Understanding Distantly Supervised Relation Extraction through Semantic Error Analysis

Jan-Christoph Kalo* J.C.KALO@VU.NL
Knowledge Representation and Reasoning Group, Vrije Universiteit Amsterdam
DReaMS Lab, Huawei, The Netherlands

Benno Kruit* B.B.KRUIT@VU.NL
Knowledge Representation and Reasoning Group, Vrije Universiteit Amsterdam
DReaMS Lab, Huawei, The Netherlands

Stefan Schlobach K.S.SCHLOBACH@VU.NL
Knowledge Representation and Reasoning Group, Vrije Universiteit Amsterdam

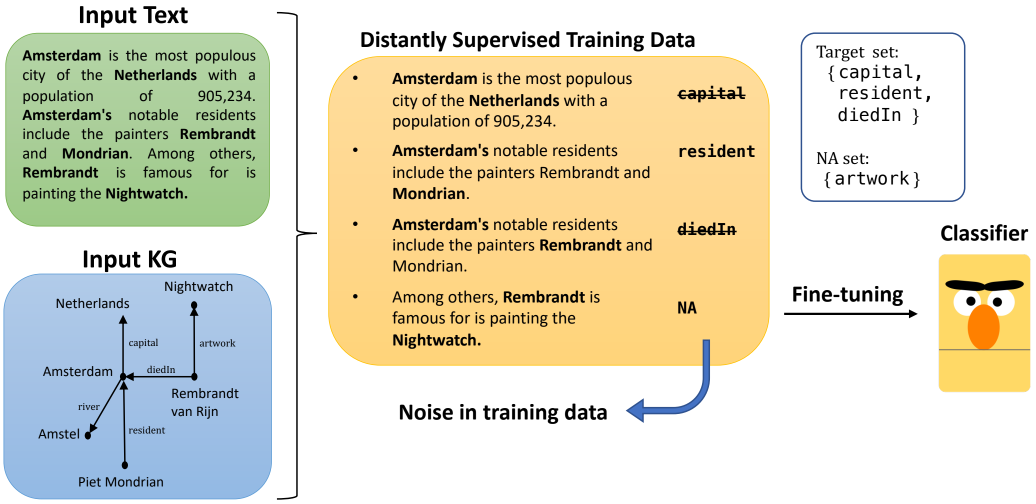
Abstract

Automatic knowledge graph construction, using supervised relation extraction from text, has become the state-of-the-art to create large-scale repositories of background knowledge for various applications. Recent advances in machine learning and Natural Language Processing (NLP), in particular the advent of the large language models, have improved the performance of relation extraction systems significantly. Traditional leaderboard style benchmark settings show very high performance, suggesting that these models can be employed in practical applications. Our analysis shows that in reality, though, the extraction quality varies drastically from one relation to another, with unacceptable performance for certain types of relations. To better understand this behaviour, we perform a semantic error analysis on a popular distantly supervised benchmark dataset, using ontological meta-relations to describe various error categories, which shows that relations that are confused by state-of-the-art systems are often semantically closely related, e.g., they are inverses of each other, in subproperty relations, or share the same domain and range. Such an extensive semantic error analysis allows us to understand the strengths and weaknesses of extraction models in a semantic way and to provide some practical recommendations to improve the quality of relation extraction in the future.

1. Introduction

Building knowledge graphs is still a very costly process as facts often have to be curated manually [Vrandečić and Krötzsch, 2014]. While various extraction techniques from semi-structured data have helped to automate the process of large-scale knowledge graph creation [Auer et al., 2007], modern KG applications require the extraction of facts for increasingly many distinct relation types at high precision. To meet these requirements, information (or triple) extraction is often split into two steps: entity linking and relation extraction [Martinez-Rodriguez et al., 2020]. Relation extraction is usually formulated as a classification problem: given a sentence and entity mentions the goal is to predict one of multiple relationships of a given KG schema. Large amounts of training data may be generated by *distant supervision* [Mintz et al., 2009], where training data is created automatically by linking an existing knowledge graph to textual data, which is then used for some form

*. Equal contribution



Amsterdam is the most populous city of the **Netherlands** with a population of 905,234. **Amsterdam's** notable residents include the painters **Rembrandt** and **Mondrian**. Among others, **Rembrandt** is famous for is painting the **Nightwatch.**

