applied sciences

MDPI

Article

An Open Relation Extraction System for Web Text Information

Huagang Li and Bo Liu *

College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China; lihuagang21@163.com
* Correspondence: kyle.liu@nudt.edu.cn

Abstract: Web texts typically undergo the open-ended growth of new relations. Traditional relation extraction methods lack automatic annotation and perform poorly on new relation extraction tasks. We propose an open-domain relation extraction system (ORES) based on distant supervision and few-shot learning to solve this problem. More specifically, we utilize tBERT to design instance selector 1, implementing automatic labeling in the data mining component. Meanwhile, we design example selector 2 based on K-BERT in the new relation extraction component. The real-time data management component outputs new relational data. Experiments show that ORES can filter out higher quality and diverse instances for better new relation learning. It achieves significant improvement compared to Neural Snowball with fewer seed sentences.

Keywords: open relation extraction; few-shot learning; knowledge extraction; tBERT; K-BERT

Citation: Li, H.; Liu, B. An Open Relation Extraction System for Web Text Information. *Appl. Sci.* **2022**, *12*, 5718. https://doi.org/10.3390/app12115718

Academic Editors: Katia Lida Kermanidis, Phivos Mylonas and Manolis Maragoudakis

Received: 6 May 2022
Accepted: 31 May 2022
Published: 4 June 2022

1. Introduction

Information and knowledge are the basis for the development of human society. Text records 80 (https://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/, accessed on 5 May 2022) percent of the information of human civilization. The core task of information extraction (IE) is to obtain structured triples from unstructured text. It relies on two fundamental tasks: entity recognition and relation extraction. Li et al. [1] proposed an entity recognition method that performs well. For relation extraction, new relation prediction is a challenge. Traditional relation extraction mainly adopts supervised learning methods for predefined relations. Its essence is to transform relation extraction into relation classification. There are two paradigms: pipeline relation extraction [2] and joint relation extraction [3]. Traditional RE performs well but faces two challenges. The first challenge is that predefined relation classifications do not work well on new relation extraction tasks. The second challenge is that relational data relies too much on manual cleaning and labeling, which is costly. In addition, for large-scale knowledge bases such as Wikidata, manual annotation would be challenging to accomplish.

To solve this problem, Banko [4] first proposed the concept of open information extraction. That is, extracting structured relational facts from open and growing unstructured text. Information extraction should not be limited to a small set of known relations. RE should be able to extract a wide variety of relations in a text. The scope of its research is that the entity pair of the relation is known, and the relationship type between the entity pair is unlimited. Open-domain relation extraction should meet three academic requirements: automation, non-homologous corpus, and high efficiency.

- Automation
  The open relation extraction system can execute automatically, and the algorithm only needs to go through the corpus once for triple tuples extraction. It should be based on an unsupervised extraction strategy and cannot be a predefined relation. In addition, the cost of manually constructing training samples is small, and only a tiny number of initialization seeds need to be labeled or a small number of extraction templates need to be defined.

- Non-homologous corpus
  The goal of open-domain relation extraction is domain-independent and should not be limited to a particular domain. In addition, it is important to avoid using domain-dependent deep syntactic analysis tools, such as syntactic analysis.
- Efficiency
  Open information extraction systems handle large-scale corpora, and a high efficiency is required to ensure a rapid response. Prioritizing shallow syntactic features is also necessary.

The open-domain relation extraction methodology mainly includes four methods: based learning, clause, rule, and tuple association. OLLIE [5] is a knowledge base based on the open information extraction system REVERB [6]. Its idea is to learn a pattern to extract relations, which requires high generalization and robustness of the pattern. ReNoun [7] is an open information extraction system that entirely focuses on noun relation extraction. In this system, a high-precision lexical template is first formulated manually. Then, based on the template, the little relation tuples are mined, and the confidence is marked. This work introduces a distant supervision method that reduces the reliance on labeled data. Many complex sentences have relations associated with multiple clauses, and the model has a low performance when dealing with complex sentences.

To improve the accuracy, Mausam [8] proposed a clause-based system, ClausIE, which converts the dependencies of input sentences into simple coherent clauses based on linguistic knowledge. Then, the argument information is extracted according to the clause type. The Stanford open information extractor was proposed by Angeli et al. [9]. In addition, the rule-based method is more effective in improving accuracy. Gabriel [10] proposed a non-lexicalization rule to convert the dependency syntax tree into a relational directed graph for processing. It works well on multilingual tasks. Rule-based methods are more accurate for specific tasks but rely on human rules and are more complex. In order to improve the accuracy of extracting triples, a method based on tuple association has also been proposed to solve the problem of sentence association.

The method reconstructs sentences, transforming a complex non-normalized sentence segment into a set of clauses with dependencies and easy-to-extract triples. Graphene et al. [11] proposed a method for the hierarchical simplification of complex sentences from the perspective of obtaining tuple associations and used syntactic and lexical patterns to predict the modification relationship between sentences. The process is sentence reconstruction, clause splitting, information reduction, and clause triple extraction. This method is beneficial in improving accuracy.

The method based on rules, clauses, and obtaining tuple associations uses templates to extract relational tuples from a syntactic and lexical point of view. Learning-based methods learn a pattern from semantic information and use the learned pattern to extract tuples. At present, rule-based methods can achieve a high accuracy. Furthermore, learning-based methods such as Neural Snowball are effective, and form the baseline for new relation extraction tasks. This paper uses the Neural Snowball [12] as the open-domain RE baseline model for subsequent experiments. Most of these open-domain relation extraction methods only utilize a small amount of supervised data and do not effectively utilize the newly growing relational data.

Most of the current open relation extraction methods only utilize a small amount of supervised data and do not utilize the setting of newly growing relational data. However, these new relational data can predict unseen relations, which is valuable. Based on the Bootstrap [13] algorithm, this paper comprehensively utilizes small-scale labeled data, large-scale unsupervised data, and few-sample new relational data. We propose a system that can realize data self-labeling and new relation extraction, ORES. Figure 1 shows the process of how ORES utilizes different kinds of data to learn new relations. After distant supervision obtains sentences, the model based on the self-sampling algorithm screens instances. Furthermore, a new-relation classifier is trained by inputting filtered high-confidence instances and large-scale supervised data. The classifier can iteratively discover

more reliable instances with new relation facts and adapt them to open growth scenarios. In more detail, we design sample selector (SS), based on a Siamese network (SN) [14] to select high-confidence instances. The SN is used to classify whether existing instances and new ones express the same relation. The experimental results show that ORES can select high-quality instances and significantly improve extracting new relations through few-shot samples. In conclusion, our main contributions are as follows:

- We propose a new open relation extraction system (ORES), a novel structure to train a neural relation classifier with few initial new relation instances by iteratively accumulating new instances and facts from unlabeled data with prior data knowledge of historical relations. We design three functional components of ORES to realize the mining of Web text information, automatic labeling, and extraction and output of new relation data.
- We design and combine two new encoders based on tBERT and K-BERT language models to better express features in textual information. Experiments show that combined cooperation is beneficial for improving model performance. Specifically, for the sample selector 1 of the data mining component, we introduce a topic model and design a new encoder based on tBERT, which improves the recall rate. For the example selector 2 of the relation extraction component, we design a new encoder based on K-BERT to inject external knowledge for further improvement in accuracy. We also conducted a fusion experiment to prove the effectiveness of the combination.

