algorithms

MDPI

Article

Metamorphic Testing of Relation Extraction Models

Yuhe Sun [1], Zuohua Ding [1], Hongyun Huang [2], Senhao Zou [1] and Mingyue Jiang [1,*]

[1] School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China
[2] Library Multimedia Big Data Center, Zhejiang Sci-Tech University, Hangzhou 310018, China
[*] Correspondence: mjiang@zstu.edu.cn

Abstract: Relation extraction (RE) is a fundamental NLP task that aims to identify relations between some entities regarding a given text. RE forms the basis for many advanced NLP tasks, such as question answering and text summarization, and thus its quality is critical to the relevant downstream applications. However, evaluating the quality of RE models is non-trivial. On the one hand, obtaining ground truth labels for individual test inputs is tedious and even difficult. On the other hand, there is an increasing need to understand the characteristics of RE models in terms of various aspects. To mitigate these issues, this study proposes evaluating RE models by applying metamorphic testing (MT). A total of eight metamorphic relations (MRs) are identified based on three categories of transformation operations, namely replacement, swap, and combination. These MRs encode some expected properties of different aspects of RE. We further apply MT to three popular RE models. Our experiments reveal a large number of prediction failures in the subject RE models, confirming that MT is effective for evaluating RE models. Further analysis of the experimental results reveals the advantages and disadvantages of our subject models and also uncovers some typical issues of RE models.

Keywords: relation extraction; metamorphic testing; metamorphic relation; quality evaluation; testing and validation

check for updates

Citation: Sun, Y.; Ding, Z.; Huang, H.; Zou, S.; Jiang, M. Metamorphic Testing of Relation Extraction Models. *Algorithms* **2023**, *16*, 102. https://doi.org/10.3390/a16020102

Academic Editor: Frank Werner

Received: 6 December 2022
Revised: 19 January 2023
Accepted: 4 February 2023
Published: 10 February 2023

1. Introduction

Relation extraction (RE) is a fundamental task of natural language processing (NLP) and is also an important sub-task of information extraction. The ultimate goal of RE is to extract the relation between some entities according to a given text. Specifically, given a text and two entities (that appear in the text) as inputs, RE predicts the semantic relation (if any) holding between the two entities. Formally, given a triplet of (s, e_h, e_t), RE determines which relation $r \in \mathcal{R}$ holds between e_h and e_t in s or indicates "no_relation" (ϕ). Here, $\mathcal{R}$ is a predefined relation set which contains a variety of person-oriented or organization-oriented relation types, such as "per:children" (indicating that the tail entity e_t is the child of the head entity e_h) and "org:alternate_names" (representing that e_t is an alias of e_h). Furthermore, RE supports the construction of knowledge graph [1] and further facilitates a series of downstream applications, such as question answering systems [2], search engines [3], sentiment analysis [4], and text summarization [5].

In light of its importance, various deep learning models have been developed for RE, including models based on attention mechanisms [6], models combining external information sources [7], and pretrained language models [8]. Although many efforts have been made to advance RE models, the quality of the state-of-the-art RE models is still far from perfect [9]. For example, RE models may rely on shallow heuristics instead of learning to perform the intended task. As a result, although these models are reported to be effective on a dataset, they may perform poorly on data beyond the dataset [10]. Additionally, some RE models have been reported to have over-reliance on entity information [11], and some models even have difficulties in distinguishing relations with similar types of signatures [12]. What is worse, some models have been shown to ignore the information expressed by the context semantics and thus fail to understand the context correctly [13]. Therefore, there is an urgent need to evaluate RE models properly.

Unfortunately, evaluating RE models is non-trivial. Existing studies mainly apply the dataset-based evaluation method. One critical issue in this method lies in the reliance on ground truth labels, which always require tedious human annotations. Furthermore, low-quality annotations may in turn affect the evaluation of RE models [13,14]. This situation is known as the oracle problem [15] in software testing. On the other hand, dataset-based evaluation reports the accuracy of a model with respect to the evaluation dataset, which is insufficient to reveal the characteristics of the model in terms of various aspects relating to RE.

To address these issues, in this study, we propose applying the technique of metamorphic testing (MT) to RE. MT [16,17] is a property-based testing technique that is well known for alleviating the oracle problem. The key component of MT is metamorphic relation (MR), which encodes the properties of the target system via the relationship among multiple related inputs and outputs. Based on this, MT checks the relations among the related inputs and their outputs against some MRs, and thus it can be conducted without using oracles. Moreover, the testing results of MT can be interpreted with respect to the MR and accordingly reflect the performance of the target system in terms of the relevant properties. Overall, MT-based evaluation for RE can not only alleviate the oracle problem but also help to better understand the characteristics of RE models from various viewpoints.

To apply MT to RE models, we identify a total number of eight MRs. These MRs focus on different aspects of the task of relation extraction. We further demonstrate the effectiveness of MT by conducting experiments on three RE models. Our experiments report a large number of prediction failures in the subject RE models, and the experimental analysis also reveals the advantages and disadvantages of three subject RE models concerning different MRs. To summarize, this paper makes three major contributions:

- We propose applying MT to RE models. Our method can evaluate RE models without using ground truth labels.
- We design eight MRs for RE, each of which focuses on one specific property of the RE task. These MRs support the application of MT in RE and also contribute to the investigation and understanding of the characteristics of RE models.
- We conduct experiments on three RE models, demonstrating the feasibility and effectiveness of MT in evaluating RE models. Further analysis of the experimental results reveals the characteristics of the subject RE models and also uncovers typical issues for RE models.

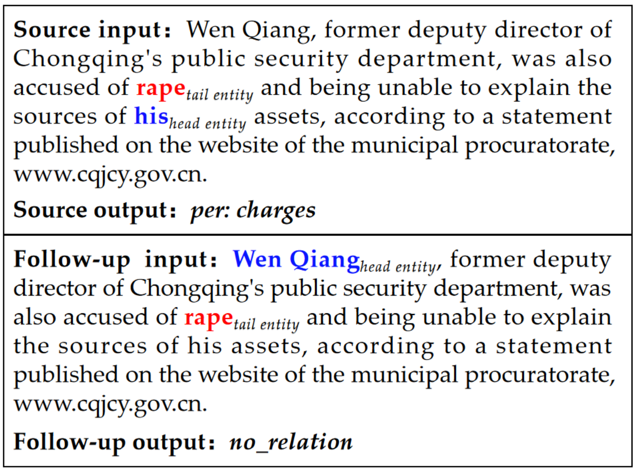
The remainder of this paper is organized as follows. Section 2 introduces the technique of metamorphic testing. The details of our approach are presented in Section 3. Our experimental set-up is presented in Section 4, and the experimental results are reported and analyzed in Section 5. Threats to the validity of this study are articulated in Section 6. Section 7 discusses the related work, and Section 8 concludes the present study.

2. Metamorphic Testing

Metamorphic testing (MT) [16,17] is a property-based testing technique that is well known for its ability to alleviate the oracle problem [16,18]. Instead of verifying the correctness of the outputs of individual program inputs, MT checks the relations among multiple inputs and outputs against some properties, which are known as metamorphic relations (MRs). Because of this, MT can be conducted without using oracles.

Generally, when applying MT, the program under test f will be executed at least twice. On one hand, f is executed with a source input X, yielding the source output $f(X)$. On the other hand, follow-up inputs X' will be constructed from X according to the given MR. Then, f is run with X', and the follow-up output $f(X')$ can be collected. Note that a group of X and X' is called a metamorphic test group (MG) of the MR. As a reminder, although an MG normally consists of a source input and a follow-up input, it can also contain multiple source or follow-up inputs. Lastly, the relation among X, X', $f(X)$, and $f(X')$ is checked against the given MR, reporting an MR violation or satisfaction.

Consider the example shown in Figure 1. For a given input (for example, the source input shown in Figure 1), it is difficult to judge the correctness of the prediction results without human effort. Nevertheless, by considering the property of RE that "replacing a head or tail entity with its coreferential entity should not affect the prediction result of RE" and treating this input as a source input, a relevant follow-up input can be constructed. In this example, the entity "his" in the source input is replaced by another entity, "Wen Qiang", with the same referential meaning. Accordingly, the relations predicted for the source and follow-up inputs are expected to be identical. As a result, the inconsistency among the actual prediction results "per: charges"and "no_relation" is detected, indicating that the RE model fails to correctly predict the relation between at least one pair of entities.