- **Amsterdam** is the most populous city of the **Netherlands** with a population of 905,234.
- **Amsterdam's** notable residents include the painters Rembrandt and **Mondrian**.
- **Amsterdam's** notable residents include the painters **Rembrandt** and Mondrian.
- Among others, **Rembrandt** is famous for is painting the **Nightwatch.**

Figure 1: Overview of Distantly Supervised Relation Extraction. The first and third generated training examples are incorrect, the second and forth ones are correct.

of fine-tuning of large pre-trained language models (Figure 1). While this kind of training data generation may lead to some noisy training examples, its quality is usually sufficient to produce acceptable results, which makes relation extraction with distant supervision the de-facto standard approach for creating knowledge graphs automatically.

Traditionally, relation extraction (RE) for knowledge graph construction is evaluated with standardized benchmark datasets, where novel methods are evaluated using their averaged F1 scores in long leader board lists[1]. In particular, training RE models using distant supervision is a popular way to expand an existing KG with new information. Most recent distantly supervised relation extraction (DSRE) methods reach F1 scores of 80%-90%, which might suggests that the task is almost *solved*: Methods are ready for real applications and can be employed to automatically construct high-quality knowledge graphs from textual data. However, a more detailed analysis of a recently published manually labeled dataset shows that the reality is different, and rather more bleak. This is because the averaged measures reported in leaderboard lists hide the fact that the individual extraction quality of relations varies significantly, with performance for some relations unacceptably low. Even though the best performing model in our experiments achieves more than 85% macro-averaged as an overall F1 score, large parts of the extracted relations have an F1 score below 50%, while others easily achieve 100%.

Our in-depth analysis reveals that models actually still make many mistakes and cannot be used in practical scenarios directly as of yet. Instead, more careful considerations of their strengths and weaknesses are necessary. A typical example is that models often confuse *similar* relations, e.g., `capital` and `largestCity`. Using well understood notions from the Semantic Web literature, we use ontological meta-relations to describe the confusion of extraction models. Take `capital` and `largestCity`, which are actually connected via a shared superproperty `city` and, hence, are siblings according to the ontological schema.

[1] https://paperswithcode.com/task/relation-extraction

In this paper, we show that, among other meta-relations, sibling relations are frequently confused by recent models.

In this paper we come up with semantic error categories for relation extraction built on well-known ontological concepts to make a *semantic error analysis*. Thus, we are able to classify errors into different categories, providing a comprehensive overview of the strengths and weaknesses of more recent model types. The insights can be used in practice to pick an appropriate relation extraction model architecture, refine the knowledge graph ontology, gather additional training data, or add a human-in-the-loop for difficult semantic categories. Our datasets and an interactive Jupyter Notebook are available on GitHub. [2].

Overall, our contributions can be summarized as follows.

1. We employ meta-relations, i.e., `inverseOf`, `subproperty`, `superproperty`, and others, to describe typical confusions of state-of-the-art distantly supervised relation extraction models on a representative benchmark.
2. We employ this knowledge to create semantic error categories that give the user a better understanding of the problems of current extraction models.
3. We give recommendations on how these semantic insights can be used to improve the distantly supervised relation extraction quality in practice.

2. Relation Extraction: Basics and State-of-the-Art

In most existing work, relation extraction is formulated as a classification task, where the input is a sentence and a pair of entities that are mentioned in this sentence. The output is, then, one (or multiple) relations between the given pair of entities. As most other supervised classification tasks in Natural Language Processing, the best results have recently been obtained by approaches that leverage large pre-trained Neural Language Models, which can be categorized into two groups. In *fine-tuning*-based relation extraction, a classification layer is added to a pre-trained language model [Soares et al., 2019], which is then fine-tuned, i.e., trained on the training dataset to extract relations from text. More recently, *prompt tuning*-based approaches have shown a slight increase in performance [Han et al., 2021, Chen et al., 2021, Shin et al., 2020, Josifoski et al., 2021]. Here, the problem is modeled as a word prediction task for which the language model is trained to predict a word (out of all relation labels) connecting the two entity labels [Liu et al., 2021].