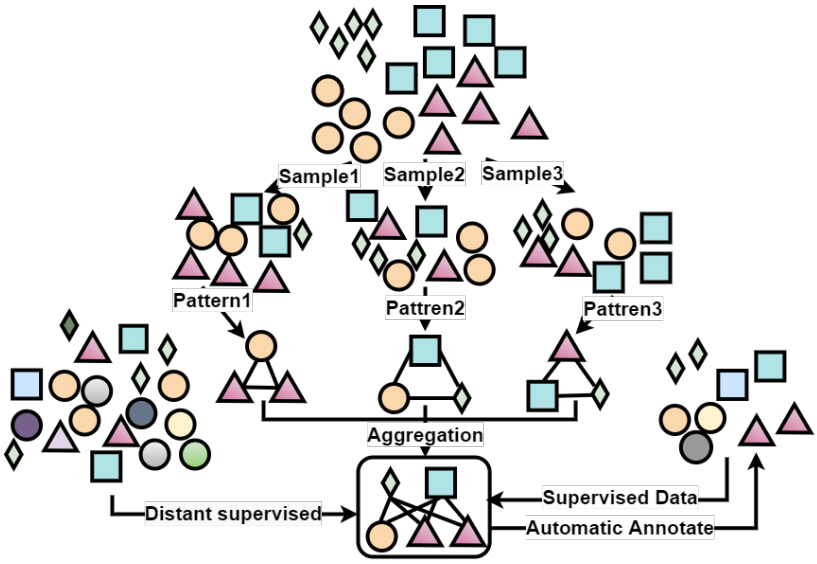

Figure 1. An illustration of how ORES utilizes 3 kinds of data for relation discovery based on distant supervision and bootstrapping.

The rest of the paper is structured as follows: Related Work treats important methods in relation extraction and ORES. Methodology concerns principles and operation process of ORES. The Experiments and Results section provides detailed information on the design, setup, implementation, and results of the experiments. Discussion discusses and analyzes the experimental results in detail. Conclusion and Future Work summarizes the work of this paper and proposes better research directions for the future.

2. Related Work

2.1. Supervised RE

Relation extraction (RE) [15] aims at extracting relational fact triples of the form (head, relation, tail) from unstructured data. The main goal of RE is to classify relations between those entities mentioned from their context. PROPS [10], based on rule transformation, transforms the dependency parsing tree of sentences into a directed graph representing

the association of relation tuples and then directly uses rules to extract triples from the directed graph. Traditional restricted-domain relation extraction mainly adopts supervised research methods. The pipeline method consists in entity recognition followed by relation extraction, and the two models are relatively independent. It is generally believed that the pipeline method suffers from error propagation, which affects the performance of relation extraction. The joint method unifies the two sub-models of entity recognition and relation extraction to use the mutual information of the sub-models to delay error propagation. Recently, Chen [16] proposed a pipeline method research that broke this inertial understanding. It achieves several SOTA scores on standard benchmarks. Moreover, Chen designed experiments to confirm that entity recognition and relation extraction learn different features, and joint modeling will interfere.

Supervised relation extraction relies on labeled data, but the performance is degraded for open scenarios due to insufficient labeled data. Its essence is to train a relation classifier based on predefined relations. The limitation of traditional relation extraction methods is that they rely on manually annotated relational datasets. In addition, based on the predefined relation classification, the new relation extraction problem cannot be effectively solved. To solve the problem of traditional domain-limited relation extraction, semi-supervised relation extraction [17] has received widespread attention. Semi-supervised relation extraction uses a small amount of annotation information to learn for open scenarios. Its idea is to learn local features from a small number of labeled samples, learn the overall distribution from a large amount of unlabeled data, and then build a model. There are two main methods for semi-supervised relation extraction: bootstrap and distant supervision methods.

2.2. Bootstrapping Algorithm for RE

In order to solve the problem of low resources, Efron [18] proposed an augmented sample statistical method called bootstrapping. The bootstrapping algorithm, which repeatedly samples a limited number of samples, creates a new sample that can represent the distribution of the parent sample. The idea is to replace the relation between the observed statistic and the true value with the relation between the bootstrap statistic and the observed statistic. The deviation of the observed statistic is estimated from the true value by studying the bootstrap statistic. Bootstrap is a process of repeated resampling, and the key process is to calculate observation statistics. Assuming there are n observations, the calculation steps are:

- Record and store each observation;
- Put all samples in black box;
- Randomly sample and record its observations;
- Put the sample back, and then randomly sample.

Repeat steps 3 and 4 n times to obtain a self-help sample. The value and distribution of this sample are then used to estimate the true value and distribution of the parent sample. The mathematical definition of bootstrap is to set the random sample:

$$X = [x_1, x_2, \ldots, x_n] \tag{1}$$

It is independent and identically distributed, namely:

$$x_i \sim F(x), i = 1, 2, 3, \ldots, n \tag{2}$$

Mark $R(X, F)$ as a function of random samples X and distribution function F. Now estimate the distribution characteristics of $R(X, F)$ based on the observed samples $[x1, x2, \ldots, xn]$. θ is a parameter of the population distribution F, and F_n is an estimate of the empirical distribution function of the observed sample X:

$$\hat{\theta} = \hat{\theta}(F_n) \tag{3}$$

The error T_n is:

$$T_n = \hat{\theta}(F_n) - \hat{\theta}(F) \quad (4)$$

The traditional bootstrapping RE mainly uses a small number of instances as the initial seed (seed, tuples) set and then learns a pattern. It then extracts more (seed, tuples) based on this pattern. Through iterative training, the seed set and pattern library are continuously expanded. Compared with the convenience sampling method, the bootstrap method can calculate the statistic's value and simulate the statistic's distribution. Therefore, the actual sample space can be inferred based on the bootstrap statistic.

The earliest application of the bootstrapping method in knowledge extraction is the DIPRE system [19]. A small number of instances is inputted, Web text instances are mined by learning seed patterns, and then a new (seed, tuples) set is expanded. Based on DIPRE, Agichtein carried out the credit score and pattern completeness description for the newly extracted instances and proposed Snowball [20] for the first time. Some subsequent systems have continued the bootstrapping method and studied how to add more reasonable pattern descriptions, more reasonable constraints, and scoring strategies.

In addition, there is work to build large-scale patterns based on the extraction results. Typically, the NELL (never-ending language learner) system uses the method of Internet mining to extract triple knowledge from Web texts automatically. More than 3.1 million relational facts have been extracted to date. NELL initializes an ontology and seed pattern and then learns from large-scale Web text. Finally, the accuracy is improved by scoring what is learned. The idea of traditional bootstrap is to learn patterns based on seed examples and then to iteratively expand the knowledge base; its initial seed is more restrictive to the extracted content. This paper draws on the work of Neural Snowball and others to further study the problem of open relation extraction.

2.3. Distant Supervised RE

Distant Supervised Relation Extraction [21] (DSRE) aligns entity pairs in the corpus based on existing relations in the knowledge base. It assumes that the same entity pairs also have approximately the same relation. Through the mechanism, distant supervision can generate relational labels and automatically label relational data. The at-least-one hypothesis is proposed to reduce the wrong labeled data, which considers that at least one example mined by distant supervision has a known relation. However, many methods based on this hypothesis have low sentence information utilization and fail to fully utilize other information in all examples. Later, Zeng et al. [22] propose segmented convolutional neural networks (PCNNs) for multi-instance learning, which can effectively utilize the information of multiple sentences.

Lin et al. [23] proposed a sentence-level attention mechanism. By calculating the relation weights of the sentences in the bag, the probability that the marked sentence belongs to the target relation can be determined. This method can fully specify the information value of all sentences in the sentence bag. Ji et al. [24] proposed a sentence-level attention model (APCNN) which can effectively select high-confidence instances. APCNNs inject the background knowledge of entities from Freebase and Wikipedia, making full use of the supervision information in the knowledge base. Methods that inject external knowledge can provide more helpful information for relation prediction. It is beneficial for improving the accuracy of the model. Some of the above studies mainly study automatic annotation from remote supervision and noise reduction by aligning existing relationships in the knowledge base.

In addition, automatically generating annotation by generating interpretable relation labels is also a promising research direction. Jia [25] proposedan interpretable model, ARNOR, based on attention regularization to generate relation labels. The model filters out noise and selects true positive instances based on semantic information and pattern matching. The noise reduction effect of ARNOR is noticeable, but there is a problem with the pattern tendency, and the generalization is not good. Liang et al. [26] proposed the BOND framework based on distant supervision methods for solving open-ended named-

entity recognition tasks. BOND is the baseline for the NER task, and its ideas can also be used in relation extraction research. In this paper, the remote supervision method is used to realize the function of the data mining component.