Source input: Wen Qiang, former deputy director of Chongqing's public security department, was also accused of **rape**$_{\textit{tail entity}}$ and being unable to explain the sources of **his**$_{\textit{head entity}}$ assets, according to a statement published on the website of the municipal procuratorate, www.cqjcy.gov.cn.

Source output: *per: charges*

Follow-up input: **Wen Qiang**$_{\textit{head entity}}$, former deputy director of Chongqing's public security department, was also accused of **rape**$_{\textit{tail entity}}$ and being unable to explain the sources of his assets, according to a statement published on the website of the municipal procuratorate, www.cqjcy.gov.cn.

Follow-up output: *no_relation*

Figure 1. A motivating example. The head entity is marked in blue, and the tail entity is marked in red.

Although MT was originally proposed for software verification, it has been extended to quality assessment [19] and system comprehension [20]. Aside from that, MT has seen successful applications in other domains beyond testing, such as fault location [21], program repair [22], and program proving [23]. Recently, we have also seen MT show promising results in various NLP tasks, such as question answering systems [24–26], sentiment analysis [27,28], and natural language inference [29]. In this paper, we apply MT to relation extraction in order to break the dependency on the labels, which alleviates the oracle problem, as well as gain a deeper understanding of the strengths and weaknesses of relation extraction models in terms of the related properties via MRs.

3. Approach

In this section, we introduce our approach of evaluating the task of relation extraction using MT. We first explain the process of performing MT on RE and then present the details of the designed MRs.

3.1. Applying Metamorphic Testing to RE

RE is a fundamental task of NLP which aims to identify the relation between two entities in a given context. Formally, let $\mathcal{R}$ be an RE model. Suppose that s is a sentence and e_h and e_t are two entities (the head and tail entities, respectively) involved in s. $\mathcal{R}$ takes a triplet of (s, e_h, e_t) as an input and yields the prediction relation r based on a predefined relation set; that is, $r = \mathcal{R}(s, e_h, e_t)$. For example, consider the sentence "Obama was born in Honolulu", with a head entity "Obama" and a tail entity "Honolulu". An RE model predicts the relation between them as "per:city_of_birth". Note that the prediction of r

requires the understanding of s as well as the roles of e_h and e_t in the context described by s, which is thus non-trivial.

In this paper, we apply MT to evaluate RE models. An overview of the approach is presented in Figure 2. We first identified a list of MRs based on the necessary and expected properties of the RE task. Each MR was defined as $MR_i = (t_i, r_i)$, where t_i is a transformation operation to be conducted on the source input and r_i is the corresponding output relation. Then, we constructed a set of MGs based on MRs; that is, for a source input e_i satisfying the MR_i condition, we used the transformation t_i to obtain the follow-up input e'_i. After that, we ran each MG (e_i, e'_i) on RE models and collected the prediction results as source and follow-up outputs. Finally, we checked the source and follow-up outputs against relevant r_i values. If r_i is violated, then failures of the RE model are revealed. Specifically, in this study, three categories of transformation operations were employed in our method, namely replacement, swap, and combination. We present the concrete transformation operation and corresponding output relationship of individual MRs in Table 1.

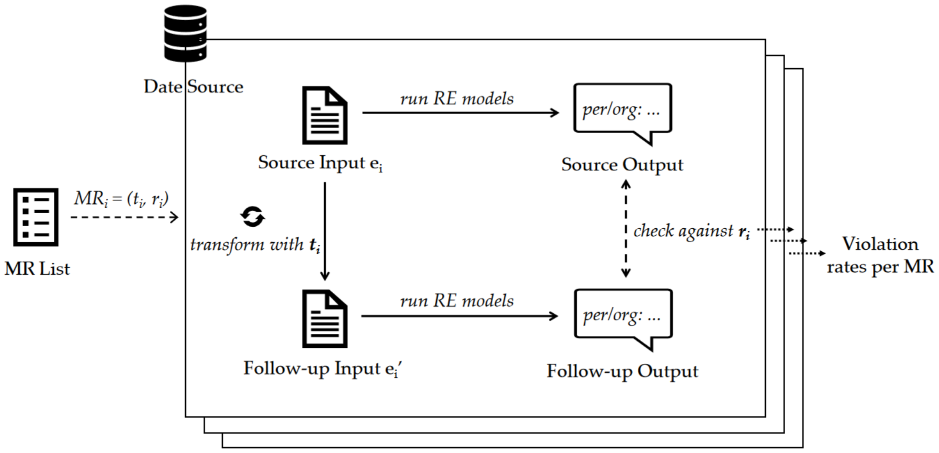


Figure 2. Overview of how metamorphic testing (MT) is applied to evaluate RE models.

Table 1. Transformation operation and output relationship of MRs.

		MR Description	
Category	**Name**	**Transformation Operation**	**Output Relationship**
Replacement	MR-R$_1$	Replacing head (tail) entity with the same type of entity	Identical
	MR-R$_2$	Replacing head (tail) entity with the coarser-grained type of entity	Consistent with the transformation of entity granularity
	MR-R$_3$	Replacing head (tail) entity with co-related entity having different entity type	Identical
	MR-R$_4$	Replacing head (tail) entity with its coreferential entity	Identical
Swap	MR-S$_1$	Swaping head and tail entities in symmetric relations	Identical
	MR-S$_2$	Swaping head and tail entities in antisymmetric relations	Opposite
Combination	MR-C$_1$	Combining two pairs of entities sharing the same head entity on multiple source inputs	Identical with the second source input
	MR-C$_2$	Combining two pairs of entities sharing the same tail entity on multiple source inputs	Identical with the second source input

3.2. Metamorphic Relations for RE

Designing effective MRs is a critical step in MT. In this study, we design MRs for RE by considering various different input operations as well as varying output relationships. Each MR perturbs the source inputs ($t_s = (s^s, e^s_h, e^s_t)$) to generate follow-up inputs (t_f) and also describes the expected relation among the relevant source and follow-up outputs ($\mathcal{R}(t_s)$ and $\mathcal{R}(t_f)$).

3.2.1. MRs with the Replacement Operation

Previous studies have revealed that neural network models have poor performance when confronted with randomly permuted contexts [25]. For RE, **its prediction heavily relies on the information taken by the input entities (including entity names and entity types)** [11], while the capability of different RE models for handling different entities and relations is still unknown. According to this, **we design MRs to investigate how consistently RE can handle different entities** by replacing the head or tail entity with some candidate entities; that is, we leverage the replacement operation to generate test groups and investigate the capability of RE models. The replacement operation, which constructs follow-up inputs by changing some parts of the source inputs with appropriate replacements, is commonly adopted in the practice of MT [21]. Our replacement operation can be conducted at different levels by considering the characteristics of different entities.

MR-R$_1$: Replacement with the same type of entity.

The same type of entity expresses the same category of information. Consider, for example, a set of entities such as French, American, and Chinese, which have the same entity type: NATIONALITY. Although they take specific information, they all describe the information of nationality. Therefore, when replacing the head or tail entity with another one having the same entity type, RE should provide the same prediction.

Suppose that there is an entity e'_h (e'_t) whose entity type is identical to that of e^s_h (e^s_t). The follow-up input is constructed as (s^f, e'_h, e^s_t) or (s^f, e^s_h, e'_t), where s^f is generated from s^s by replacing e^s_h (e^s_t) with e'_h (e'_t). Then, $\mathcal{R}(t_s)$ and $\mathcal{R}(t_f)$ are expected to be identical.

An example MG of MR-R$_1$ is shown in the first row of Table 2. In the given sentence "French filmmaker Claude dies at 80", RE predicts the relation between the entities "French" and "Claude" as "per:origin". After the tail entity "French" is replaced by "American", where both "French" and "American" have the entity type NATIONALITY, the prediction result remains unchanged.

To obtain candidate entities of the same type as the original entity, we applied the named entity recognition tool Stanford NER (https://nlp.stanford.edu/software/CRF-NER.html, accessed on 10 July 2022) to analyze the sentences, yielding sets of candidate entities of various different entity types. Then, for a given input, we randomly selected a candidate entity with the designated type and used it as a replacement.

Table 2. Sample MGs of MRs with replacement and swap operations.