In both paradigms, the input consists of sentences with two highlighted entities. Moreover, the input is annotated with some additional tokens: (1) `<CLS>` is a special token added to the beginning of each input sentence. Later in the classification process, the embedding vector of the CLS token is used to perform sentence classification. (2) `<E1>` and `</E1>` are a start and end token to mark the words that belong to the first entity, (3) `<E2>` and `</E2>` similarly for the second entity. In multi-class relation extraction, as considered in this work, the output is usually represented as one relation, out of a set of target relations, which has to be classified as either correct or incorrect.

In practice, many input sentences express none of the target relations. This problem is often solved by introducing an additional class, called `NA` (none of the above), which is handled like the other classes. However, selection of good training data for the NA class is difficult given its broad semantics, as it is meant to cover all other relations.

[2] https://github.com/JanKalo/SemanticErrorAnalysis

Distant supervision is the process of generating training data automatically from a text corpus and an existing knowledge graph to overcome the lack of manually annotated training data [Mintz et al., 2009]. Whenever a given sentence contains entities with an existing triple in a knowledge graph, it is assumed that the sentence expresses at least this triple and thus can be used for training a relation extraction classifier for the respective relation. As an example of distant supervision, we present a small text corpus, a knowledge graph, and the resulting training examples in Figure 1. Here, the first sentence contains the entities `Amsterdam` and the `Netherlands`. In the knowledge graph, these two entities are related by the `capital` relation, so that this sentence is used as a training example for the relation class `capital`. This example shows, though, that distant supervision can construct noisy training data as the sentence does actually **not** express the capital relation (it only states that Amsterdam is the city with the most inhabitants).

Similar to supervised relation extraction, the choice of negative training examples for the NA class is a difficult and, as yet, unsolved problem. In some work the NA class is randomly sampled, but is also created from relations that are not part of the extraction process [Gao et al., 2021]. In the fourth example, the sentence is labeled as `NA` instead of `artwork`, due to definition of the NA class. Later in this paper we will discuss the impact of the definition of the NA class on the extraction results.

Usually, distant supervision is a reliable approach to generate large amounts of high-quality training data. Before the advent of the language model-based approaches, still, a lot of research went into improving distant supervision training and training data [Roth et al., 2013, Smirnova and Cudré-Mauroux, 2018, Ye and Ling, 2019, Shang et al., 2020].

Datasets. The availability of datasets which can be used to evaluate distantly supervised relation extraction is rather limited. The most popular benchmarks is *NYT10*, which is based on Freebase and NYT articles. Unfortunately, this dataset has several quality issues [Riedel et al., 2010, Surdeanu et al., 2012] and even more problematic is the lack of manually annotated test data for an informative evaluation of systems. While such a manually created test set was recently published by Gao et al. [Gao et al., 2021] under the name *NYT10m*, it still only has a small number of 25 relations (most of them covering geographic relations). In addition, the authors proposed *wiki20m*, a new dataset built from earlier distantly supervised relation extraction datasets based on Wikidata and English Wikipedia articles, using existing hyperlinks as entities. In the test set, all sentences were manually filtered by human annotators, thus guaranteeing a high quality.

We have opted for using this dataset in our paper because it is at the moment the largest DSRE dataset available that has a manually annotated test set, as well as a set of target relations that are described by a highly expressive knowledge graph using meta-relations. We further discuss the choice of this dataset in Section 4.2.

3. Motivation: Standard Benchmark Revisited

In typical relation extraction papers, state-of-the-art models are evaluated on existing benchmark datasets, similar to the ones described in the previous section, focusing on **averaged performance metrics**. In this section, while we replicate state-of-the-art results on the recently published distantly supervised wiki20m dataset, our focus is a more detailed analysis of the evaluation results. We will show that there is a significant discrepancy between the average performance across relations and the performance on individual relations, which motivates our semantic error analysis provided later in the paper.

