candidate span to perform relation discovery on.

[4]https://en.wikipedia.org/w/api.php
[5]https://www.wikidata.org/

Property Name	Property Definition	Property Aliases
discoverer or inventor (P61)	subject who discovered, first described, invented, or developed this discovery or invention	inventor, discoverer, inventor or discoverer, developer, coined, first described, invented by, created by, invented, discovered by, developed by, introduced by, devised by

Table 1: Example relation aliases for the relation *"discoverer or inventor"* (P61)

4.2 Identifying Novel Relation Phrases

We propose a prompting strategy to identify relation phrases between head and tail entities. We use the encoder-decoder transformer model BART (Lewis et al., 2019) and bootstrap from existing data sources in order to fine-tune the model for relation generation. Wikidata catalogs relation (property) aliases along with relations (example in table 1). We use the distant supervision setting explained in section 5 to identify paragraphs in which entities that have a relation in the Wikidata KB. Among these paragraphs, we filter out ones in which one of the aliases of the relation between the two entities does not occur. The semi-supervised data to train our prompting models comprises this filtered set of paragraphs and their corresponding head, tail entities, and relation surface-forms.

We experiment with a number of different prompting strategies. The optimal prompt variation, as determined by the results, is illustrated in Figure 4. The remaining three prompting strategies are presented in the appendix (Figure 6). Comparisons of the results of the different strategies on relation identification and unsupervised relation discovery are presented in tables 3 and 4, respectively. We briefly explain these different strategies below. `<ARG>`, `</ARG>`, `<HEAD>`, `</HEAD>`, `<TAIL>`, `</TAIL>`, `<PRED>` and `</PRED>` are all additional trainable embeddings that are fine-tuned with the rest of the model. `<MASK>` is the same mask token embedding used in BART's pre-training tasks.

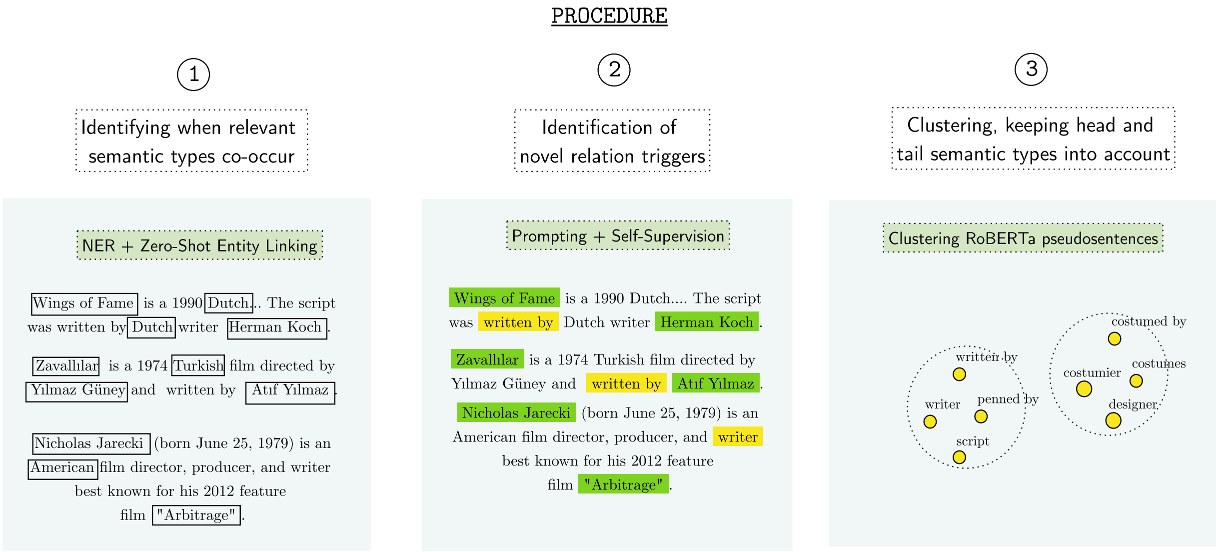

Figure 2: The overall architecture of our relation discovery model.