MR	Source Input	Source Output	Follow-Up Input	Follow-Up Output
MR-R$_1$	French$_{tail}$ filmmaker Claude$_{head}$ dies at 80.	per: origin	American$_{tail}$ filmmaker Claude$_{head}$ dies at 80.	per: origin
MR-R$_2$	Richard$_{head}$ was born in San Francisco$_{tail}$.	per: city_of_birth	Richard$_{head}$ was born in the United States$_{tail}$.	per: country_of_birth
MR-R$_3$	Jupp$_{head}$, 46$_{tail}$, works in the lab.	per: age	Jupp$_{head}$, 46-years-old$_{tail}$, works in the lab.	per: age
MR-R$_4$	John is a father, he$_{tail}$ loves his child Mary$_{head}$.	per: parents	John$_{tail}$ is a father, he loves his child Mary$_{head}$.	per: parents
MR-S$_1$	Lily$_{head}$ is Mary$_{tail}$'s sister.	per: siblings	Lily$_{tail}$ is Mary$_{head}$'s sister.	per: siblings
MR-S$_2$	John$_{head}$ is Mary$_{tail}$'s father.	per: children	John$_{tail}$ is Mary$_{head}$'s father.	per: parents

The head entity is marked in blue, and the tail entity is marked in red.

MR-R$_2$: Replacement with the coarser-grained type of entity.

Some entity types describe the same category of information but exhibit varying granularity. For example, both the entity “San Francisco” (of the type CITY) and the entity “the United States” (of the type COUNTRY) describe information referring to a location. Nevertheless, the scope represented by the latter is broader, and it includes the scope denoted by the former. Moreover, it can be observed that RE assigns different labels expressing similar relations for such entities. For instance, “per:city_of_birth” and “per:country_of_birth” both describe the relation between a PERSON entity and an entity representing a location but with different granularities. Accordingly, for such types of relations, if we replace a head or tail entity with entities of the relevant coarser-grained type, RE should consistently reflect this change in the resulting prediction.

Suppose that $\mathcal{R}(t_s)$ is a relation with varying granularity and e'_h (e'_t) is an entity of a coarser-grained type compared with e^s_h (e^s_t). Then, $t_f = (s^f, e'_h, e^s_t)$ or (s^f, e^s_h, e'_t), where s^f is constructed from s^s by replacing e^s_h (e^s_t) with e'_h (e'_t). In addition, $\mathcal{R}(t_s)$ and $\mathcal{R}(t_f)$ express the similar relation but with granularities encoded by e^s_h (e^s_t) and e'_h (e'_t), respectively.

Consider the example shown in the second row of Table 2. After the replacement of “San Francisco” with “the United States” in the input, the prediction result accordingly changed from “per:city_of_birth” to “per:country_of_birth”.

To properly implement MR-R$_2$, we applied the Geopy API (https://github.com/geopy/geopy, accessed on 27 June 2022) to obtain the relevant state and country according to the name of a city. Then, for a given input containing entities of the type CITY, we replaced the CITY entity with the relevant country or state entity.

MR-R$_3$: Replacement with co-related entities having different entity types.

Due to the varying ways of expressing a natural language sentence, different types of entities may be able to convey similar meanings in the context of the sentence. Such entities, although they have different entity types, are co-related in terms of their semantic meanings. Hence, replacing a head or tail entity with its co-related entity should not affect the prediction result of RE.

Suppose that there is an entity e'_h (e'_t) which is co-related with e^s_h (e^s_t). $t_f = (s^f, e'_h, e^s_t)$ or (s^f, e^s_h, e'_t), where s^f is constructed from s^s by replacing e^s_h (e^s_t) with e'_h (e'_t). Then, $\mathcal{R}(t_s)$ and $\mathcal{R}(t_f)$ are expected to be identical.

An example MG of MR-R$_3$ is reported in the third row of Table 2. As can be observed, by replacing the entity “46” of the type NUMBER (the tail entity in t_s) with an entity “46-years-old” of the type DURATION (the tail entity in t_f), the relations between the head entity and the tail entity in t_s and t_f are identical.

Implementing MR-R$_3$ requires constructing a collection of candidate co-related entities for different types of entities. For this, we searched for different entity types that were semantically co-related across all sentences, where the same entity mentions may be labeled with different entity types as co-related entities. Then, for a given input, we randomly selected a candidate co-related entity of a different type from the original entity as a replacement.

MR-R$_4$: Replacement with coreferential entities.

In a natural language sentence, different entity mentions may refer to the same real-world entity information. This phenomenon is known as coreference resolution, which is a common task in NLP and critical to the success of RE [30]. Naturally, replacing a head or tail entity with its coreferential entity should not affect the prediction result of RE.

Suppose that there is an entity e'_h (e'_t) in the given sentence s^s that refers to the same information as e^s_h (e^s_t). Then, t_f is constructed as (s^s, e'_h, e^s_t) or (s^s, e^s_h, e'_t), which takes the entity e'_h (e'_t) as the head (tail) entity. Thus, $\mathcal{R}(t_s)$ and $\mathcal{R}(t_f)$ are expected to be identical.

It is common that a noun and a relevant pronoun are coreferential. For example, consider the example MG shown in the fourth row of Table 2, where “John” and “he” both refer to the same person in the given sentence. Therefore, after the replacement, the prediction result should remain the same.

To implement MR-R$_4$, we applied NeuralCoref (https://github.com/huggingface/neuralcoref, accessed on 2 July 2022), a widely used neural network-based coreference resolution tool, to find all coreferential entities (if any) of a given entity in a sentence. An entity may have multiple coreferential entities in a sentence, and the one with the highest coreference resolution score (correlation coefficient) was selected as a candidate replacement in our experiments.

3.2.2. MRs with the Swap Operation

In logic, a binary relation can be either symmetric or asymmetric. The property of symmetry can be found in many application systems. For example, in the Google Maps navigation service, symmetry means that exchanging the destination and origin should return a route with a similar cost (in terms of time or distance) [20]. For the face recognition functions of Facebook, symmetry refers to flipping the image to cause a mirror image, which should not affect the face recognition results because faces are usually approximately symmetrical [31]. However, prior studies revealed that none of the tested systems satisfied these symmetry properties. Inspired by these, we intend to investigate the symmetry satisfaction of RE models. In the context of RE, it has been observed that some relations are symmetrical, for which the relation between entities A and B is the same as that between B and A. Nevertheless, some relations are antisymmetric such that the relation between A and B is reversed to that between B and A. As a result, swapping the head and tail entities should have different impacts on the prediction results of RE.

MR-S$_1$: Swap entities with symmetrical relations.

There are some symmetric relations between two entities, such as a spouse relationship or sibling relationship. In these situations, the two entities are equally treated, and thus their order should not affect the prediction of the relation. In Table 2, an illustrative example is given in the fifth row. As can be observed, in the context of the given sentence, the "per:sibling" relation between "Lily" and "Mary" is the same as that between "Mary" and "Lily".

Suppose that $\mathcal{R}(t_s)$ is a symmetrical relation. Then, t_f is constructed by swapping the head and tail entities such that $t_f = (s^s, e^s_t, e^s_h)$. Thus, $\mathcal{R}(t_s)$ and $\mathcal{R}(t_f)$ are expected to be identical.

MR-S$_2$: Swap entities with antisymmetric relations.

The relation between two entities can be antisymmetrical. In this case, different orders between the head and tail entities will lead to opposite relations. For example, as shown in the sixth row of Table 2, the relation between "John" and "Mary" is "per:children", while the relation between "Mary" and "John" is "per:parents".

Suppose that $\mathcal{R}(t_s)$ is an antisymmetric relation. Then, t_f is constructed by swapping the head and tail entities such that $t_f = (s^s, e^s_t, e^s_h)$. Thus, $\mathcal{R}(t_s)$ and $\mathcal{R}(t_f)$ are expected to be opposites.

To identify the symmetric and antisymmetric relations, we manually classified the relation types. Then, for each input satisfying the relation type, we treated the tail (head) entity in the source input as the head (tail) entity of the follow-up input.

3.2.3. MRs with the Combination Operation on Multiple Source Inputs

In a sentence, it is common that there exist more than two entities. Moreover, different pairs of entities exhibiting varying relations may have common entities. In particular, entities with a symmetrical relation play equal roles in a sentence. Motivated by this, it is feasible to combine two pairs of entities (one of which has the symmetric relation) to construct a new pair of entities. This type of MR involves two source inputs, ($t_{s1} =$

$(s^s, e^s_{h1}, e^s_{t1})$ and $t_{s2} = (s^s, e^s_{h2}, e^s_{t2})$), and one follow-up input, the example MGs for which are shown in Table 3.