Benchmark Setup The *wiki20m* dataset comprises 698,721 / 64,607 / 137,986 sentences in the training / test / validation set, respectively. In addition to the 80 target relations, one NA (none of the above) class is used. The training and test examples for the NA class consist of a set of Wikidata relations which are not part of the 80 target relations.[3]

Our relation extraction methods comprise several BERT-based approaches [Han et al., 2020] based on both fine-tuning and a recent prompt-based approach [Han et al., 2021]. The BERT-based techniques are available through the framework OpenNRE[4], and based on pre-trained BERT models. The second big model class that we are using is the prompt-based model PTR [5]. Overall, these adds up to the following models:

- The BERT-CLS model uses the `<CLS>` token of BERT to perform the classification task, which is the standard approach for sentence classification tasks in BERT.
- The literature has shown small improvements when special entity tokens are used to mark the entity mention span in input sentences. Based on this, the BERT-ENT model uses the embedding vectors of special entity tokens to perform classification.
- Additionally, we use *masked* versions of the same models (BERT-M-CLS and BERT-M-ENT). This means that the entity mentions are hidden during training, so that the model is less prone to biases from entity names.
- PTR is prompt-tuning approach with rules which uses a set of manually defined prompts [Han et al., 2021]. The manually written prompts for this dataset have been written by the authors and are available as supplemental material.

Results The performance of the five relation extraction systems is evaluated using precision, recall, micro-, and macro-averaged F1-measure. The micro-averaged scores reflect the precision and recall as averaged over all test instances, thus they are more influenced by relations that occur more frequently in the test set. Macro-averaged scores, in contrast, are averaged over all relations and thus treat them equally. As both training and test sets are very imbalanced (reflecting different data collection distributions of the distant supervision and manual annotation processes), micro- and macro-averaged scores can differ significantly on this task. The overall results are presented in Table 1.

We observe that the two masked systems (BERT-M-CLS, BERT-M-ENT) have similar average performances, with an accuracy of around 40% and low F1 measures. While BERT-CLS outperforms both masked models, it is in turn dominated by BERT-ENT and PTR, which achieve F1 measures of over 80%.

Due to the choice of distinct NA classes in training and test dataset in the creation of wiki20m, the number of confusions with the NA class is high. In the right most column in Table 1, we show the errors which are not due to confusions with the NA class. These non-NA errors are hardly influenced by the models themselves, i.e., significantly better models

[3] Unlike other distantly supervised datasets, the Wikidata relations used for forming the NA class in training set are distinct from the set of relations used to create the test set. This leads to different semantics of the NA class in training and test, which is a questionable design decision for the benchmark dataset. We will therefore not focus on the errors caused by the NA class, but leave this for future work.

[4] https://github.com/thunlp/OpenNRE

[5] https://github.com/thunlp/PTR

Table 1: Overall performance on *wiki20m*.

	Macro			Micro				Non-NA
Model	Precision	Recall	F1	Precision	Recall	F1	Acc.	Errors
BERT-M-CLS	0.62	0.30	0.35	0.61	0.31	0.41	0.43	0.15
BERT-M-ENT	0.69	0.32	0.37	0.60	0.32	0.41	0.41	0.16
BERT-CLS	0.79	0.68	0.71	0.79	0.68	0.73	0.71	0.14
BERT-ENT	0.84	0.78	0.79	0.84	0.78	0.81	0.79	0.11
PTR	0.83	0.82	0.81	0.83	0.81	0.82	0.79	0.12

Table 2: The five worst classified relations ordered by F1 measure for BERT-ENT and PTR. We also provide Precision and Recall values.

	PTR				BERT-ENT		
Relation	Precision	Recall	F1	Relation	Precision	Recall	F1
residence	0.63	0.26	0.36	after a work by	0.91	0.14	0.24
screenwriter	0.30	0.52	0.38	residence	0.76	0.17	0.28
part of	0.48	0.34	0.40	part of	0.46	0.23	0.31
after a work by	0.85	0.32	0.47	followed by	0.96	0.26	0.41
work location	0.45	0.54	0.49	screenwriter	0.39	0.50	0.44

like BERT-ENT and PTR achieve much better overall results, but the non-NA errors are hardly influenced. Therefore, in this work we only focus on non-NA errors.