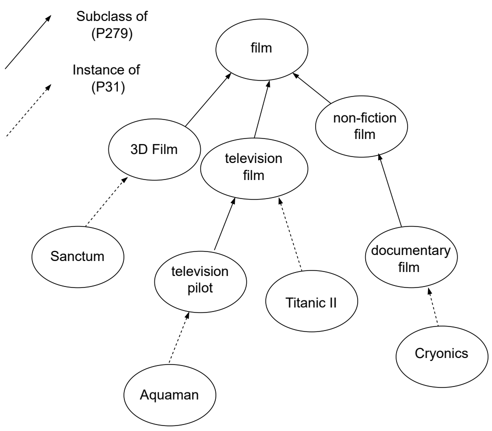

Figure 3: An illustration of our entity typing procedure.

4.2.1 Prompt Variations

Vanilla Prompt: The input to the encoder and the decoder target of BART are illustrated in fig. 6a. The input to the encoder is the candidate paragraph concatenated with the surface forms of the head and tail entities separated by the <MASK> token. The decoder is expected to reconstruct the input and generate the relation alias in place of <MASK>

d1 & d2: These variations are illustrated in figures 6b and 6c. In these variations, the decoder is only expected to generate the relation and entity surface-forms.

del: This variation is illustrated in figure 4. We introduce six additional trainable token vectors to delimit the head and tail in the encoder input and the head, tail and predicate in the decoder target. These trainable vectors serve a dual role. They act as an additional signal to the language model to help identify the head and tail entities for relation extraction and make the relation phrase easier to isolate during post-processing.

4.3 Clustering

Once we obtain the generated relation phrases, we aim to cluster phrases that denote the same underlying relation together in embedding space. Crucial to this step is the disambiguation between relations with the same surface form. For instance, the phrase *"written by"* corresponds to the relation *screenwriter* if it occurs between a human and a movie and the relation *author* if it occurs between a human and a comic book. To address this, we construct pseudosentences using the head and tail semantic types and the relation phrase. For instance, the generated relation phrase *"voiced by"* between a head of semantic type human and a tail of semantic type movie would result in the sentence *"Film voiced by human."*. We obtain the mean-pooled RoBERTa(Liu et al., 2019) embeddings of these pseudosentences and perform k-means clustering on them. The effect of including the head and tail types to our clustering process is shown in table 5. The results of clustering for unsupervised relation discovery is shown in table 4.

5 Dataset

In this section, we briefly describe our dataset Wiki-Art. We utilize the paragraphs scraped us-del:

Encoder Input: <HEAD>Jan Lostowski < /HEAD>(13 January 1951 - 12 November 2009) was a Polish weightlifter. He competed in the men's featherweight event at the <TAIL>1976 Summer Olympics. < /TAIL>. <ARG><HEAD>Jan Lostowski < /HEAD><PRED><MASK> < /PRED> <TAIL>1976 Summer Olympics<TAIL>< /ARG>

Decoder Target: <ARG><HEAD>Jan Lostowski < /HEAD> <PRED> competed in < /PRED> <TAIL> 1976 Summer Olympics < /TAIL> < /ARG>

Figure 4: The highest-performing prompting strategy, **Prompt-del**; additional approaches are discussed in §4.2.1 and illustrated in Figure 6.

ing distant supervision from the REBEL dataset (Huguet Cabot and Navigli, 2021). Under the distant supervision setting, a paragraph containing a head and tail entity is assumed to express a relation between two entities if the two entities are related in a background knowledge base (Wikidata in our case). Head and tail entities are identified using the links from anchor texts. We extract 326 documents (Wikipedia abstracts) from the Wikipedia pages of movies and comic books. The statistics of our dataset are shown in table 2.

We briefly explain the difference between the two settings- *unsupervised relation discovery* and *corpus-based relation discovery*.

Unsupervised Relation Discovery: To compare different IE models on discovery using fully quantitative metrics, we require a uniform number of test instances. So we only evaluate discovery on paragraphs that contain head and tail entities that we know to have a ground-truth relation between them. That is, we ignore paragraphs that contain entities with the head and tail semantic types outlined by our task if the entities do not have a relation between them in Wikidata. The six relations in table 2 are the relations that occur with the highest frequency in our corpus with at least forty occurrences.

Corpus-Based Relation Discovery: Clearly, for realistic relation discovery from a corpus, we can not assume pre-existing knowledge base information to filter out false-positive paragraphs where our task's semantic types co-occur but a relation isn't expressed. So, in this case, we perform discovery on all the paragraphs that contain both task semantic types. In this setting, the model is required to discover all nineteen task-specific relations expressed in the corpus, not just the six most common ones.