Table 3. Sample MGs of MRs with the combination operation.

MR	First Source Input	First Source Output	Second Source Input	Second Source Output	Follow-Up Input	Follow-Up Output
MR-C$_1$	Peterson$_{head}$ leaves behind his wife, Kelly$_{tail}$, and their daughter Celine.	per:spouse	Peterson$_{head}$ leaves behind his wife, Kelly, and their daughter Celine$_{tail}$.	per:children	Peterson leaves behind his wife, Kelly$_{head}$, and their daughter Celine$_{tail}$.	per:children
MR-C$_2$	Liu Mingkang, chairman of the China Banking Regulatory Commission$_{tail}$ (CBRC$_{head}$), was the representative of the mainland to host a meeting.	org:alternate_names	Liu Mingkang$_{head}$, chairman of the China Banking Regulatory Commission$_{tail}$ (CBRC), was the representative of the mainland to host a meeting.	per:employee_of	Liu Mingkang$_{head}$ chairman of the China Banking Regulatory Commission (CBRC$_{tail}$), was the representative of the mainland to host a meeting.	per:employee_of

The head entity is marked in blue, and the tail entity is marked in red.

MR-C$_1$: Combining two pairs of entities sharing the same head entity.

We first focus on the combination of two source inputs sharing the same head entity; that is, $e^s_{h1} = e^s_{h2}$.

Suppose that $e^s_{h1} = e^s_{h2}$ and $\mathcal{R}(t_{s1})$ is a symmetrical relation. Then, t_f is constructed as $(s^s, e^s_{t1}, e^s_{t2})$, which takes the tail entity of t_{s1} as the head entity and uses the tail entity of t_{s2} as the tail entity. Thus, $\mathcal{R}(t_f)$ is expected to be identical to $\mathcal{R}(t_{s2})$.

MR-C$_2$: Combining two pairs of entities sharing the same tail entity.

Similarly, the combination can also be performed on two pairs of entities sharing the same tail entity.

Suppose that $e^s_{t1} = e^s_{t2}$ and $\mathcal{R}(t_{s1})$ is a symmetrical relation. Then, t_f is constructed as $(s^s, e^s_{h2}, e^s_{h1})$, which takes the head entity of t_{s2} as the head entity and uses the head entity of t_{s1} as the tail entity. Thus, $\mathcal{R}(t_f)$ is expected to be identical to $\mathcal{R}(t_{s2})$.

Its implementation process is as follows. The first source input t_{s1} is known. We need to construct the second source input t_{s2} and build t_f from t_{s1} and t_{s2}. First, we create a relation set $\{R_1, R_2\}$ with the conditions for combination, where R_1 has symmetry. Then, we obtain the type information of the tail entity in R_2 and apply the named entity recognition tool Stanford NER to identify all entities that are consistent with the tail entity type in R_2, forming a set E. Next, we remove the entities e^s_{h1} and e^s_{t1} (if any) from the set E to obtain E$'$. We construct a candidate source input $t_{cs} = (s^s, e^s_{h1}, e^s_{t2})$ for each entity e^s_{t2} in set E$'$ and obtain $\mathcal{R}(t_{cs})$ via RE. If $\mathcal{R}(t_{cs}) \in R_2$, then the candidate source input satisfies the condition and is used as the second source input $t_{s2} = (s^s, e^s_{h2}, e^s_{t2})$. Finally, t_f is constructed as $(s^s, e^s_{t1}, e^s_{t2})$, which is generated from the original source input t_{s1} and the constructed source input t_{s2}.

4. Experimental Set-Up

This section presents our experimental set-up, including our subject RE models, data source, details of MR implementation, and the way of constructing MGs for individual MRs.

4.1. Subject RE Models

We conducted experiments on three RE models: BERT$_{\text{EM}}$+MTB, LUKE, and NCB. BERT$_{\text{EM}}$+MTB uses BERT [32] to represent textual relations and is a task-agnostic relation extraction model [33]. The core idea of the model is that two relational representations should be similar if they contain the same entity pair. The model uses a large amount of unsupervised data, and additional matching the blanks (MTB) tasks are added in the BERT pretraining process, which improves the performance of relation extraction in the pretraining stage.

LUKE [8] is a transformer-based language model that is pretrained on large-scale text corpora and knowledge graphs while using entity information as an additional input embedding layer. It regards words and entities in the text as independent tokens and finally outputs the context-processed entity representation. Moreover, LUKE uses an entity-aware self-attention mechanism, which considers the category of the token (word or entity) when calculating the attention score.

Noise-robust co-regularization BERT-large (NCB) [34] proposes a co-regularization framework for entity-centric information extraction, which consists of several neural models with the same structure but different parameter initializations to prevent overfitting to noisy labels. It can efficiently learn supervised models on noisy datasets without any additional learning resources.

The details of three RE models are reported in Table 4. Among them, the latter two models achieved SOTA performance on the RE task (https://paperswithcode.com/, accessed on 10 June 2022.)

Table 4. Information of RE models.

Model	Year	Backbone
BERT$_{\text{EM}}$+MTB	2019	BERT
LUKE	2020	Transformer
NCB	2021	NCB

We implement BERT$_{\text{EM}}$+MTB using OpenNRE (https://github.com/thunlp/OpenNRE, accessed on 20 June 2022), an open-source and extensible toolkit launched by the Natural Language Processing Group at the Department of Computer Science and Technology, Tsinghua University (THUNLP), which provides a unified framework for implementing relation extraction models. Based on huggingface's Transformers platform (https://github.com/huggingface/transformers, accessed on 20 June 2022), we implemented the LUKE model. For the NCB model, since only the source code was available, we locally trained it using the train split of the TACRED dataset (https://nlp.stanford.edu/projects/tacred, accessed on 20 June 2022) and then conducted evaluations of the resulting models.

4.2. Data Source

The TACRED dataset [6] was used to prepare the source inputs in our experiments. TACRED is the largest and most widely used RE dataset, covering 42 relation types and containing 106,264 sentences. Sentences are annotated with person-oriented or organization-oriented relation types (e.g., "per:title", "org:founded", and "no_relation" for negative examples), and each sentence in the dataset has only one relation label [6,13]. In addition, the entity types of the head and tail entities of each sentence were also identified, including a total of 17 entity types. As a reminder, our source inputs from the TACRED dataset were triples of a sentence and head and tail entities rather than annotated relation labels, since our method does not require true labels during evaluation.

4.3. Construction of MGs

In the process of constructing MGs, each MR has a **varying number of valid source inputs** due to different preconditions and operations used for constructing follow-up inputs. Meanwhile, for partial MRs, the construction of follow-up inputs depends on the source outputs. Therefore, for each MR, we selected the source inputs by checking whether the sentence head (tail) entity triples from the TACRED dataset and the relation labels of the source outputs satisfied the preconditions of the MR, and those that satisfied the preconditions constituted the set of source inputs. When the source inputs were ready, we constructed the corresponding follow-up inputs against the relevant MR.

To ensure the validity of the generated follow-up inputs, for each MR, we randomly sampled 100 generated follow-up inputs for manual inspection. We examined the syntactic validity of the generated follow-up inputs in terms of grammar and semantics. Overall, we found that less than 8% of the follow-up inputs had minor errors. This showed a considerably low error rate compared with the 22–52% [13,35] label error rate in the TACRED dataset.

Because different RE models may predict varying relations for the same sentence and head (tail) entity triples, the MG sets constructed by different RE models may be different, even with the same MR. Table 5 summarizes the average number of MGs for all MRs. Based on all our MRs, a total of over 32,400 MGs were used to evaluate each subject RE model.

Table 5. Number of MGs.

MR	**No. of MGs**
MR-R$_1$	8198
MR-R$_2$	4350
MR-R$_3$	3866
MR-R$_4$	2605
MR-S$_1$	2731
MR-S$_2$	7584
MR-C$_1$	1650
MR-C$_2$	1435
Total	32,419

5. Results and Analysis

This section presents our experimental results on the three RE models. We first report the overall results of MT. Then, we compare the performances of our subject RE models, revealing their strengths and weaknesses. With further investigation of the evaluation results, we list some representative failures detected and reveal several types of issues exposed by the RE models. Finally, we summarize our observations on the common characteristics of the RE models.