In leaderboards, usually the average F1 measures and the accuracy is reported. With around 80% in both measures, these models seem, apparently, to achieve an excellent quality in relation extraction. The actual performance, however, varies dramatically between the different relations. Table 2 shows the performance of the bottom five relations from the two BERT-ENT and the PTR model. Note that the worst relation (`after a work by`) only achieves an F1-measure of as low as 24%. This is significantly worse than the average F1 measures over all relations. For many practical scenarios this is far from being acceptable. Furthermore, this low performance cannot be attributed to the lack of training data. The dataset contains ample training examples for these relations, but models score better on other, less frequent relations. This under-performance is therefore not a long-tail problem. Even though both models perform bad for similar relations (`after a work by`, `residence`, `part of`, `screenwriter`), there are also important differences. `followed by` is significantly better handled by PTR. Motivated by these first findings, we further investigate the model's confusions.

Model Confusions As a first step of a more in-depth error analysis, let us look at the model confusions, i.e., a matrix showing the predicted relations and the relations they were confused with by the model. Since the entire confusion matrix for 80 classes is hard to visualize, we focus on the top confusions for the two best models PTR and BERT-ENT in Table 3. It is clear that confused relations are highly similar to each other. Often, e.g., they share the same domain and range: the types of entities in subject and object position. The top of the confused relation pair for both models are `residence` and `work location`,

Table 3: Most frequently confused relations for the *BERT entity* model and *PTR*. *Gold Relation* is the true label in the test dataset and *Prediction* the relation classifier's prediction.

Model	Gold Relation	Prediction	Frequency
PTR	residence	work location	515
	after a work by	screenwriter	367
	location of formation	headquarters location	247
	publisher	developer	178
	tributary	mouth of the watercourse	158
BERT-ENT	residence	work location	767
	after a work by	screenwriter	395
	followed by	follows	350
	publisher	developer	228
	headquarters location	location of formation	224

which have a very similar semantics; for many persons they coincide (are even identical), since they live at the same place they are working.

4. A Semantic Error Analysis of Relation Extraction

In the previous section we have shown that for a large number of relations automatic extraction still gives unacceptably bad results, which makes the approach unsuitable for automatic knowledge graph construction. More specifically, our experiments have revealed that highly similar relations often are confused by models. These problems occur due to the noise in the training data, as well as to the problems related to the prediction models. In this section, we perform a semantic error analysis of these confusions, by employing ontological information about relations to describe and categorize them.

4.1 Creating Semantic Error Categories

When evaluating the RE models, we can categorize their errors using *meta-relations*. Here, we consider meta-relations to be relations describing the ontology schema, i.e. the relations between the relations, e.g., their domain and range, sub-/superproperty, or inverse. The meta-relations that we use stem from the RDFS and OWL vocabulary, such as `rdfs:domain`, `rdfs:range`, `rdfs:seeAlso`, and `rdfs:subPropertyOf`, as well as `owl:inverseOf`.

Since we are working with Wikidata, we employ the Wikidata equivalents to the respective RDFS and OWL relations. In particular, we use *inverse property* (`P1696`), *subproperty of* (`P1647`) and *related properties* (`P1659`) for the last three mentioned meta-relations. Using the subproperty relation, we materialize the property hierarchy. However, not every meta-relation discussed above has a direct match. Instead of defining the range and domain of relations as per RDFS, Wikidata employs *property constraints* (`P2302`) to describe for which classes of entities the property should be used, using qualifers of constraint entities. The union of qualifier values (e.g. using `owl:unionOf`) for these constraint entities (*type constraint* (`Q21503250`) for subjects and *value-type constraint* (`Q21510865`) for objects) may then be considered to describe the range and domain of the properties.

For this analysis, our aim is to create a semantic confusion matrix which categorizes the confusions as shown in Table 3 into semantic categories. The goal is to create a categorization that is as informative as possible when describing the errors that the RE models make. Therefore, we extend the basic meta-relations in various ways:

- *inverse:* The two relations are directly inverse to each other, e.g. `following` and `followed by` .
- *subProperty:* One property is either directly or transitively a sub relation of another relation, e.g. `capital` and `location`.
- *superProperty:* This category is the inverse category to the previous one, e.g. `location` and `capital`.
- *sibling:* Sibling relations share the same super property, either directly or transitively, e.g. `work location` and `residency` both have the same superproperty `location`.
- *seeAlso:* The two relations are connected via a `seeAlso` relation in Wikidata, since they are related to each other, but are not in any of the other meta-relations, e.g. *tributary* and *mouth of the watercourse*.
- *rangeDomainMatch:* The two relations share at least one subject and object type constraint specified in Wikidata, e.g. `publisher` and `developer` share the subject type constraint `video game series` and the object type constraint `organization`.
- *onlyDomainMatch:* The two relations share at least one subject type but no object type constraint, e.g. `distributor` and `genre` share the subject type constraint `movie`.
- *onlyRangeMatch:* The two relations share at least one object type but no subject type constraint, e.g. `head of government` and `architect` share the object type constraint `human`.