6 Experimental Setup

We describe our baselines, metrics and the details of the three experimental settings for relation identification, unsupervised relation discovery and corpus-based relation discovery.

Compared Models We compare the performance of our model against two SOTA OpenIE approaches.

- `Stanford-OpenIE` (Angeli et al., 2015) uses fourteen hand-crafted patterns defined over a dependency parse of the input text sequence in order to identify relational triples.
- `OpenIE5`[6] combines four approaches- CALMIE(Saha and Mausam, 2018), BONIE(Saha et al., 2017), RelNoun(Pal and Mausam, 2016) and SRLIE(Christensen et al., 2011) to extract relational triples. It uses a combination of hand-crafted and automatically mined patterns using syntactic and surface-form information.

6.1 Relation Identification

The purpose of this experiment is to evaluate the relative performance of different models on identifying a relation between two entities directly. As the same relation between two entities can be expressed in a number of ways, we perform this evaluation manually. We report the proportion of instances where the relation is identified accurately among the same 30 randomly sampled paragraphs. The results are shown in table 3.

6.2 Unsupervised Relation Discovery

The distinction between unsupervised relation discovery and corpus-based relation discovery is explained in section 5. The purpose of this experiment is to determine relative performance on relation discovery quantitatively. We report three

[6] `github:dair-iitd/openie-standalone`

Task	#Data	#Relations	Target Relations
Unsupervised Relation Discovrery	813 Paragraphs	6	colorist, prod. designer, dir. of photography, after a work by, screewriter, director
Corpus-Based Relation Discovrery	326 Documents	19	librettist, inspired by, screenwriter, main subject, participant, director, producer, author, after a work by, production designer, choreographer, director of photography, voice actor, film editor, creator, based on, illustrator, cast member, notable work. composer, colorist

Table 2: Dataset Statistics

commonly used clustering metrics. Adjusted Rand Index (**ARI**) measures the number of item pairs in the same vs. different clusters compared to the ground truth label assignment. Normalized Mutual Information (**NMI**) measures the mutual information between assigned and ground-truth cluster assignments. Permutation Accuracy (**ACC**) measures the accuracy between assigned clusters and ground-truth class labels with the best possible permutation matching clusters to labels. The results are shown in table 4.

6.3 Corpus-Based Relation Discovery

This experiment measures the proportion of ground truth relations (table 2) in a corpus we identify using our end-to-end procedure. We follow the same procedure outlined by Huang et al. to evaluate our models. We cluster the relation phrase embeddings of all compared methods into 100 clusters. We isolate the instance closest to the cluster centroid of all 100 clusters. We then manually inspect the isolated instances to determine if the extracted relation phrase corresponds to one of the ground truth relations in the corpus. The results are shown in table 6.

7 Results and analysis

7.1 Relation Identification

The results of our approach on relation identification are tabulated in table 3. Throughout, we indicate the size of our pretraining set in parenthesis. We outperform OpenIE on relation identification by 76 points. From tables 3 and 4 we observe that increasing the size of our pretraining step to 250,000 instances does not improve performance. For future analysis it would be useful to determine how much we can decrease the size of our pre-training step without significantly affecting performance. We also observe that the improved prompting variations outperform the vanilla prompt by 46 points on average. A comparison between the errors of

Approach	Accuracy
Baselines	
Stanford OpenIE	0.11
OpenIE5	0.17
Prompting	
Prompt-pointer n/w (10k)	0.23
Prompt-v (10k)	0.43
Prompt-d1 (10k)	**0.93**
Prompt-d1 (250k)	0.87
Prompt-d2 (10k)	0.87
Prompt-del (10k)	**0.90**

Table 3: Comparing the performance of different techniques on relation identification

Approach	NMI	ARI	ACC
Baselines			
Stanford OpenIE	.25	.03	.36
OpenIE5	.19	.04	.35
Prompting			
Prompt-v (10k)	.32	.16	.47
Prompt-d1 (10k)	.67	.66	.84
Prompt-d1 (250k)	.55	.50	.74
Prompt-d2 (10k)	.65	.64	.81
Prompt-del (10k)	**.68**	**.69**	**.85**

Table 4: Comparing the performance of different approaches on unsupervised relation discovery.