2.4. Few-Shot Learning for RE

The remarkable learning ability of humans is due to the interaction between the prefrontal cortex (PFC) in the human brain and working memory stored in other brain regions. Humans can build a global understanding of new things or concepts by learning from a few samples. However, it is difficult for machines to have strong generalization performance with few samples. Whether a machine with prior knowledge can generalize the overall sample features with only a small number of support set samples is a problem to be studied in few-shot learning. Traditional machine learning is aimed at a classification task; it obtains supervised experience, E, through pre-training, and improves the accuracy of the classifier according to E. Few-shot learning [27] means that the machine learns a large amount of data of a specific category and obtains supervised experience, E. When processing a new category, only a small amount of data can be input to learn global features quickly. It is the application of meta-learning in supervised learning, but it is different from the classification problem of supervised learning.

In few-shot learning, different meta-tasks are sampled. The overall training process contains different class combinations. The mechanism enables the model to identify common parts in different meta-tasks, such as extracting important features, comparing samples for similarity, and dropout meta-tasks in task-related parts. Therefore, the model predicts more accurately when faced with new unseen relations. Unlike traditional machine learning, the goal of few-shot relation learning is not for the machine to identify relation types in the training set and generalize to the test set. Its goal is for machines to learn to recognize and predict new types of relations, embodying a learn-to-learn philosophy.

As shown in Figure 2, few-shot learning trains a neural network based on the training set. Then, the support set and query set instances are inputted into the neural network. Let the classifier judge an instance Sn of the query set and an instance of the support set: F1, G1, H1, I1, and S1, of which one instance belongs to the same relationship. The purpose of learning is to enable the model to identify unknown relations and understand the similarities and differences of the relations to which instances belong. Based on this thought, this paper trains a binary classifier to discriminate new relations to solve the problem of new relation extraction.

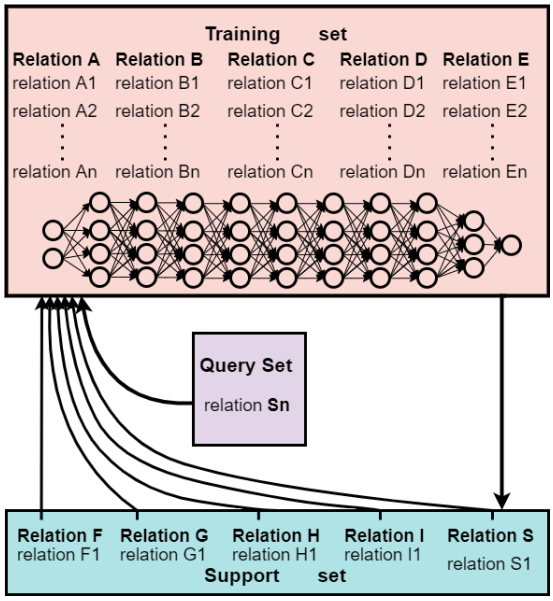

Figure 2. Intelligent education blueprint based on open-domain relation extraction and AI technology.

3. Methodology

Bert [28] can learn the semantic representation of most entities in the pre-training stage, but the representation of specific entities is not accurate enough. Therefore, we introduce tBERT and K-BERT language models to improve component performance. Chen et al. [29] proved that awareness topics could yield improvements in machine translation. Inspired by these works, we integrate a topic model into BERT to compute topic information for each sentence. In the similarity detection stage, it can consider its topic-relatedness. Although there is no standard method for combining topic models with pre-trained models, Nicole Peinelt [30] proposed the tBERT model by combining best and topic models for semantic similarity. It performs better than BERT in specific fields, and we also cite the idea and methods of this work. This paper will introduce the structure and principle of each component when introducing the working process of ORES and explain the corresponding formulas and symbols.

3.1. Few-Shot Relation Extraction Learning Framework

As shown in Figure 3, ORES includes three parts: a data mining component, an open relation extraction component, and a real-time data management component. The data mining component is used to mine instances from Web text with the same relation as the seed set. The open relation extraction component trains a classifier based on few-shot learning to predict a new relation, r. It is a real-time data management component that stores and manages the output of new relational data. The working process of ORES is multiple iterations, and each iteration has two stages:

- Phase 1
  (a) Add sentences in the initial seed set to the coarse selection set. As shown in Figure 4: it put "Einstein founded the theory of relativity" into the coarse selection set from the seed set.
  (b) Using distant supervision, mine sentence pairs from Web text and add them to the coarse selection set and the selection set. At first, perform named-entity recognition, and extract the head and tail entities from the initial seed sentence X as entity pair δ.

$$\delta = \{(e_h, e_t) | Ent(X) = (e_h, e_t), X \in S_i\} \tag{5}$$

  S_i is the initial seed set, Ent is the entity pair matching function, and X is the sentence in the initial seed set. Then mine all sentences containing entity pair γ in the Web text denoted as X'.

$$Cs = \{(X' | Ent(X') \in \delta, X' \in W_t\} \tag{6}$$

  Cs is a rough selection, and Wt is a Web text library.
  (c) Load sentences in the coarse selection set into instance selector 1. The tBERT encoder in instance selector 1 outputs the representation vectors of the embedding features of the sentence pair, denoted as em. The instance selector 1 through the Siamese network calculates the similarity of the two sentences and scores them. The similarity distance function is:

$$S(X, Y) = \sigma(W_s^T (e_m(X) - e_m(Y))^2 + b_s) \tag{7}$$

  where W_s^T and b_s are learnable parameters.
  (d) After the instance selector 1 filters out sentences that meet the threshold, add them to selection 1 and the seed set. The instance selector 1 and instance selector 2 are denoted as f, and the threshold is set to 0.5, marked as α. Selection 1 is denoted as $S1$:

$$S_1 = \{X | f(X) > \alpha, X \in W_t\} \tag{8}$$

  where W_t represents Web text.

(e) Train a relation classifier with a small amount of labeled data and instances in selection 1 as input to identify sentences with a new relation r.

- Phase 2
  (a) Classifier g mines sentences in Web text that may belong to relation r. The confidence region θ of g is set to 0.9. When the input instance meets the threshold condition, it is filtered out and added to the selection set.

$$S_s = \{X | g(X) > \theta, X \in R^r\} \tag{9}$$

  S_s is the sentence selection set. R^r is the large-scale annotated relation set.
  (b) Filter the selection set again to enhance the performance of the classifier. Then, load the sentences in the selection set to the instance selector 2.
  (c) After sentences are loaded into the coarse selection set, one iteration is complete. To further improve the accuracy, the instance selector 2 adopts the K-BERT encoder. K-BERT is a language model that can inject external knowledge.

At the end of the first stage, the relation classifier can extract a new relation r. When mining new relational sentences based on relation r, entity pairs can change. As shown in Figure 3, when the classifier learns the new relation "of_founder", the sentence pair is matched: "Newton established the classical mechanics system" and "Tesla invented the alternating current system". The first instance reflects the founder relationship. The second reflects the new relationship inventor. As shown in Figure 4, such instances, called query instances, will be expanded to the rough selection set in subsequent iterations. Query instances and seed sentences will perform similarity detection. When the similarity is less than or equal to the threshold α, it is regarded as a new relation. In subsequent rounds of iterations, it will be annotated and expanded as new relational data. The above two stages are an iterative process, and multiple iterations can continuously learn and expand new relationship types.