Some relation pairs may be meta-related both by *sub/superProperty* and *sibling* (e.g. if they form a chain in the property hierarchy). The match-based categories and *seeAlso* may also overlap with other categories. Despite our efforts to categorize all model errors, there are still confused relations that do not fit our scheme. `field of work` may, e.g., be confused for `sport`, as both might concern a celebrity's claim to fame, but these properties are not directly related in the Wikidata ontology. However, we have observed that the best-performing models also make fewer uncategorized errors, i.e. the most challenging errors are those that can be described in our scheme.

4.2 Ontology-driven Semantic Error Analysis

As a first analysis, we investigate the frequency of the different error categories introduced above for each of the relation extraction models (Figure 2). Please note, that each confusion (error) can be counted for multiple error categories. The errors depicted in the figure are the relative number of errors with respect to the overall number of predictions. Hence, the numbers can be compared between models and between the different categories.

The most frequent errors are *domain*, *rangeDomain* and *sibling* errors. There are significantly fewer errors for relations which only share the same *range*. *Inverse* and *sub-* or *superproperty* errors are even more rare. Please note, however, that the differences between different error categories might also be due to the choice of relations in wiki20m, as there are many properties without any *inverse* or *superproperty* relations.

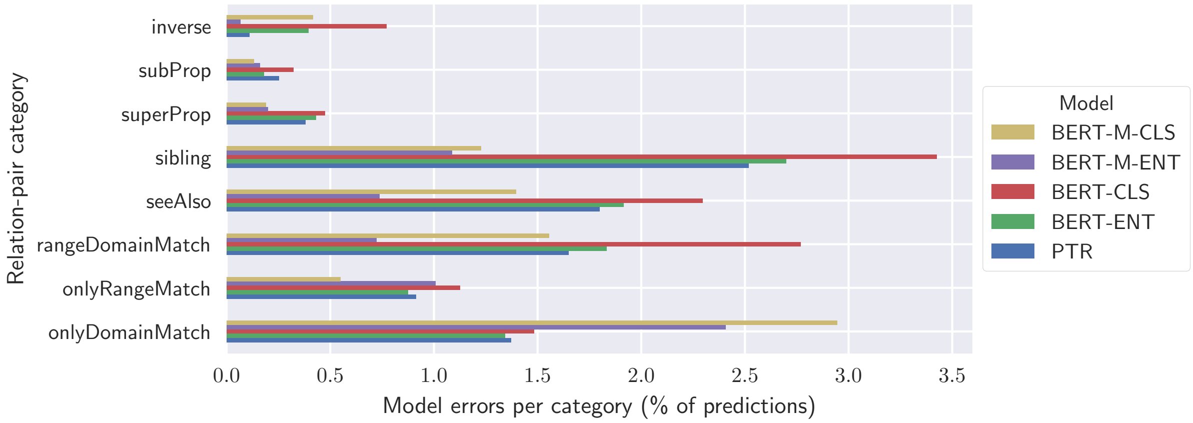


Figure 2: Frequency of errors for the semantic error categories per model.

Table 4: The most frequently confused relations for the *BERT entity* model and *PTR*. *Gold Rel.* is the true label in the test dataset and *Prediction* the relation classifier's prediction.