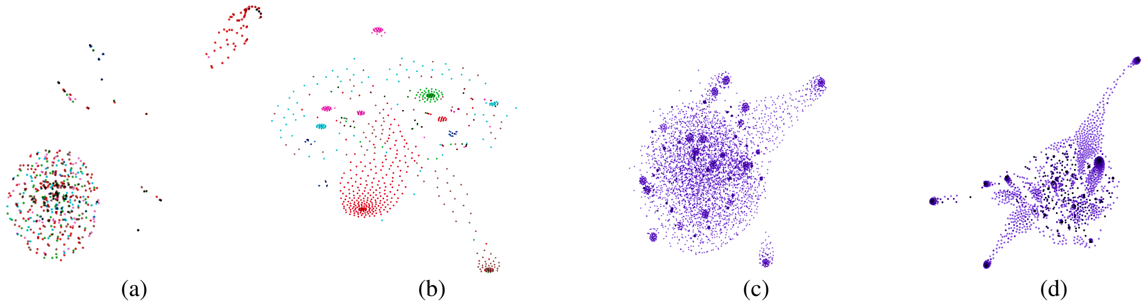

Figure 5: t-SNE visualizations of the relations discovered under the unsupervised relation discovery (a & b) and the corpus-based relation discovery settings (c & d) (sec. 5). Fig. (a) shows the relations discovered using Stanford-OpenIE. Different colors indicate different ground-truth labels. (b) shows the relations discovered using `Prompt-del(10k)` under the same setting. Fig. (c) shows the relations discovered using OpenIE5 and Fig. (d) shows the relations discovered by `Prompt-del(10k)` under the same setting. In this case we do not have ground-truth label assignments as all extracted relation triples are clustered.

Approach	NMI	ARI	ACC
without entity type information			
Stanford OpenIE	.15	-.02	.32
Prompt-v(10k)	.21	.10	.37
Prompt-del(10k)	.60	.62	.77
with entity type information			
Stanford OpenIE	.25	.03	.36
Prompt-v(10k)	.32	.16	.47
Prompt-del(10k)	.68	.69	.85

Table 5: Ablation study comparing the performance of different models with and without head and tail semantic type information taken into account while clustering. The performance of different models is compared on unsupervised relation discovery. For more information about our clustering procedure please refer to section 4.3

Approach	#Relns. Disc.	Relations Missed
OpenIE5	12 / 19	*inspired by* *choreographer* *based on* *notable work* *voice actor* *illustrator* *colorist*
Prompt-del(10k)	15 / 19	*after a work by* *choreographer* *created by* *notable work*

Table 6: Comparing the performance of different models on corpus-based relation discovery. For more details about the evaluation setting please refer to section 6.3

the improved variations and the vanilla prompt can be observed in table 8 in the appendix. Restricting the decoder's output mitigates the problems of poor quality extractions and spelling errors. The improved prompting models generate incorrect extractions when there are multiple relations between the head and the tail or when the relation expressed is semantically complex.

7.2 Clustering & Discovery

The results of our model on unsupervised relation extraction are shown in table 4. `Prompt-del(10k)` outperforms Stanford-OpenIE by 49 points on permutation accuracy. t-SNE visualizations of these results are presented in figures 5a and 5b. We find that all six relations are discovered by our prompting approach and our clustering approach produces coherent, well-separated clusters. The improvements to our clustering strategy by taking head and tail semantic types into account are shown in table 5. Stanford-OpenIE on the other hand, only discovers relations expressed frequently as verbs such as *"written by"* and *"directed by"* with high precision.

Table 6 reports our results on corpus-based relation discovery. Our end-to-end pipeline discovers 15 out of the 19 ground-truth relations in our corpus. Figures 5c and 5d present t-SNE visualizations of our clustering results. We observe that most relations discovered by our framework coalesce into well-separated clusters. The majority of the relations extracted by OpenIE on the other hand, are too general to form well separated clusters.