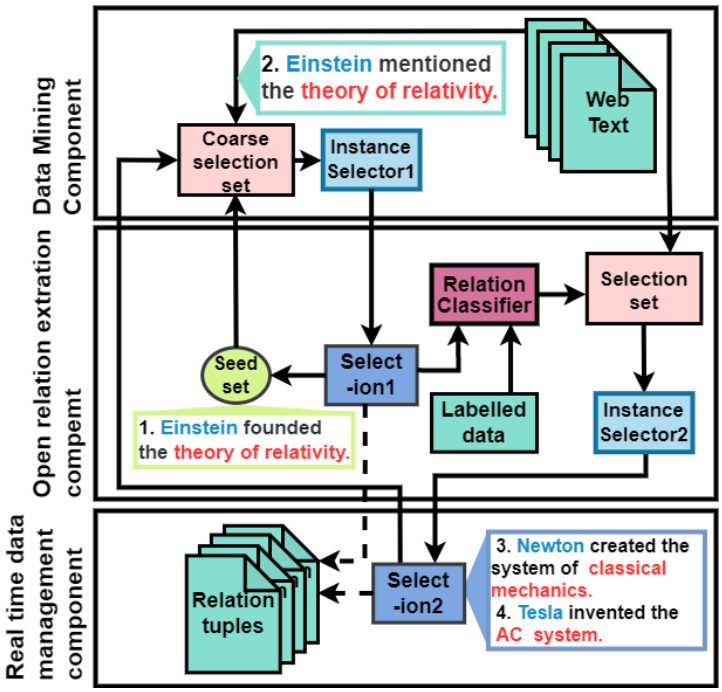

Figure 3. The framework of ORES and internal components' structure, along with some examples during one iteration. ORES includes 3 components to achieve different functions. Instance selector 1 is the core structure of the data mining component to filter Web text information. ORES's core model is a relation classifier which can extract increasing relation facts. Web texts are the input and relation tuples are ORES's outputs. There are 4 sentences. The first one is one of seed samples. The second sentence is mined by distant supervision. The third and the fourth are ORES's outputs.

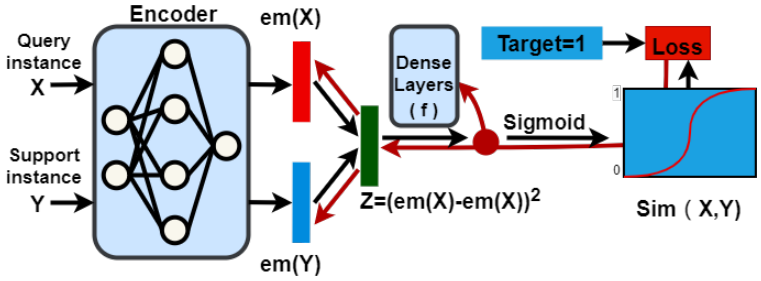

Figure 4. The description of the structure, principle, and training process of the instance selector.

3.2. Instance Selector

The instance selector is pre-trained, and it performs similarity detection between sentences obtained from Web text and initial seed sentences for filtering out instances with high confidence. The instance selector adopts the relation Siamese network structure. As shown in Figure 5, sentence pair X and Y are input. The similarity score $Sim(X, Y)$ is output, which represents the likelihood that X and Y share the same relation in the range [0, 1]. After the $s(X, Y)$ sentence pair is input to the encoder, the word embedding process is performed. The encoder learns an embedding matrix to produce the representation vectors of instances' feature and output extracted word vector. Then, the Z vector is processed with a fully connected layer and a scalar is output. Finally, a sigmoid activation function is used to obtain a real value between 0 and 1. After the $s(X, Y)$ sentence pair is input to the encoder, the word embedding process is performed.

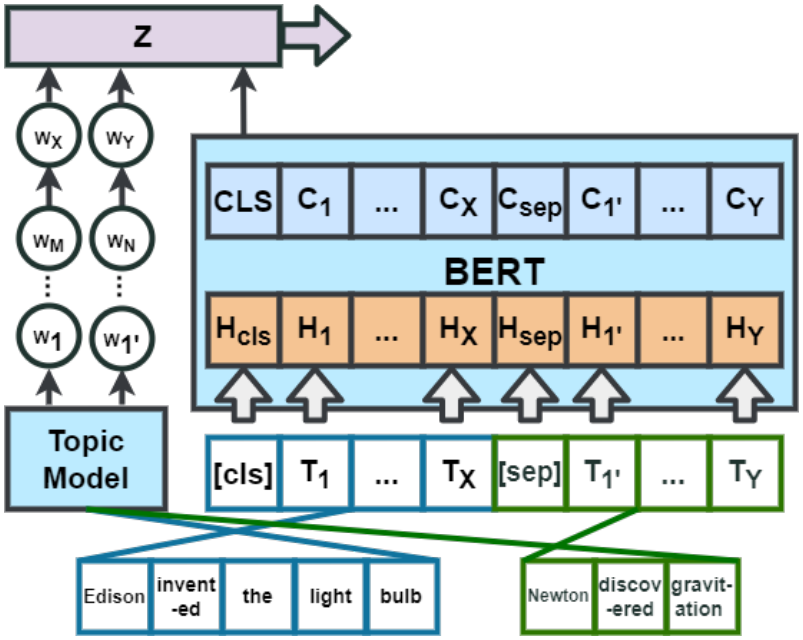

Figure 5. An illustration of tBERT's architecture and the process of tokens converting to topic vectors.

The encoder learns an embedding matrix to produce the representation vectors of instances' features, and an extracted word vector is output. Then, the Z vector is processed with a fully connected layer and a scalar is output. Finally, a sigmoid activation function is used to obtain a real value between 0 and 1. The cross-entropy of the label and prediction is used as the loss function. Backpropagation is used to compute gradients and update model parameters by gradient descent. The instance selector has two parts. One is the encoder neural network, which extracts instance embedding features to generate a word vector. The other is a fully connected layer used to predict similarity. The training process is to update the parameters of these two parts. Specifically, the gradient is passed from the loss function back to the fully connected layer and the parameters of the vector Z. Then, gradients are

further propagated from the vector Z to the encoder neural network, where gradients are used to update the parameters of the encoder neural network. The above is one iteration of the sample selector training process.

The encoder trained through few-shot learning needs to prepare the same number of positive samples and negative samples. The labels of negative samples are set to 0, and the goal is to predict the neural network close to 0 as well. Likewise, the model updates the fully connected layer parameters and the encoder neural network parameters through negative backpropagation. In particular, the training data for the Siamese network do not contain the new relation. Encoders are pre-trained and embedded into the ORES instance selectors. Trained instance selectors (IS) can predict whether an instance represents a new relation. Then, the instance selector can predict which instance in the seed set has the highest similarity in relation to distant supervised instances.

3.3. Encoder

Instance selector 1 and instance selector 2 adopt the same structure, namely, the Siamese network. The difference is the encoder setting, which is determined by the functionality of the components. The instance selector filters the examples obtained through distant supervision in the data mining component. If only based on the semantically similar screening instance, it is easy to fall into the comfort zone of semantically similar but insufficient diversity. Instances processed by the data mining component are topic-specific. In the open relation extraction component, we adopt K-BERT as the encoder of sample selector 2 in order to improve the similarity detection effect.