5.1. Overall Results of MT

To evaluate the RE models, in this paper, we adopted a common MT evaluation metric, namely the violation rate (VR). The VR measures the extent to which an RE model's behavior deviates from the behavior encoded by an MR. Given an MR and a RE model, let n be the number of MGs used to evaluate this RE model and x be the number of MGs that violated the MR. Then, the VR value of the MR for this RE model is $\frac{x}{n}$, which represents the percentage of MGs that revealed MR violations. The violation rate is in the range [0, 1], where a higher (lower) VR value indicates that the model had worse (better) performance on the relevant MR. In particular, a violation rate of zero means that no violation of the relevant MR was revealed among all used MGs, suggesting that the RE model is likely to perform well according to the MR.

We performed MT on each of the eight proposed MRs on the three subject RE models, and MT detected a total of **23,136** MR violations. The results show that all eight MRs effectively revealed a considerable number of violations in every subject RE model. We summarize the overall quantitative results of our evaluation in Table 6. In this table, we report the VR values of each MR for individual RE models. In addition, the overall VR value for each RE model across all MRs (as shown in the Overall row) and the average VR value for each MR across all RE models (as shown in the Average column) were also calculated.

Table 6. Violation rates (%) of RE models for different MRs.

RE Model	BERT$_{\text{EM}}$+MTB	LUKE	NCB	Average
MR-R$_1$	34.2	28.6	4.2	25.2
MR-R$_2$	26.8	19.8	15.6	20.4
MR-R$_3$	39.8	39.7	49.5	42.7
MR-R$_4$	20.6	18.4	19.9	19.6
MR-S$_1$	10.5	5.5	9.2	8.4
MR-S$_2$	20.4	17.7	26.0	21.3
MR-C$_1$	24.5	19.8	28.4	24.2
MR-C$_2$	26.3	18.2	30.1	24.9
Overall	25.9	22.5	21.7	23.6

Table 6 shows that all eight MRs successfully revealed the failure of each RE model, since all VR values were greater than zero. Consider, for example, that the VR value of NCB with respect to MR-R$_3$ was 49.5%. This VR value indicated that among all MGs of MR-R$_3$ that were applied to evaluate NCB, 49.5% revealed MR violations. According to the data reported by the TACRED dataset, the test F1 scores of BERT$_{\text{EM}}$+MTB, LUKE, and NCB are 71.5%, 72.7%, and 73.0%, respectively (https://paperswithcode.com/sota/relation-extraction-on-tacred, accessed on 20 July 2022), where the value of "$1-\text{F1}$" can be seen as the rate of samples that are inconsistent with a given reference label in the dataset. We display the VR values for all MRs and the value of "$1-\text{F1}$" for each RE model in Figure 3. Compared with these, it can be seen that **at least two MRs detected many more incorrect cases for all models**. These results demonstrate that **our method delivers fairly good performance in revealing the erroneous behavior of the subject RE models**.

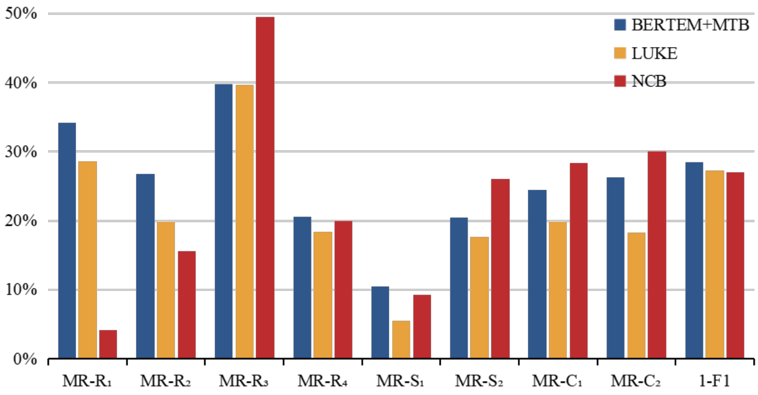


Figure 3. Comparison of F1 scores and violation rates on different MRs for three RE models.

In addition, it can also be found from Table 6 that all three RE models typically showed varying VR values for the same given MR. For example, the VR values of MR-S$_2$ on the three RE models were 20.4%, 17.7%, and 26.0%, indicating that different RE models performed differently on MR-S$_2$. In addition, for each RE model, there was also a clear numerical variation in the VR values relative to different MRs. For instance, the VR values were 28.6%, 19.8%, 39.7%, 18.4%, 5.5%, 17.7%, 19.8%, and 18.2% on LUKE, and the numerical changes in the VR values indicate that LUKE performed differently for varying MRs. In other words, every RE model violated different MRs with varying VR values, and different RE models also violated the same MR with varying VR values. This result further suggests that **different MRs have varying abilities to reveal defects or abnormal behaviors and can reflect the shortcomings of each RE model in the corresponding properties**. Specifically, the three MRs that only focused on entity generalization of RE (that is, MR-R$_1$, MR-R$_2$ and MR-R$_3$) had relatively high VR values compared with the remaining MRs. **These results not only demonstrate the strong ability of this category of MR to detect failures but also further indicate the challenge of RE in enhancing the generalization capability of entities**.

5.2. Performance Comparison of Subject RE Models

We first compare the overall performance of the three RE models. The Overall row of Table 6 reports the overall VR values across all MRs for individual RE models. A lower VR value indicates a better overall performance. According to the results, NCB (*VR* = 21.7%) exhibited the best overall performance, which was followed by LUKE (*VR* = 22.5%). BERT$_{\text{EM}}$+MTB had the worst overall performance with respect to all of our MRs (*VR* = 25.9%).

We next focus on individual RE models to understand their strengths and weaknesses. First, from Figure 3, we were disappointed to find that BERT$_{\text{EM}}$+MTB did not perform the best with respect to any of the MRs.

Another model, LUKE, exhibited the lowest VR value on MR-R$_3$ (*VR* = 39.7%) and MR-R$_4$ (*VR* = 18.4%), indicating that this model performed better than the other models for handling co-related entities with different entity types as well as coreferential entities. Meanwhile, relatively lower VR values were observed for MR-S$_1$ (*VR* = 5.5%) and MR-S$_2$ (*VR* = 17.7%), suggesting that LUKE is less sensitive to the order of entities. In addition, LUKE outperformed the other models for both MR-C$_1$ and MR-C$_2$ (with the two lowest VR values of 19.8% and 18.2%, respectively). Therefore, it can be inferred that LUKE performs better in correctly capturing the relations between new entity pairs, which are combined from two pairs of entities with different relations.

Finally, for NCB, we found that the performance of the models varied widely for MR-R$_1$, MR-R$_2$, and MR-R$_3$, which all involved entity replacement. When the model achieved the best performance with the lowest VR value on both MR-R$_1$ (*VR* = 4.2%) and MR-R$_2$ (*VR* = 15.6%), NCB showed a large number of violations on MR-R$_3$ (*VR* = 49.5%) related to entity type changes. We consider that NCB may expose the potential problems of relying heavily on entity type information. To investigate further, we performed a white box analysis of NCB's source code. We learned that the model utilizes entity masking techniques that replace the head (subject) and tail (object) entities with their named entity types. For example, a short sentence such as "*Bill Gates founded Microsoft*" would become "[SUBJECT-PERSON] *founded* [OBJECT-ORGANIZATION]" after the entity mask. Hence, in this case, changes in entity types were extremely likely to incur prediction failures. This explains the large difference in model performance between MR-R$_1$ (entity type unchanged) and MR-R$_3$ (entity type changed), with both involving entity replacement. This also further illustrates the effectiveness of our evaluation of RE.

5.3. Analysis of the Detected Failure Examples

We explored the details of the violations to intuitively understand the issues revealed by our MRs. As shown in Table 7, we list some representative examples of prediction failures and, through further statistics and analysis of the violation examples, reveal several types of issues exposed by RE models.

Table 7. Typical prediction failures of RE models.