A clearer illustration of the differences between our approach and OpenIE can be seen in table 7. We observe that both OpenIE baselines perform

Input	Relation Discovery Methods	
	OpenIE5	Prompt-del(10k)
The Iron Giant is a 1999 American... The film stars the voices of Vin Diesel (voicing the titular character)...	(It, was scripted, by Tim McCanlies) (It, was published, in the United States)	<arg> <head> Iron Giant </head> <pred> voiced by </pred> <tail> Vin Diesel </tail>. </arg>
Obelix and Co. is the twenty-third volume of the Asterix comic book series, by Rene Goscinny (stories) and Albert Uderzo (illustrations).	(Co., is, the twenty - third volume of the Asterix comic book series) (Co. is, the twenty - third volume of the Asterix comic book series , by Rene Goscinny) (Obelix, is, the twenty - third volume of the Asterix comic book series , by Rene Goscinny)	<arg> <head> Asterix </head> <pred> illustrator </pred> <tail> Albert Uderzo </tail> . </arg>

Table 7: A comparison between OpenIE and Prompt-del(10k) on the same text spans.

poorly on relations not expressed as verb phrases. As a result, OpenIE5 fails to discover *"colorist"* and *"voiced by"*, two relations frequently expressed as nouns.

8 Conclusions

We formulate a new task of corpus-based task-specific relation discovery and introduce a new dataset for the same. We empirically demonstrate that existing art are inadequate to tackle this task. To address this, we propose an end-to-end self-supervised pipeline for relation discovery that significantly outperforms our baselines on both quantitative and qualitative metrics. In the future, we plan on extending our approach to multiple domains in order to identify and correct possible gaps in our methodology.

Limitations

Domain Shift: In the current implementation, our prompting model relies on the availability of a training set. This assumption may not hold in cases where the relations to be discovered exhibit a significant domain shift from the training set. To address this limitation, future work should explore fully unsupervised prompting approaches that can better adapt to new domains and mitigate the impact of domain shift.

Limited Number of Relations: In this study, our analysis is restricted to a total of 25 relations. While this allows for a focused exploration of these specific relations, it also limits the scope and potential applications of our model. To broaden the applicability and effectiveness of our approach, future work should aim to utilize Wikidata more comprehensively, incorporating a larger number of relations for more extensive and diverse analysis.

Ethics Statement

We conform to the ACL ethics policy. Our research utilizes data from Wikipedia, which is governed by the Creative Commons Attribution-ShareAlike License, and Wikidata, which is governed by the CC0 Public Domain Dedication License. Importantly, we have taken care to ensure that no personal information of any user is used in our study.

References

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

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *K-CAP '11*.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. Question answering on knowledge bases and text using universal schema and memory networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–365, Vancouver, Canada. Association for Computational Linguistics.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *AAAI*.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. 2016. Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268, Berlin, Germany. Association for Computational Linguistics.

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.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Mausam, and Soumen Chakrabarti. 2020. IMoJIE: Iterative memory-based joint open information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5871–5886, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Sha Li, Liyuan Liu, Yiqing Xie, Heng Ji, and Jiawei Han. 2022. Piled: An identify-and-localize framework for few-shot event detection. *arXiv preprint arXiv:2202.07615*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ArXiv*, abs/2107.13586.

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

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? paie: Prompting argument interaction for event argument extraction. *arXiv preprint arXiv:2202.12109*.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Harinder Pal and Mausam. 2016. Demonyms and compound relational nouns in nominal open ie. In *AKBC@NAACL-HLT*.

Matthew E. Peters, Mark Neumann, Robert L Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *EMNLP*.

Swarnadeep Saha and Mausam. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Swarnadeep Saha, Harinder Pal, and Mausam. 2017. Bootstrapping for numerical open IE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323, Vancouver, Canada. Association for Computational Linguistics.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. Association for Computational Linguistics.

Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A. Gers, and Alexander Löser. 2017. Analysing errors of open information extraction systems. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 11–18, Copenhagen, Denmark. Association for Computational Linguistics.

Shikhar Vashishth, Prince Jain, and Partha Pratim Talukdar. 2018. Cesi: Canonicalizing open knowledge bases using embeddings and side information. *Proceedings of the 2018 World Wide Web Conference*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, Lisbon, Portugal. Association for Computational Linguistics.

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 353–362, New York, NY, USA. Association for Computing Machinery.

A Comparative Error Analysis of Prompting Strategies

In our evaluation, we conducted an error analysis to compare the relative performance of different prompting models, which provided insights into their strengths and weaknesses. Qualitative results are displayed in Table 8.