3.3.1. tBERT Encoder

Neural Snowball filters instances based on BERT-encoded semantic information. Experiments show that the model has the problem of a low recall rate. In the process of screening instances, they stay in the comfort zone with similar semantics but insufficient diversity. Further analysis can also find that filtering instances based on semantic similarity can easily lead to the problem of semantic similarity but insufficient diversity. Adding topic information to the model is beneficial to increase the diversity of the selected instances and help the model get out of the "comfort zone" of semantic similarity but insufficient diversity. tBERT performs better than BERT in specific domains, and this paper draws on the ideas and methods of this work. In addition, Topic Snowball [31] is our previous work, and this paper further extends on this basis. The tBERT encoder consists of BERT and topic model. After the sentence pair is input, the specific process is as follows:

Firstly, encode the sentence pair and launch all tokens into a topic model, denoted as tModel. The topic model matches a topic ω_i for each word in the sentence:

$$\omega_i = tModel(t_i) \in R^t \tag{10}$$

where t is the number of topic types. Then, pooling obtains sentence-level topics W_X,W_Y:

$$W_X = \frac{1}{N_X} \cdot \sum_{i=1}^{N_X} \omega_i \in R^t \tag{11}$$

$$W_Y = \frac{1}{N_Y} \cdot \sum_{i=1}^{N_Y} \omega_i \in R^t \tag{12}$$

N_X and N_Y represent the number of words in sentence X and Y, respectively. Meanwhile, it pushes all tokens into BERT for encoding sentence X and sentence Y, using the C vector as a sentence pair representation:

$$C = BERT(S_X, S_Y) \in R^h \tag{13}$$

where h represents the hidden layer dimension inside BERT. The joint real-valued vector C with sentence topic vectors W_X and W_Y is:

$$Z = [C, W_X, W_Y] \in R^{t+2h} \tag{14}$$

Finally, cross-entropy is used as the loss function for training. The loss function is as follows:

$$H(p, g) = -\sum_{i=1}^{n} p(x_i) \lg(g(x_i)) \tag{15}$$

where $p(x)$ is the true outcome distribution, and g is the output distribution predicted by the relation classifier. Cross-entropy is used as the loss function and the parameters are adjusted according to the loss. The fine-tuning stage uses a grid search to select the parameter settings for the best experimental results. Adding a topic model to the BERT encoder essentially considers the topic a similarity weight. Semantic-based similarity detection enables the model to slightly reduce the weight of semantic similarity while noticing that the topic is related. It is conducive to screening more diverse instances and improving recall.

3.3.2. K-BERT Encoder

BERT does well in common named-entity representation, but not in the professional field of named entities. It is related to the pre-train mechanism that BERT captures entity representation on a large-scale common corpus but lacks domain-specific knowledge. The underlying reason is that domain knowledge is rarely present in the pre-training stage. Although expertise can be absorbed during the fine-tuning stage, the actual effect is not good. If it is directly trained on the corpus in the professional field, it will need to invest a significant cost and have low transferability. Therefore, Liu proposes directly injecting domain knowledge annotation that would only appear in the fine-tuning stage into the pre-training stage. The knowledge graph is injected into the instance as external domain knowledge. This will not only improve performance but also effectively reduce costs.

There are two reasons for choosing a knowledge graph [32] as an external knowledge source. The first is that a knowledge graph is marked and verified structured knowledge, and thus is rich, reliable, and easy to query. The other is that the construction of the knowledge graph is controllable, and the model has good interpretability. Knowledge injection needs to solve two problems. The first one is physical alignment. Generally speaking, the entity's word embedding vector space in text and external knowledge are different. Therefore, it is necessary to align the entities so that they have the same word vector representation. The second is the problem of knowledge noise. Improperly injecting external knowledge can adversely affect the original semantics. Correct and reasonable injection methods can effectively reduce or even avoid noise generation. Liu proposed K-BERT [33], which is compatible with any pre-trained BERT model and does well in embedding space. For this paper, the authors designed sample selector 2 based on K-BERT. As shown in Figure 6, the K-BERT working process can be divided into three stages:

Stage 1: The nowledge layer. This stage injects domain knowledge into the original sentence sequence and outputs an abundant graph-structure sentence tree. It can be further subdivided into two processes: knowledge injection and sentence tree transformation. The external professional knowledge base KG equips the model with knowledge triple-tuple and then injects domain knowledge. The knowledge layer transforms primary sentences into abundant sentence trees.

First, perform a knowledge query (*K_Query*). Take the entities in the original sentence as the query, and traverse the KG. *K_Query* is a knowledge query function, S represents a query sentence, and K represents an external knowledge graph. External knowledge E can be expressed as:

$$E = K\_Query(S, K) \tag{16}$$

Its specific form is:

$$E = \{(w_i, r_{i0}, w_{i0}), \ldots, (w_i, r_{ik}, w_{ik})\} \tag{17}$$

where is a collection of the corresponding triples. The specific form of the sentence S is:

$$S = \{w_0, w_1, w_2, \ldots, w_n\} \tag{18}$$

The specific form of the sentence tree St is:

$$St = \{w_0, w_1, \ldots, w_i\{(r_{i0}, w_{i0}), \ldots.(r_{ik}, w_{ik})\}, \ldots, w_n\} \tag{19}$$

Injecting external knowledge according to the knowledge graph can be formulated as:

$$S_t = K\_Inject(S, K\_Query(S, K)) \tag{20}$$

where K_Inject is a function that injects knowledge into the sentence tree and keeps structure index information.

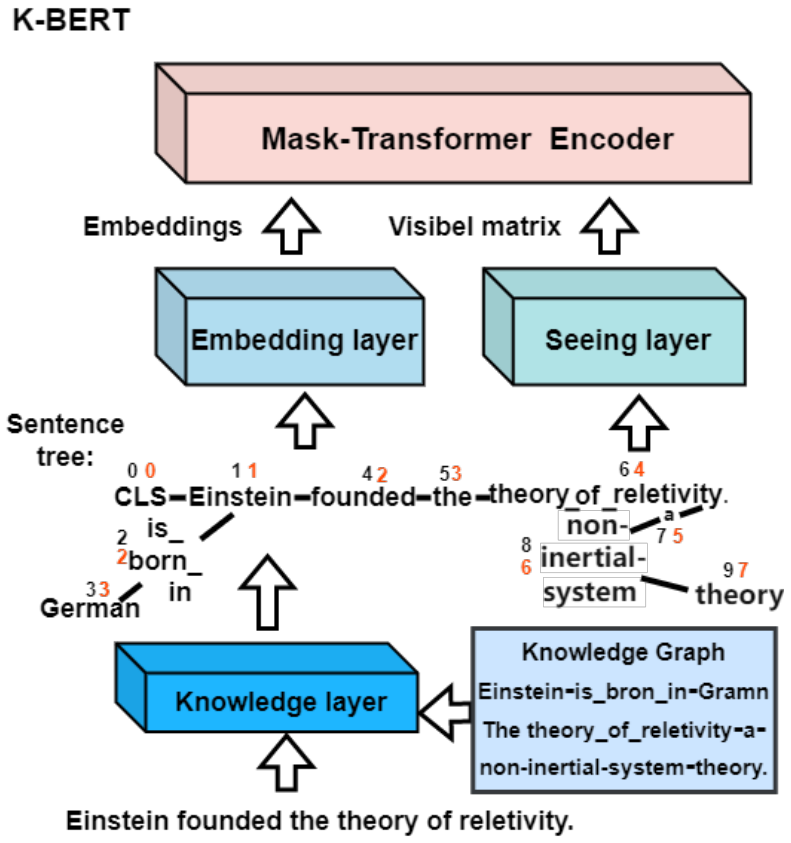

Figure 6. An illustration of K-BERT structure and the processing of a sentence sample. Black numbers represent hard position indices. Red numbers represent soft position indices.

The queried knowledge is placed directly after the entity, and the retrieval process is based on the visible matrix and soft position. In this paper, there are many branches in a sentence tree. Furthermore, its depth is limited to 1 for subsequent processing.

As shown in Figure 6, the input sentence X is: "Einstein created the theory of relativity". "Einstein" is associated with the tuple information "is born in Germany" in the knowledge graph. The relativity theory is related to "a non-inertial-system theory". After tuple injection, a sentence tree with a rich knowledge background is generated.

Stage 2: The embedding layer and seeing layer. BERT cannot handle graph-structured sentences. The role of the embedding layer is to solve this problem. The role of the seeing layer is to control the reference domain of external knowledge and avoid knowledge noise.