MR	RE Model	Source Input	Source Output	Follow-Up Input	Follow-Up Output	Correct Result
MR-R_3	NCB	$\text{US}_{type:\text{NATIONALITY}}$ actress Patricia Neal, winner of both Academy and Tony awards, died at her home...	per: origin	$\text{US}_{type:\text{COUNTRY}}$ actress Patricia Neal, winner of both Academy and Tony awards, died at her home...	per:countries_of_ residence	per: origin
MR-R_1	BERT_{EM} + MTB	Alessi said that she was $\text{Catholic}_{type:\text{RELIGION}}$ but that she had long ago lost her illusions.	per: religion	Alessi said that she was $\text{Muslim}_{type:\text{RELIGION}}$ but that she had long ago lost her illusions.	no_relation	per: religion
MR-R_4	NCB	Dunne was part of a famous family that also included his brother, novelist and screenwriter John Gregory Dunne; his brother's wife, author Joan Didion...	per: other_family	Dunne was part of a famous family that also included his brother, novelist and screenwriter John Gregory Dunne; his brother's wife, author Joan Didion...	per: spouse	per: other_family
MR-S_2	LUKE	It is unknown as of now whether or not Britney's mother Lynne, pregnant sister Jamie Lynn or brother Brian are on their way to LA.	per: parents	It is unknown as of now whether or not Britney's mother Lynne, pregnant sister Jamie Lynn or brother Brian are on their way to LA.	per: siblings	per: children
MR-C_2	BERT_{EM}+ MTB	World soccer chief Joseph Sepp Blatter is expected in Madagascar on Tuesday, the president of the Madagascan Football Federation -LRB-FMF-RRB-Ahmad said on Monday.	org: alternate_names	World soccer chief Joseph Sepp Blatter is expected in Madagascar on Tuesday, the president of the Madagascan Football Federation -LRB- FMF -RRB-Ahmad said on Monday.	no_relation	per: employee_of
		World soccer chief Joseph Sepp Blatter is expected in Madagascar on Tuesday, the president of the Madagascan Football Federation -LRB-FMF -RRB-Ahmad said on Monday.	per: employee_of			

The head entity is marked in blue, and the tail entity is marked in red.

(1) First, for the MR-R_3 with the highest VR values, we observed considerable prediction failures. As shown in the first row of Table 7, we were surprised to find that NCB gave inconsistent prediction results when the sentences and entities entered into the model were identical but only had the label of the entity type name changed. This further **confirms the issue of NCB's heavy reliance on entity type information** mentioned in the previous section.

(2) The second type of prediction failure was revealed by violating MR-R_1. An illustrative example is shown in the second row of Table 7. For substitutions between two entities, namely "Catholic" and "Muslim" with the same entity type RELIGION but different entity mentions (names), the RE model predicted different relations.
In addition, we found that most of the prediction failure cases in LUKE and BERT_{EM}+ MTB were due to replacing the original entity with entities that appeared less frequently in the TACRED dataset. (This issue was not exhibited in NCB because entities of the same type were represented in the same form after entity masking.) From this, we speculate that **RE models may overfit the training samples and thus only pass part of the test cases where entities appear frequently in the training samples** .
Therefore, we conducted further experiments on LUKE and BERT_{EM}+MTB. We divided the entities into two categories according to the frequency of the entity vocabulary (names) in the training samples of the dataset, one of which was high-frequency entity vocabulary while the other was low-frequency entity vocabulary. Then, we limited the replacement objects in MR-R_1 to these two different categories of entities; that is, we randomly selected one of the high-frequency entity vocabulary and the other of the low-frequency entity vocabulary to replace it. It was found that the model performance varied greatly, as shown in Table 8. Substitutions from high-frequency entity words yielded a low average VR value (VR = 16.4%), while substitutions from low-frequency entity words showed a high average VR value (VR = 62.5%). These variations indicate that the model was overfitting the training samples to some extent.

Table 8. Comparison of violation rates (%) of high-frequency entity vocabulary and low-frequency entity vocabulary replacement in MR-R_1.

RE Model	High-Frequency Entity Vocabulary	Low-Frequency Entity Vocabulary
$BERT_{EM}$+MTB	15.9	64.4
LUKE	17.8	60.8
Average	16.4	62.5

(3) Based on violations of MR-R_4, the failures of RE related to handling coreferential entities were detected. In the third row of Table 7, the entities "Dunne" and "his" have the same referential meaning. However, the RE model predicted a "per:other_family" relation between the entities "Joan Didion" and "his" but a "per:spouse" relation between "Joan Didion" and "Dunne". This reveals a prediction failure: replacing the original entity with its coreferential entity should not affect the relation between the entities.

(4) Based on violations of MR-S_2, the failures of RE in the face of an entity order swap in antisymmetric relations were detected. As shown in the fourth row of Table 7, the RE model predicted an antisymmetric relation "per:parents" between the entities "Lynne" and "Jamie Lynn". After exchanging the head and tail entities, it failed to successfully predict the opposite relation, namely *per:children*.

(5) Violations of MR-C_2 revealed another type of failure. The fifth row of Table 7 shows that the RE model predicted that the entity "Madagascan Football Federation" was an alias for the entity "FMF", and there was a "per:employee_of" relation between the entities "Madagascan Football Federation" and "Ahmad". However, the RE model failed to capture the "per:employee_of" relation between the entities "FMF" and "Ahmad".

5.4. Findings

We investigated the evaluation results and observed some common properties of the RE models, which are summarized below:

(1) *RE models are more sensitive to changes in entity type than changes in entity mentions (names).* Entity mentions (names) and entity types are important pieces of information for entities. Peng et al. [11] reported that RE models may improve model performance with some cues that entity mentions exhibit, while Rosenman et al. [36] also observed that RE models expose shallow heuristics in the type of candidate arguments. In this study, two MRs, MR-R_1 and MR-R_3, focused on the entity mentions and entity types of RE, respectively. As can be seen from Figure 3, the performances of the three RE models on MR-R_1 varied, but they all showed the highest VR value on MR-R_3, which indicates that the failure caused by the change in entity type information was widely revealed in all three models. The RE models showed poor robustness when facing changes in entity type information. From this perspective, our findings are consistent with existing observations that RE models suffer from overly dependence on entity types [12,36].

(2) *Compared with entity exchange in symmetric relations, RE models are more sensitive to the changes in entity order in antisymmetric relations.* In this study, the two MRs, namely MR-S_1 and MR-S_2, applied entity swap operations to symmetric and antisymmetric relations, respectively. To reveal whether symmetric and antisymmetric relations were more affected by changes in entity order, we examined the VR values of individual MRs of MR-S_1 and MR-S_2. The results are depicted in Figure 3. It was found that for every RE model, the VR value of MR-S_1 was lower than that of MR-S_2. These results indicate that RE models are more sensitive to the changes in entity order in antisymmetric relations than in symmetric relations. In other words, entity order perturbations in antisymmetric relations are more likely to incur prediction failures.

5.5. Discussion

In this study, we proposed a method to evaluate RE using MT, which breaks the dependence on human-annotated labels and generates a large number of test cases at a relatively low cost. In other words, our approach can effectively alleviate the issues of expensive manual annotation and low dataset quality. Moreover, our experimental results indicate that our approach has a quite high chance of revealing issues in RE models.

Compared with traditional dataset-based evaluation methods, our method provides assistance in understanding the behavioral capabilities of RE models and revealing shortcomings in specific aspects. For instance, according to our experiments, the relatively high VR values for MR-R_1, MR-R_2, and MR-R_3 indicate that the performance of RE models is not satisfactory in dealing with the consistency of entities. Specifically, the highest VR values for MR-R_3 suggest that the RE model relies heavily on entity type information when making decisions. Furthermore, our experiments revealed issues with data overfitting in RE models, which may be prevalent in current AI models and also demonstrate instability and poor robustness when encountering the same type of entity replacement.

Consequently, we expect that the improved RE models will be capable of learning to perform the intended task based on the understanding of entity information and contextual semantics, rather than making shallow template matches or relying on shallow heuristics that are effective for solving many existing dataset instances but may fail in more challenging examples.