B Prompting Strategies

In our exploration of prompting strategies for identifying relation phrases, we examined three additional alternative approaches, as illustrated in figure 6.

Example	Category
Vanilla Prompt (Prompt-v)	
Encoder Input: The cinematography was done by R. Rathnavelu, and the film was edited by Suresh Urs. The film score and soundtrack were composed by V. Harikrishna. "Gaalipata" was the first in Kannada to be shot in Super 35 format. <arg> Gaalipata <MASK> R. Rathnavelu </arg> Decoder Output: The cinematography was done by R. Rathnavelu, and the film was edited by Suresh Urs. The film score and soundtrack were composed by V. Harikrishna. "Gaalipata" was the first in Kannada to be shot in Super 35 format. <arg> Gaalipata written by Rathavelu </arg>	Poor quality extraction, Relation not present in input paragraph.
Encoder Input: ... Aagadu marked S. Thaman's 50th film as a music composer. K. V. Guhan provided the cinematography ... Decoder Output: ... <arg> Aagadatu provided cinematography K. J. Vhan. </arg>	Spelling errors, Hard to isolate relation phrase for clustering.
Encoder Input: ...Love from Paris (German: Monpti) is a 1957 West German romantic comedy-drama film dir... <arg> Love from Paris <MASK> Heinz Pehlke </arg> Decoder Output: ...unds of Paris. Love from Paris writer Heinz Penhlke.	Relation delimiters not generated, Hard to isolate relation phrase for clustering.
Prompt-d1	
Encoder Input: House of Bamboo is a 1955 American film noir shot in CinemaScope and DeLuxe Color. It was directed and co-written by Samuel Fuller. <arg> House of Bamboo <MASK> Samuel Fuller </arg> Decoder Output: <arg> House of Bamboo written by Samuel Fuller </arg>	Multiple relations between head and tail entities
Encoder Input: Road to Perdition is a 2002 American crime drama film directed by Sam Mendes. The screenplay was adapted by David Self from the graphic novel of the same name written by Max Allan Collins and illustrated by Richard Piers Rayner. <arg> Road to Perdition <MASK> Max Allan Collins </arg> Decoder Output: <arg> Road to Perdition written by Max Allan Collins. </arg>	Hard relations with multiple qualifiers
Encoder Input: Pavithram () is a 1994 Indian Malayalam-language drama film directed by T. K. Rajeev Kumar and written by P. Balachandran from a story by Balachandran and Kumar. <arg> Pavithram <MASK> T. K. Rajeev Kumar. </arg> Decoder Output: <arg> Pavithram written by T. K. Rajeev Kumar. </arg>	

Table 8: Error analysis comparing the relative performance of different prompting models.

Encoder Input

Jan Lostowski (13 January 1951 - 12 November 2009) was a Polish weightlifter. He competed in the men's featherweight event at the 1976 Summer Olympics. <ARG> Jan Lostowski <MASK> 1976 Summer Olympics < /ARG>

Decoder Target

Jan Lostowski (13 January 1951 - 12 November 2009) was a Polish weightlifter. He competed in the men's featherweight event at the 1976 Summer Olympics. <ARG> Jan Lostowski **competed in** 1976 Summer Olympics < /ARG>

(a)

d1:

Encoder Input

Jan Lostowski (13 January 1951 - 12 November 2009) was a Polish weightlifter. He competed in the men's featherweight event at the 1976 Summer Olympics. <ARG> Jan Lostowski <MASK> 1976 Summer Olympics < /ARG>

Decoder Target

<ARG> Jan Lostowski **competed in** 1976 Summer Olympics < /ARG>

(b)

d2:

Encoder Input

Jan Lostowski (13 January 1951 - 12 November 2009) was a Polish weightlifter. He competed in the men's featherweight event at the 1976 Summer Olympics. <ARG> Jan Lostowski <MASK> 1976 Summer Olympics < /ARG>

Decoder Target

<ARG> Jan Lostowski 1976 Summer Olympics `competed in` < /ARG>

(c)

Figure 6: Three Alternative Prompting Strategies for Identifying Relation Phrases: The Optimal Strategy, **Prompt-del**, is Displayed in Table 4. The Strategies, Arranged from Top to Bottom, Include **Prompt-v** (Top), **Prompt-d1** (Middle), and **Prompt-d2** (Bottom).