- Process 1:
  BERT can only process sentence input with sequence structure. The structural information will be lost if the sentence tree is tiled into a sequence. The visual matrix can convert the sentence tree into a sequence and save the structural information, and its principle is shown in Figure 7.
  The embedding representation includes three parts: token embedding, position embedding, and segment embedding. In the token embedding section, K-BERT flattens the tokens in the sentence tree into embedding sequences according to the hard position index. "[CLS](0) Einstein(1) is_born_in(2) German(3) founded(4) the_theory_of relativity(5) is_ proposed _by(6) Einstein(7)". Obviously, tiling causes losses in sentence structure and readability. For example, the positional index of "is born in(2)" and "founded(2)" are both (2), and they all follow "Einstein(1)". This information can be recovered by soft-position coding.
  As shown in Figure 7, "[CLS](0) Einstein(1) is born in(2) German(3) founded(2) the(3) theory of relativity(4) a(5) non-inertial-system(6) theory (7)". In the soft-position embedding section, K-BERT restores the graph structure and readability of the sentence tree through soft-position encoding. K-BERT uses soft-position embedding to map the readable information of sentence trees. Nevertheless, only using soft positions is not enough because it will make the model misunderstand that found(2) follows German(3), which will also lead to knowledge noise. In order to solve this problem, we can convert the soft-position coding into a visible matrix through the process 2 seeing layer. The role of this last section segment embedding is to identify sentence 1 and sentence 2. For the sentence pair, it is marked with a sequence of segment tags, $S_1, S_1, \ldots, S_1, S_2, S_2, \ldots, S_2$.
- Process 2:
  The seeing layer's role is to limit the field of view of the knowledge referenced by the entities. It can avoid false references to knowledge and the noise caused by mutual interference between knowledge. The visible matrix M is defined as:

$$M_{ij} = \begin{cases} 0, & W_i \ominus W_j \\ -\infty, & W_i \odot W_j \end{cases} \tag{21}$$

  where i and j are the hard-position index, W_i and W_j are co-branched, meaning that they are the same entity, and its distance is 0. On the contrary, its distance is negative infinity.

The seeing layer's core idea is to ensure that the original sentence entities and the injected background knowledge do not interfere. As shown in Figure 7, the sentence tree has nine tokens, which are mapped into a 9 * 9 visible matrix. White means that the two tokens in the corresponding position are visible to each other, and black means that they are invisible to each other. The black cell in column 3, row 9, means German(3) cannot see "theory(9)". Similarly, [German] is invisible to [CLS], but through [Einstein] it indirectly acts on [CLS], thereby reducing knowledge noise.

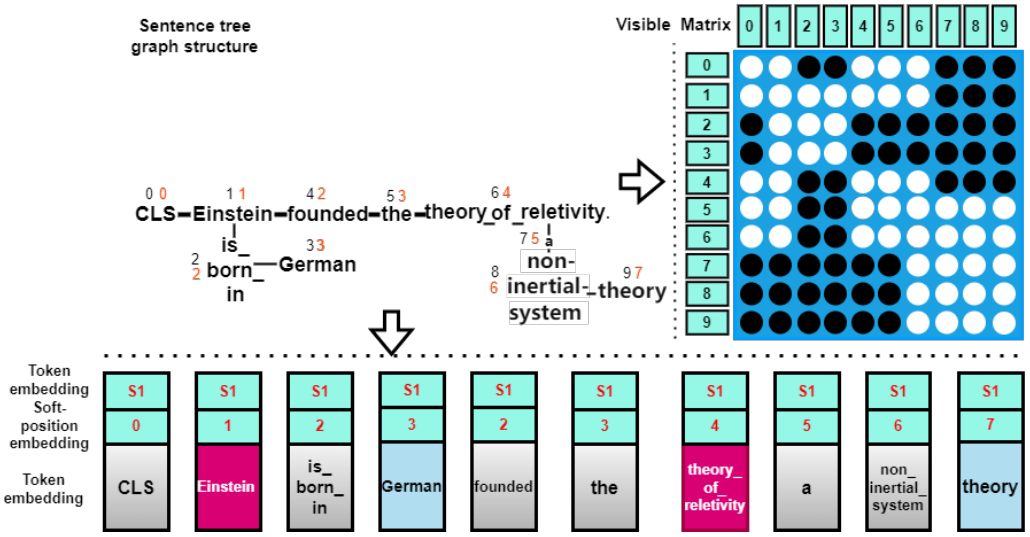

Figure 7. The principle of mapping sentence trees to transition matrices and the process of converting to embedding representations. In the graph structure of the sentence, the black numbers are hard position indices, and the red numbers are soft position indices. (1) For token embedding, the sentence tree is serialized into token embedding according to the hard-position index; (2) for the soft-position embedding, use it as position embedding along with the token embedding; (3) in segment embedding, the all the first tokens in a sentence are recorded as an "A"; (4) in the visible matrix, white means visible and black means invisible. For example, the cell in column 3, row 9, is black.

Stage 3: The mask-transformer encoder processing. Traditional transformer encoders cannot accept a visible matrix as input. The role of the mask-transformer encoder is to introduce the structural information in the sentence tree into the model. It is a stack of multiple mask self-attention blocks. The formula is as follows:

$$Q^{i+1}, K^{i+1}, V^{i+1} = h^i W_q, h^i W_k, h^i W_v \tag{22}$$

$$S^{i+1} = softmax(\frac{Q^{i+1}K^{i+1^T} + M}{\sqrt{d_k}}) \tag{23}$$

$$h^{i+1} = S^{i+1}V^{i+1} \tag{24}$$

W_q, W_k, and W_v are the parameters that the model needs to learn, the mask self-attention block of the hidden state; d_k is the scaling factor.

4. Experimental Results and Experiment Analysis

4.1. Datasets and Evaluation Settings

FewRel [34] is a large-scale supervised dataset suitable for few-shot relation classification tasks. It was derived from Wikipedia through semi-supervision to build the candidate set. Then, 70,000 instances are manually cleaned and tlabeled. It contains 100 relationships with 700 instances of each relationship. FewRel is accurate in labeling, has a large amount of data, and is suitable for distant supervision. The corpus used for iterative training contains 899,997 instances, and the entities of each instance are labeled. In addition, the corpus contains 464,216 entity pairs, and part of the data is collected from Wikipedia. This paper divides FewRel into a training set (64 relations), a validation set (20 relations), and a test set (16 relations). The training set is split into T1 and T2. There are three steps for the evaluation of each iteration:

- Train the ORES model on the training set T1, namely, sample selector 1 and instance selector 2.

- Randomly sample a relation r and its K instances from the training set T2, the validation set, and the test set to form a query set Q. Moreover, randomly sample a sentence SR from Q. Its entity–pair relation is recorded as R.
- Input SR as a query sentence into the model and let the classifier judge whether SR and Sr, sampled from the validation set and the test set, are the same types of relationship in a binary classification manner. A new relation extraction classifier is obtained by evaluating and tuning the relation classifier.

4.2. Experiments

The steps of few-shot relation learning are: first, pre-train the encoder, then pre-train the sample selector K-tBERT; then, iteratively train the relation classifier in Graph Snowball and output the model performance evaluation value.

We conducted three sets of experiments on the settings of each baseline model. Each set of experiments sampled 5 seed relation instances, 10 seed relation instances, and 15 seed relation instances of relation r, respectively. The first experiment followed the one-way five-shot setup. The corpus used for multiple iterations contained 899,997 relation instances and 464,216 entity pairs during each set of experiments. BREDS [35] is an advanced version of the original Snowball [20]. Relation Siamese network (CNN) is a network structure that uses CNN as the encoder. It adopts CNN [36] to learn an embedding function, which can map sentence pairs in the input space to a new embedding space and then calculate the similarity in the embedding space.

Tables 1–3 show ORES's results for few-shot relation learning. We also list the experimental data for several previous baseline models. Since these models do not have settings for extracting new relations, this paper does not make specific comparisons and evaluations. It can be found that the performance of the ORES model is better. Our experimental results are mainly compared with Neural Snowball with the setting of extracting new relations. Furthermore, we set up the compatibility experiments, comparing the experimental results of sample selector 2 with t-BERT and K-BERT encoding settings, respectively. Sample selector 2 adopts the encoding setting of t-BERT from our previous work, Topic Snowball.

Table 1. The 5-shot experimental results on various previous baseline models.