6. Threats to Validity

The first threat to the validity of this work comes from the correctness and syntactic validity of the follow-up inputs. For MR-R_2 and MR-R_4, we employed various existing NLP tools to facilitate the implementation of MRs, upon which follow-up inputs could be automatically constructed. However, the reliability of the NLP tools can also affect the quality of the follow-up inputs. On the other hand, for MR-R_1 and MR-R_3, random replacement of candidate entities may lead to slight grammatical and semantic errors in sentences. In fact, this is a common problem when validating deep learning tasks with MT [37,38], and it is never possible to guarantee that the test inputs are completely error-free. In this work, to alleviate this threat, we inspected the syntactic and semantic validity of the generated follow-up inputs as described in Section 4.3. Compared with the error rate of the labels in the TACRED dataset, the generated follow-up test inputs exhibited a fairly low error rate.

Another threat to effectiveness is related to identified MRs. In our study, we identified and implemented eight MRs for RE, based on which we analyzed and discussed the performance of the subject RE models. However, these eight MRs do not cover all aspects and functions of the RE task, so our evaluation results may lead to missing some potential errors and insufficiently reflecting the strengths and weaknesses of the RE models. In the future, new MRs will need to be designed for more properties of RE. At the same time, a more powerful MR is needed to enhance the ability to reveal violations.

7. Related Works

In this section, we summarize recent research works related to our study. We divide them into two parts: the application of MT and the evaluation of RE models.

7.1. Applications of MT

MT has achieved a series of successful applications in quality assessment and traditional software verification [16,20,22,39–41], and it has also shown good results in performance defect detection [42] and automatic driving system fault detection [43,44]. Recent works applied MT to some NLP tasks and applications, including question answering systems [24–26], sentiment analysis [25,27,28], natural language inference [29], and machine translation [45–47]. Using MT can not only alleviate the oracle problem but also reveal its robustness [25,47,48], fairness [27,28], and other properties through the performance of the system under testing on different MRs. For example, Zhou et al. [47] used MT to evaluate the quality of online search engines, which not only guides developers to find out the weaknesses of the system under testing but also helps users choose an appropriate online search engine in specific situations. Inspired by these studies, *this paper proposes evaluating RE tasks by using MT*.

7.2. Evaluation of RE Models

RE models are usually validated on a held-out dataset. TACRED is one of the largest and most widely used crowdsourced datasets in RE [6]. However, even though recent progress has been made in knowledge enhancement and pretrained language models, RE models still showed high error rates of over 25% under the TACRED dataset. However, recent studies have found that the annotation quality of the TACRED dataset may strongly impact RE performance [14,49]. Alt et al. [13] revealed a large number of incorrect labels in TACRED. They found that label errors accounted for 8% of the absolute F1 test error and that more than 50% of the examples needed to be relabeled. On the relabeled test set, the average F1 score of the large baseline model set was greatly improved. Stoica et al. [35] reannotated TACRED and publicly released the revised dataset: Re-TACRED. After reannotation, it can be observed that 22.18% of the labels were different from the TACRED dataset. In fact, issues from the quality of the dataset made it difficult to tell whether the failures were due to model capabilities or label errors in the dataset itself. At the same time, it is doubtful whether the evaluation method of RE models based on the TACRED dataset is accurate and reliable. *Notably, our source inputs from the TACRED dataset were only sentence and head (tail) entity triples, which bypassed the need for high-quality labels.*

Li et al. [50] introduced the adversarial attack [51], counterfactual test [52], and bias (i.e., selection and semantics) tests [53] for BERT in the RE task to evaluate its generalization ability in terms of robustness. Alt et al. [13] found that when entities are not masked, the RE models may exploit the shallow cues present in entities to correctly classify relations, even with limited access to a sentence's context. Inspired by this study, Peng et al. [11] used the entity-masked contrastive pretraining framework to prove that RE models have the problem of overreliance on entity information, especially the entity type. Through comparative experiments, Rosenman et al. [36] pointed out that the RE model adopts a shallow heuristic method and cannot be generalized to more challenging datasets, but the construction of the comparison set requires human effort. It is noted that these studies are insufficient to comprehensively reveal the strengths and weaknesses of different RE models. *This paper designed eight MRs to encode various properties of RE, which support the application of MT in RE and also contribute to the investigation and understanding of the characteristics of RE models.*

8. Conclusions

In this paper, we proposed evaluating relation extraction (RE) models by metamorphic testing in order to alleviate the oracle problem and also support a deeper understanding of the characteristics of RE models. Eight metamorphic relations (MRs) covering the important properties of the RE task were identified, and experiments were conducted to demonstrate the validity and effectiveness of our approach. Our experiments expose a large number of prediction failures in the subject RE models and also revealed the advantages and disadvantages of our subject RE models. Further analysis of the experimental results revealed typical issues detected from RE models as well as the common features of RE models. In the future, we would like to further evaluate the effectiveness of our method on more RE models and datasets. Meanwhile, we will try to design more diverse MRs by taking into account more properties of RE models, and we will continue to improve the validity of the revealed violations. We will also diagnose the causes of prediction failures as well as repair the revealed issues of RE models.

Author Contributions: Conceptualization, M.J. and Z.D.; methodology, M.J. and Z.D.; software, Y.S.; validation, S.Z. and H.H.; data curation, Y.S.; writing—original draft preparation, M.J. and Y.S.; writing—review and editing, M.J., Z.D., Y.S., H.H. and S.Z.; funding acquisition, Z.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Nature Science Foundation of China (Grant No. 62132014).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yu, H.; Li, H.; Mao, D.; Cai, Q. A relationship extraction method for domain knowledge graph construction. *World Wide Web* **2020**, *23*, 735–753. [CrossRef]
2. Diefenbach, D.; Lopez, V.; Singh, K.; Maret, P. Core techniques of question answering systems over knowledge bases: A survey. *Knowl. Inf. Syst.* **2018**, *55*, 529–569. [CrossRef]
3. Sharma, D.; Shukla, R.; Giri, A.K.; Kumar, S. A brief review on search engine optimization. In Proceedings of the 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 10–11 January 2019; pp. 687–692.
4. Zad, S.; Heidari, M.; Jones, J.H.; Uzuner, O. A Survey on Concept-Level Sentiment Analysis Techniques of Textual Data. In Proceedings of the 2021 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 10–13 May 2021; pp. 0285–0291. [CrossRef]
5. Bharti, S.K.; Babu, K.S. Automatic keyword extraction for text summarization: A survey. *arXiv* **2017**, arXiv:1704.03242.
6. Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; Manning, C.D. Position-aware attention and supervised data improve slot filling. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017.
7. Peters, M.E.; Neumann, M.; Logan IV, R.L.; Schwartz, R.; Joshi, V.; Singh, S.; Smith, N.A. Knowledge enhanced contextual word representations. *arXiv* **2019**, arXiv:1909.04164.
8. Yamada, I.; Asai, A.; Shindo, H.; Takeda, H.; Matsumoto, Y. LUKE: Deep contextualized entity representations with entity-aware self-attention. *arXiv* **2020**, arXiv:2010.01057.
9. Han, X.; Gao, T.; Lin, Y.; Peng, H.; Yang, Y.; Xiao, C.; Liu, Z.; Li, P.; Sun, M.; Zhou, J. More data, more relations, more context and more openness: A review and outlook for relation extraction. *arXiv* **2020**, arXiv:2004.03186.
10. Wang, Y.; Chen, M.; Zhou, W.; Cai, Y.; Liang, Y.; Liu, D.; Yang, B.; Liu, J.; Hooi, B. Should We Rely on Entity Mentions for Relation Extraction? Debiasing Relation Extraction with Counterfactual Analysis. *arXiv* **2022**, arXiv:2205.03784.
11. Peng, H.; Gao, T.; Han, X.; Lin, Y.; Li, P.; Liu, Z.; Sun, M.; Zhou, J. Learning from context or names? An empirical study on neural relation extraction. *arXiv* **2020**, arXiv:2010.01923.
12. Brody, S.; Wu, S.; Benton, A. Towards Realistic Few-Shot Relation Extraction. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online, 7–11 November 2021; pp. 5338–5345.
13. Alt, C.; Gabryszak, A.; Hennig, L. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. *arXiv* **2020**, arXiv:2004.14855.
14. Bassignana, E.; Plank, B. What Do You Mean by Relation Extraction? A Survey on Datasets and Study on Scientific Relation Classification. *arXiv* **2022**, arXiv:2204.13516.
15. Barr, E.T.; Harman, M.; McMinn, P.; Shahbaz, M.; Yoo, S. The oracle problem in software testing: A survey. *IEEE Trans. Softw. Eng.* **2014**, *41*, 507–525. [CrossRef]
16. Chen, T.Y.; Kuo, F.C.; Liu, H.; Poon, P.L.; Towey, D.; Tse, T.; Zhou, Z.Q. Metamorphic testing: A review of challenges and opportunities. *ACM Comput. Surv. CSUR* **2018**, *51*, 1–27. [CrossRef]
17. Segura, S.; Towey, D.; Zhou, Z.Q.; Chen, T.Y. Metamorphic testing: Testing the untestable. *IEEE Softw.* **2018**, *37*, 46–53. [CrossRef]
18. Chen, T.Y.; Cheung, S.C.; Yiu, S.M. Metamorphic testing: A new approach for generating next test cases. *arXiv* **2020**, arXiv:2002.12543.
19. Zhou, Z.Q.; Xiang, S.; Chen, T.Y. Metamorphic testing for software quality assessment: A study of search engines. *IEEE Trans. Softw. Eng.* **2015**, *42*, 264–284. [CrossRef]
20. Zhou, Z.Q.; Sun, L.; Chen, T.Y.; Towey, D. Metamorphic relations for enhancing system understanding and use. *IEEE Trans. Softw. Eng.* **2018**, *46*, 1120–1154. [CrossRef]
21. Xie, X.; Wong, W.E.; Chen, T.Y.; Xu, B. Metamorphic slice: An application in spectrum-based fault localization. *Inf. Softw. Technol.* **2013**, *55*, 866–879. [CrossRef]
22. Jiang, M.; Chen, T.Y.; Zhou, Z.Q.; Ding, Z. Input test suites for program repair: A novel construction method based on metamorphic relations. *IEEE Trans. Reliab.* **2020**, *70*, 285–303. [CrossRef]
23. Chen, T.Y.; Tse, T.; Zhou, Z.Q. Semi-proving: An integrated method for program proving, testing, and debugging. *IEEE Trans. Softw. Eng.* **2010**, *37*, 109–125. [CrossRef]