Model	Gold Rel.	Prediction	Freq.	Category
PTR	`residence`	`work loc.`	515	*sibling / seeAlso / rangeDomainMatch*
	`after a work by`	`screenwriter`	367	*sibling / rangeDomainMatch*
	`loc. of formation`	`headquarters loc.`	247	*sibling / seeAlso / onlyDomainMatch*
	`publisher`	`developer`	178	*rangeDomainMatch*
	`tributary`	`mouth of watercourse`	158	*seeAlso / rangeDomainMatch*
BERT-ENT	`residence`	`work loc.`	767	*sibling / seeAlso / rangeDomainMatch*
	`after a work by`	`screenwriter`	395	*sibling / rangeDomainMatch*
	`followed by`	`follows`	350	*inverse*
	`publisher`	`developer`	228	*rangeDomainMatch*
	`headquarters loc.`	`loc. of formation`	224	*sibling / seeAlso / onlyDomainMatch*

Masking models make significantly more errors in the *onlyDomainMatch*, but perform quite well in the other categories. Overall, they seem to make fewer errors that are covered by our semantic error analysis. This indicates, however, that there are more confusions with the NA-class. Another interesting insight is that the prompt-based model (PTR) hardly makes any inverse confusions, while this frequently happens with the BERT-CLS model.

Interestingly, the non-masking models (BERT-ENT, BERT-CLS, and PTR) have more trouble working with *sibling* relations, or relations sharing the same *domain* and/or *range*. This may be a problem for non-masking models, since they use entity mentions as additional information for the classification. Since highly related relations are usually about entities of the same type, using entity information for the classification might confuse the models predictions.

Table 4 provides a more detailed analysis over the different error categories and the most frequent pairs of confused relations. Note that the errors that are made seem rather *natural*, i.e., since most of the errors are between relations that are semantically very similar and therefore can also be hardly distinguished by us humans[6] It is fair to say that these errors can be considered the hard, and challenging, cases.

An open question remains the generalization of these results to other settings and benchmark datasets. Our analysis is motivated by the application of DSRE to extend an existing knowledge graph, for which the *wiki20m* dataset provides a controlled setting using an semantically-described set of target relations. This may cause relations to be confused due to distant supervision noise or biases, due to the closed-world assumption that is inherent in training. Furthermore, in other domains there may be significant effects due to a domain shift between the KG used for distant supervision and the text corpus.

5. Recommendations and Conclusion

While our semantic error analysis is helps understanding the details of the performance of relation extraction systems on benchmark datasets, these insights can also have practical implications:

1. **Carefully consider the choice of model** The results from our semantic error analysis as presented in Figure 2 gives an overview of the strengths and weaknesses of different relation extraction model types. In a practical scenario, the semantic error categories of the relations of interest can be used to inform the choice of RE model. For example, the PTR and masking models perform best on inverse relations, so these may be preferred in scenarios where many inverse relations need to be extracted.
2. **Refine the knowledge graph** Another possibility to prevent mistakes in automatic extraction, is to refine the ontology of the knowledge graph. Our analysis has shown that many confused relations are highly similar, often even in a *superproperty* relation. In these cases, it might make sense to merge existing relations of the knowledge graph, depending on the intended application. This is similar to what the authors did in the improved version of the relation extraction dataset Re-TACRED [Stoica et al., 2021], where the original dataset contained many confusions between semantically similar relations, thus they were collapsed to improve the dataset quality.
3. **Add additional training examples or perform active learning** As it broadly holds that semantically similar relations are difficult to distinguish by relation extraction models, in many cases it will be useful, and even necessary, to gather additional training data for those complex error categories, e.g. through an informed active learning step where additional complex examples are manually annotated.
4. **Use semantic error analysis to support humans in the loop** The knowledge about typical error categories and their semantics can furthermore be exploited by human-in-the-loop approaches to the extraction process. A semantic error analysis of confused relations can help pinpoint annotators towards predictions according to our categories together with additional information on the type of errors and semantically enhanced annotation guidelines.

Conclusion In this paper, we have first given an extensive overview of the state-of-the-art relation extraction techniques and their implications for automatic knowledge graph cre-

[6] The property discussion pages from Wikidata reflect this insight https://www.wikidata.org/wiki/Property_talk:P974.

ation with distant supervision. We have shown that, even though recent methods achieve very high numbers in averaged benchmark scores, i.e. around 80% F1 score on a a recent Wikipedia-based relation extraction dataset, these numbers hide the actual variety of the performance across different relations. Starting from a confusion matrix, we further investigate the prediction errors that the models make. Our analysis reveals that most of those errors can be described through ontological meta-relations, and categorized into different confusion categories based on those meta-relations. Additionally, we provide recommendations on how the insights of a semantic error analysis may improve relation extraction in practice, e.g. how to adjust the KG ontology to reduce the impact of RE model weaknesses.