Model	Precision (%)	Recall (%)	F1 (%)
BREDS	33.71	11.89	17.58
Fine-tuning (CNN)	46.90	9.08	15.22
Distant Supervision (CNN)	44.99	31.06	36.75
Fine-tuning (BERT)	50.85	16.66	25.10
Distant Supervision (BERT)	38.06	51.18	43.66
Neural Snowball (BERT)	56.87	40.43	47.26
Our Topic Snowball (tBERT)	54.84	69.25	53.13
Our ORES (t-KBERT)	55.69	69.78	61.94

From the five-shot experimental setup in Table 1, it can be seen that the recall and F1 value of Topic Snowball were significantly improved compared to Neural Snowball. After introducing the topic model, the diversity of the filtered sentences was significantly improved. However, the precision was unsatisfactory. This indicates that adding the topic model sacrificed some accuracy but increased the diversity of instances. Comparing Topic Snowball and ORES, it can be seen that the model's accuracy is improved after injecting external knowledge. This indicates that injecting external knowledge is beneficial to improving the accuracy of word embedding vector representation. In addition, the accuracy of ORES was lower than that of Neural Snowball, which was related to the selection of external knowledge sources.

Table 2 shows the experimental results for 10 seed instances. Compared with Neural Snowball, the recall rate of Topic Snowball was improved. This also indicates that the topic model is effective. The decrease in precision indicates that the accuracy of Topic

Snowball is related to the setting of topic weights. A more accurate algorithm is needed for the reasonable allocation of topic and semantic weights in similarity detection. From the accuracy rates of ORES and Topic Snowball, it can be seen that the word vector representation of entities was more accurate after injecting external knowledge. Using K-BERT encoding will reduce the influence of topic weights on similarity detection. Compared with Topic Snowball, the overall performance of ORES was improved. Compared with Neural Snowball, ORES had a higher recall, lower precision, and lower F1 value. Adding a topic model was beneficial to diversity, but it slightly reduced the accuracy. The problem can be further studied by optimizing the topic model and the appropriate setting of topic weights.

Table 2. The 10-shot experimental results on various previous baseline models.

Model	Precision (%)	Recall (%)	F1 (%)
BREDS	28.29	17.02	21.25
Fine-tuning (CNN)	47.58	38.36	42.28
Distant Supervision (CNN)	42.48	48.64	45.35
Fine-tuning (BERT)	5.87	55.19	57.43
Distant Supervision (BERT)	38.45	76.12	51.09
Neural Snowball (BERT)	60.50	62.20	61.34
Our Topic Snowball (tBERT)	49.28	78.18	56.33
Our ORES (t-KBERT)	51.47	75.01	61.05

Table 3. The 15-shot experimental results on various previous baseline models.

Model	Precision (%)	Recall (%)	F1 (%)
BREDS	25.24	17.96	20.99
Fine-tuning (CNN)	74.70	48.03	58.46
Distant Supervision (CNN)	43.70	54.76	48.60
Fine-tuning (BERT)	81.60	58.92	68.43
Distant Supervision (BERT)	35.48	80.33	49.22
Neural Snowball (BERT)	78.13	66.87	72.06
Our Topic Snowball (tBERT)	50.08	79.88	58.16
Our ORES (t-KBERT)	56.23	78.15	66.74

Table 3 shows the experimental results from 15-shot setting. Continuing the previous trend, the recall rate of ORES was improved, but the precision and F1 value were lower than with Neural Snowball. Compared with Topic Snowball, the performance of ORES was improved. This shows that injecting external knowledge is effective.

To further analyze the iterative process of ORES, we find that the classifier obtained from the three experimental settings showed similar trends in performance. The performance of the ORES was not good enough in the 10-seed and 15-seed settings. However, few-shot focuses on scenarios with fewer instances. For the five-shot setting, the performance of the ORES improved significantly. In general, the model's performance can be improved by increasing the number of seed examples. However, the low-resource setting that is stronger for small-sample relation extraction can better test the model's generalization ability. To further evaluate the performance of ORES with few instances, we set up experiments with fewer instances.

As shown in Figure 8, ORES has higher precision and recall than Neural Snowball (NS) with fewer shots. This indicates that injecting external knowledge and introducing topic models are effective for few-shot relation learning. As the number of seed instances increases, the accuracy of NS begins to be higher than that of ORES after five shots. This is related to the accumulation of noise in the data mined by distant supervision. When the data noise increases, it is easy to overfit.

As shown in Figure 9, when there were fewer seed instances, ORES's accuracy was higher, indicating that the topic similarity of the instances is more valuable than the

semantic similarity. ORES worked better when there were few samples, but as the number of seed instances increased, the impact of semantic similarity on classifier performance increased. If the topic weight is too large, it will affect the further improvement of the classifier performance.

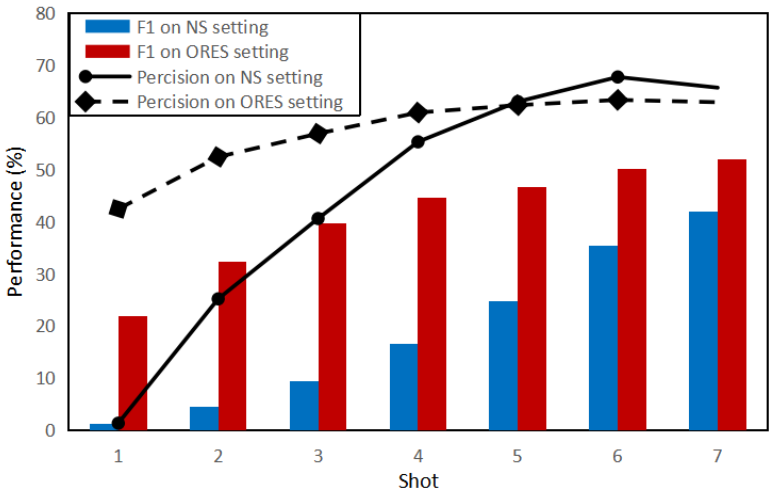

Figure 8. Evaluation results on few-shot of Neural Snowball and Topic Snowball. The horizontal axis are numbers of seed sentences, one relation type.

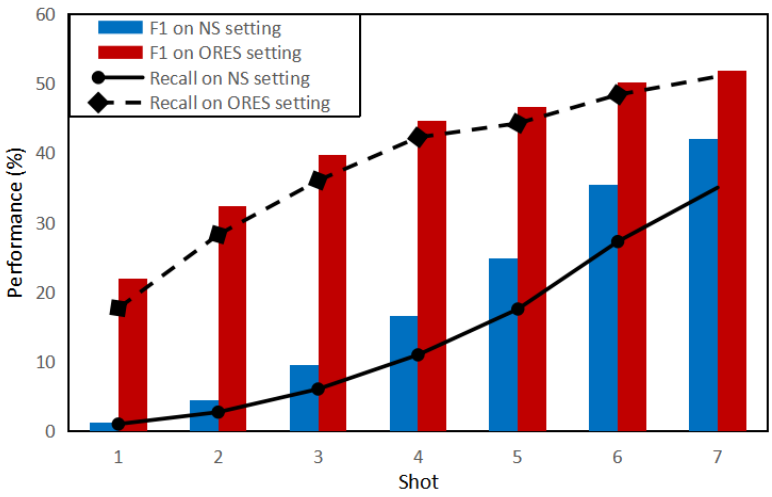

Figure 9. Demonstrates the development of evaluation results as the number of shots grows.

5. Conclusions and Future Work

This paper proposes a new relation extraction system, ORES, for Web text information. It can automatically extract and label new relational data. It is expected to contribute great value to the field of intelligent education. We designed new sample selectors based on t-BERT and K-BERT. When the number of initial seed sentences was low, ORES performed significantly better than Neural Snowball. When the number of seed sentences increased, ORES's performance improvement was not satisfactory enough compared to Neural Snowball. In the comparison experiment with Topic Snowball, we found that the K-BERT setting of sample selector 2 could improve the accuracy of entity representation in the word embedding process and reduce the weight of topic information. The experiments show that injecting external knowledge effectively improves classifier accuracy and overall performance. The few-shot relational learning approach proposed in this paper is well suited for open-domain scenarios such as Web text. Amid today's exponential growth in knowledge, it can continuously mine data from Web texts and iteratively discover new relationship types, which have broad application prospects.