24. Yuan, Y.; Wang, S.; Jiang, M.; Chen, T.Y. Perception matters: Detecting perception failures of vqa models using metamorphic testing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 16908–16917.
25. Ribeiro, M.T.; Wu, T.; Guestrin, C.; Singh, S. Beyond accuracy: Behavioral testing of NLP models with CheckList. *arXiv* **2020**, arXiv:2005.04118.
26. Tu, K.; Jiang, M.; Ding, Z. A metamorphic testing approach for assessing question answering systems. *Mathematics* **2021**, *9*, 726. [CrossRef]
27. Ma, P.; Wang, S.; Liu, J. Metamorphic Testing and Certified Mitigation of Fairness Violations in NLP Models. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020), Yokohama, Japan, 7–15 January 2021; pp. 458–465.
28. Asyrofi, M.H.; Yang, Z.; Yusuf, I.N.B.; Kang, H.J.; Thung, F.; Lo, D. Biasfinder: Metamorphic test generation to uncover bias for sentiment analysis systems. *IEEE Trans. Softw. Eng.* **2021**, *48*, 5087–5101. [CrossRef]
29. Jiang, M.; Bao, H.; Tu, K.; Zhang, X.Y.; Ding, Z. Evaluating Natural Language Inference Models: A Metamorphic Testing Approach. In Proceedings of the 2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE), Wuhan, China, 25–28 October 2021; pp. 220–230.
30. Clark, K.; Manning, C.D. Improving coreference resolution by learning entity-level distributed representations. *arXiv* **2016**, arXiv:1606.01323.
31. Hargittai, M.; Hargittai, I. *Symmetry through the Eyes of a Chemist*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
32. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
33. Soares, L.B.; FitzGerald, N.; Ling, J.; Kwiatkowski, T. Matching the blanks: Distributional similarity for relation learning. *arXiv* **2019**, arXiv:1906.03158.
34. Zhou, W.; Chen, M. Learning from noisy labels for entity-centric information extraction. *arXiv* **2021**, arXiv:2104.08656.
35. Stoica, G.; Platanios, E.A.; Póczos, B. Re-TACRED: A New Relation Extraction Dataset. In Proceedings of the 4th Knowledge Representation and Reasoning Meets Machine Learning Workshop (KR2ML 2020), at NeurIPS, Virtual, 11–12 December 2020.
36. Rosenman, S.; Jacovi, A.; Goldberg, Y. Exposing shallow heuristics of relation extraction models with challenge data. *arXiv* **2020**, arXiv:2010.03656.
37. Chen, S.; Jin, S.; Xie, X. Validation on machine reading comprehension software without annotated labels: A property-based method. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, 23–28 August 2021.
38. He, P.; Meister, C.; Su, Z. Structure-invariant testing for machine translation. In Proceedings of the 2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE), Seoul, Republic of Korea, 27 June 2020–19 July 2020; pp. 961–973.
39. Segura, S.; Fraser, G.; Sanchez, A.B.; Ruiz-Cortés, A. A survey on metamorphic testing. *IEEE Trans. Softw. Eng.* **2016**, *42*, 805–824. [CrossRef]
40. Jiang, M.; Chen, T.Y.; Kuo, F.C.; Towey, D.; Ding, Z. A metamorphic testing approach for supporting program repair without the need for a test oracle. *J. Syst. Softw.* **2017**, *126*, 127–140. [CrossRef]
41. Segura, S.; Parejo, J.A.; Troya, J.; Ruiz-Cortés, A. Metamorphic testing of RESTful web APIs. *IEEE Trans. Softw. Eng.* **2017**, *44*, 1083–1099. [CrossRef]
42. Segura, S.; Troya, J.; Durán, A.; Ruiz-Cortés, A. Performance metamorphic testing: A proof of concept. *Inf. Softw. Technol.* **2018**, *98*, 1–4. [CrossRef]
43. Tian, Y.; Pei, K.; Jana, S.; Ray, B. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In Proceedings of the 40th International Conference on Software Engineering, Gothenburg, Sweden, 30 May–1 June 2018; pp. 303–314.
44. Zhang, M.; Zhang, Y.; Zhang, L.; Liu, C.; Khurshid, S. DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems. In Proceedings of the 2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE), Montpellier, France, 3–7 September 2018; pp. 132–142.
45. Sun, Z.; Zhang, J.M.; Harman, M.; Papadakis, M.; Zhang, L. Automatic testing and improvement of machine translation. In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, Seoul, Republic of Korea, 27 June 2020–19 July 2020; pp. 974–985.
46. Sun, L.; Zhou, Z.Q. Metamorphic testing for machine translations: MT4MT. In Proceedings of the 2018 25th Australasian Software Engineering Conference (ASWEC), Adelaide, SA, Australia, 26–30 November 2018; pp. 96–100.
47. Lee, D.T.; Zhou, Z.Q.; Tse, T. Metamorphic robustness testing of Google Translate. In Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, Seoul, Republic of Korea, 27 June 2020–19 July 2020; pp. 388–395.
48. Mao, C.; Yi, X.; Chen, T.Y. Metamorphic Robustness Testing for Recommender Systems: A Case Study. In Proceedings of the 2020 7th International Conference on Dependable Systems and Their Applications (DSA), Xi'an, China, 28–29 November 2020; pp. 331–336.
49. Taillé, B.; Guigue, V.; Scoutheeten, G.; Gallinari, P. Let's Stop Incorrect Comparisons in End-to-end Relation Extraction! *arXiv* **2020**, arXiv:2009.10684.

50. Li, L.; Chen, X.; Ye, H.; Bi, Z.; Deng, S.; Zhang, N.; Chen, H. On robustness and bias analysis of bert-based relation extraction. In Proceedings of the China Conference on Knowledge Graph and Semantic Computing, Guangzhou, China, 4–7 November 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 43–59.
51. Jin, D.; Jin, Z.; Zhou, J.T.; Szolovits, P. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8018–8025.
52. Gardner, M.; Artzi, Y.; Basmova, V.; Berant, J.; Bogin, B.; Chen, S.; Dasigi, P.; Dua, D.; Elazar, Y.; Gottumukkala, A.; et al. Evaluating Models' Local Decision Boundaries via Contrast Sets. *arXiv* **2020**, arXiv:2004.02709.
53. Shah, D.; Schwartz, H.A.; Hovy, D. Predictive biases in natural language processing models: A conceptual framework and overview. *arXiv* **2019**, arXiv:1912.11078.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.