For future work, we plan to extend our work by studying more datasets and more relation extraction systems. This might reveal even more architectural differences and interesting insights. It would also be interesting to analyze which errors are due to noise in the training data from the distant supervision process. We also plan to make the error categories more fine-grained. Finally, we will study ways to automate the recommendations discussed in the previous sections, and experimentally quantify their impact. In general, we believe that our method can also be transferred to other machine learning problems as well to give a valuable insight into the performance and robustness of large models.

Acknowledgments

This research was funded by Huawei's DReaMS Lab.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the International Semantic Web Conference (ISWC)*, pages 722–735, 2007.

Xiang Chen, Ningyu Zhang, Chuanqi Tan, Fei Huang, Luo Si, Huajun Chen, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction. *Proceedings of the ACM Web Conference 2022 (WWW '22), Virtual Event, Lyon, France*, 1(1), 2021. doi: 10.1145/3485447.3511998. URL http://arxiv.org/abs/2104.07650.

Tianyu Gao, Xu Han, Yuzhuo Bai, Keyue Qiu, Zhiyu Xie, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Manual Evaluation Matters: Reviewing Test Protocols of Distantly Supervised Relation Extraction. pages 1306–1318, 2021. doi: 10.18653/v1/2021.findings-acl.112.

Xu Han, Tianyu Gao, Yuan Yao, Demin Ye, Zhiyuan Liu, and Maosong Sun. OpenNRE: An open and extensible toolkit for neural relation extraction. *EMNLP-IJCNLP*, pages 169–174, 2020. ISSN 2331-8422. doi: 10.18653/v1/d19-3029. URL https://aclanthology.org/D19-3029.pdf.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. PTR: Prompt Tuning with Rules for Text Classification. 2021. URL http://arxiv.org/abs/2105.11259.

Martin Josifoski, Nicola De Cao, Maxime Peyrard, and Robert West. GenIE: Generative Information Extraction. 2021. URL http://arxiv.org/abs/2112.08340.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. 2021. doi: 10.48550/ARXIV.2107.13586. URL https://arxiv.org/abs/2107.13586.

Jose L. Martinez-Rodriguez, Aidan Hogan, and Ivan Lopez-Arevalo. Information Extraction meets the Semantic Web: A Survey. *Semantic Web*, 0 (2), 2020. ISSN 22104968. doi: 10.3233/SW-180333. URL https://pdfs.semanticscholar.org/ae8c/331e091ba27e2671cdc63c44982b9fe66e98.pdfhttps://www.semanticscholar.org/paper/ae8c331e091ba27e2671cdc63c44982b9fe66e98.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. (August):1003–1011, 2009.

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *ECML/PKDD*, pages 148–163. Springer, 2010. doi: 10.1007/978-3-642-15939-8_10.

Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. A survey of noise reduction methods for distant supervision. *AKBC 2013 - Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, Co-located with CIKM 2013*, (Section 3):73–77, 2013. doi: 10.1145/2509558.2509571.

Yu Ming Shang, He Yan Huang, Xian Ling Mao, Xin Sun, and Wei Wei. Are noisy sentences useless for distant supervised relation extraction? *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pages 8799–8806, 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i05.6407.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. pages 4222–4235, 2020. doi: 10.18653/v1/2020.emnlp-main.346.

Alisa Smirnova and Philippe Cudré-Mauroux. Relation extraction using distant supervision: A survey. *ACM Comput. Surv.*, 51(5), nov 2018. ISSN 0360-0300. doi: 10.1145/3241741. URL https://doi.org/10.1145/3241741.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. *ACL*, pages 2895–2905, 2019. doi: 10.18653/v1/p19-1279. URL https://arxiv.org/pdf/1906.03158.pdf.

George Stoica, Emmanouil Antonios Platanios, and Barnab'as P'oczos. Re-tacred: Addressing shortcomings of the tacred dataset. In *AAAI*, 2021.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465, 2012.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM (CACM)*, 57(10):78–85, 2014.

Zhi Xiu Ye and Zhen Hua Ling. Distant supervision relation extraction with intra-bag and inter-bag attentions. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:2810–2819, 2019. doi: 10.18653/v1/n19-1288.