In the future, there are two directions worth exploring deeply. (1) We can explore how to calculate topic similarity and semantic similarity reasonably. The sample selector adds the weight of the topic when encoding sentence information. Its essence is to broaden the similarity threshold. However, the similarity domain extension also affects accuracy. Thus, the manner in which the weight of the topic vector and the semantic representation vector are balanced is a valuable research direction.

(2) Although distant supervision to obtain training samples is very effective, it is inevitably accompanied by data noise. With the number of seed sentences increasing, instances of distant supervision matching will have more wrong label problems, which may lead to overfitting. Therefore, reducing the data noise of distant supervision is a valuable research direction. succeeding in representing a negative training method performance well. In the future, we will build a negative indicator based on BTOD to improve ORES's performance.

Author Contributions: Conceptualization, H.L. and B.L.; data curation, H.L.; methodology, B.L.; visualization, B.L.; writing—original draft, H.L.; writing—review and editing, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available upon request by contacting the corresponding author.

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

References

1. Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; Li, J. A Unified MRC Framework for Named Entity Recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 5849–5859.
2. Grover, C.; Haddow, B.; Klein, E.; Matthews, M.; Neilsen, A.L.; Tobin, R.; Wang, X. Adapting a Relation Extraction Pipeline for the BioCreAtIvE II Tasks. 2007. Available online: https://www.research.ed.ac.uk/en/publications/adapting-a-relation-extraction-pipeline-for-the-biocreative-ii-ta (accessed on 5 May 2022).
3. Miwa, M.; Sasaki, Y. Modeling Joint Entity and Relation Extraction with Table Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1858–1869.
4. Yates, A.; Cafarella, M.; Banko, M.; Etzioni, O.; Broadhead, M.; Soderland, S. TextRunner: open information extraction on the web. In Proceedings of the Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), Rochester, NY, USA, 22–27 April 2007; pp. 25–26.
5. Mausam.; Schmitz, M.; Bart, R.; Soderland, S.; Etzioni, O. Open language learning for information extraction. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 12–14 July 2012; pp. 523–534.
6. Fader, A.; Soderland, S.; Etzioni, O. Identifying relations for open information extraction. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 27–31 July 2011; pp. 1535–1545.
7. Yahya, M.; Whang, S.; Gupta, R.; Halevy, Y.A. ReNoun: Fact Extraction for Nominal Attributes. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 325–335.
8. Corro, D.L.; Gemulla, R. ClausIE: Clause-based open information extraction. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 355–366.
9. Angeli, G.; Premkumar, J.J.M.; Manning, D.C. Leveraging Linguistic Structure For Open Domain Information Extraction. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 1, pp. 344–354.
10. Stanovsky, G.; Ficler, J.; Dagan, I.; Goldberg, Y. Getting More Out Of Syntax with PropS. *arXiv* **2016**, arXiv:1603.01648.
11. Cetto, M.; Niklaus, C.; Freitas, A.; Handschuh, S. Graphene: Semantically-Linked Propositions in Open Information Extraction. *arXiv* **2018**, arXiv:1807.112.
12. Gao, T.; Han, X.; Xie, R.; Liu, Z.; Lin, F.; Lin, L.; Sun, M. Neural Snowball for Few-Shot Relation Learning. *Natl. Conf. Artif. Intell.* **2020**, *34*, 7772–7779. [CrossRef]
13. Efron, B.; Tibshirani, J.R. An Introduction to the Bootstrap. In *Technometrics*; CRC Press: Boca Raton, FL, USA, 1994.
14. Bromley, J.; Guyon, I.; Lecun, Y.; Säckinger, E.; Shah, R.; Moore, C.; Säckinger, E.; Shah, R. Signature Verification Using a Siamese Time Delay Neural Network. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann: Burlington, MA, USA, 1994; pp. 737–744.

15. Xu, H.; Tianyu, G.; Yankai, L.; Hao, P.; Yaoliang, Y.; Chaojun, X.; Zhiyuan, L.; Peng, L.; Maosong, S.; Jie, Z. More Data, More Relations, More Context and More Openness: A Review and Outlook for Relation Extraction. *arXiv* **2020**, arXiv:2004.03186.
16. Zhong, Z.; Chen, D. A Frustratingly Easy Approach for Joint Entity and Relation Extraction. *arXiv* **2020**, arXiv:2010.12812.
17. Sun, A.; Grishman, R.; Sekine, S. Semi-supervised relation extraction with large-scale word clustering. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, ON, USA, 19–24 June 2011; pp. 521–529.
18. Felsenstein, J. Confidence limits on phylogenies: an approach using the bootstrap. *Evolution* **1985**, *39*, 783–791. [CrossRef] [PubMed]
19. Brin, S. Extracting Patterns and Relations from the World Wide Web. In Proceedings of the International Workshop on the World Wide Web and Databases, Valencia, Spain, 27–28 March 1998; pp. 172–183.
20. Agichtein, E.; Gravano, L. Snowball: Extracting relations from large plain-text collections. In Proceedings of the Fifth ACM Conference on Digital Libraries, San Antonio, TX, USA, 2–7 June 2000; pp. 85–94.
21. Qin, P.; Xu, W.; Wang, Y.W. Robust Distant Supervision Relation Extraction Via Deep Reinforcement Learning. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 2137–2147.
22. Zeng, D.; Liu, K.; Chen, Y.; Zhao, J. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015.
23. Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; Sun, M. Neural Relation Extraction With Selective Attention Over Instances. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Volume 1, pp. 2124–2133.
24. Ji, G.; Liu, K.; He, S.; Zhao, J. Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, Berlin, Germany, 7–12 August 2017; pp. 3060–3066.
25. Jia, W.; Dai, D.; Xiao, X.; Wu, H. ARNOR: Attention Regularization based Noise Reduction for Distant Supervision Relation Classification. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1399–1408.
26. Liang, C.; Yu, Y.; Jiang, H.; Er, S.; Wang, R.; Zhao, T.; Zhang, C. BOND: BERT-Assisted Open-Domain Named Entity Recognition with Distant Supervision. In Proceedings of the KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, 6–10 July 2020; pp. 1054–1064.
27. Wang, Y.; Yao, Q.; Kwok, J.; Ni, M.L. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Comput. Surv.* **2020**, *53*, 1–34. [CrossRef]
28. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
29. Chen, W.; Matusov, E.; Khadivi, S.; Peter, J.T. Guided Alignment Training for Topic-Aware Neural Machine Translation. *arXiv* **2016**, arXiv:1607.01628.
30. Peinelt, N.; Nguyen, D.; Liakata, M. tBERT: Topic Models and BERT Joining Forces for Semantic Similarity Detection. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7047–7055.
31. Li, H.; Liu, B. Knowledge Extraction: A Few-shot Relation Learning Approach. In Proceedings of the 2022 International Conference on Machine Learning and Knowledge Engineering (MLKE), Guilin, China, 25–27 February 2022; pp. 261–265. [CrossRef]
32. Nickel, M.; Murphy, K.; Tresp, V.; Gabrilovich, E. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* **2016**, *104*, 11–33. [CrossRef]
33. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; Wang, P. K-Bert: Enabling Language Representation With Knowledge Graph. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, the Thirty-Second Innovative Applications of Artificial Intelligence Conference and the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 2901–2908.
34. Han, X.; Zhu, H.; Yu, P.; Wang, Z.; Yao, Y.; Liu, Z.; Sun, M. FewRel: A Large-Scale Supervised Few-shot Relation Classification Dataset with State-of-the-Art Evaluation. *arXiv* **2018**, arXiv:1810.10147.
35. Batista, S.D.; Martins, B.; Silva, J.M. Semi-Supervised Bootstrapping of Relationship Extractors with Distributional Semantics. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015.
36. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.