Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 4929674, 9 pages
https://doi.org/10.1155/2018/4929674

Research Article

A Semieager Classifier for Open Relation Extraction

Peiqian Liu [1,2] **and Xiaojie Wang** [1]

[1] *School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *School of Computer Science, Henan Polytechnic University, Henan, Jiaozuo 454003, China*

Correspondence should be addressed to Peiqian Liu; liupeiqian@hpu.edu.cn

Received 12 May 2018; Revised 8 September 2018; Accepted 11 October 2018; Published 28 October 2018

Academic Editor: Giuseppe D'Aniello

A variety of open relation extraction systems have been developed in the last decade. And deep learning, especially with attention model, has gained much success in the task of relation classification. Nevertheless, there is, yet, no research reported on classifying open relation tuples to our knowledge. In this paper, we propose a novel semieager learning algorithm (SemiE) to tackle the problem of open relation classification. Different from the eager learning approaches (e.g., ANNs) and the lazy learning approaches (e.g., kNN), the SemiE offers the benefits of both categories of learning scheme, with its significantly lower computational cost ($O(n)$). This algorithm can also be employed in other classification tasks. Additionally, this paper presents an adapted attention model to transform relation phrases into vectors by using word embedding. The experimental results on two benchmark datasets show that our method outperforms the state-of-the-art methods in the task of open relation classification.

1. Introduction

Traditionally, Information Extraction (IE) is the task of collecting structured information automatically from large size of unstructured data by learning an extractor from labeled training examples for each target relation [1–3]. This approach to IE cannot scale to corpora with a large number of target relations or with no prespecified target relations. In response, researchers at the University of Washington pioneered a new paradigm of open relation extraction (ORE), which enables the extraction of arbitrary relations from sentences by automatic identification of relation phrases, obviating the restriction to a prespecified vocabulary.

In recent years, a lot of ORE systems have been presented in the literature. The first ORE system is TEXTRUNNER [4] which learns a CRF on self-supervised training data constructed over Penn Tree Bank. The CRF can work on corpus not seen at all in training data, since it uses only unlexicalized features. Reverb improves over TEXTRUNNER via carefully designed linguistic patterns for relation phrases of English text [5, 6]. Reverb reveals that about 85% of verb-based relation phrases can be expressed with a simple regular expression (see Box 1). OLLIE learns unlexicalized pattern templates on bootstrapped training data [7]. Not only verb-based patterns but also OLLIE can learn noun-based and even some inferential relation patterns. RELNOUN is a rule-based open relation extraction system [8] aiming at extracting nominal relations. It encodes various noun-mediated patterns and pays special attention to compound relational nouns.

By themselves, the generated triples from the ORE systems appear of little interest, unless they are placed in the context of some particular downstream tasks, such as the semantic Web, event schema induction, sentence similarity, text comprehension, and Q/A system [9]. In this respect, we can consider the open relation extraction as only a prior step of a semantic analysis process [10]. ORE systems are not restricted on predefined relations in their extraction process and can extract all types of relations found in a text. However, after a set of tuples are generated by an ORE system, we can classify these tuples with a classifier which is pretrained on a labeled dataset. This is quiet important to some downstream applications, since some applications are interested in only a few types of these relations (e.g., supplier relationship or members-employees relationship). In addition, some other applications require the relations to be mapped to the relations in a particular ontology. For example, for TACKBP'2013 an NLP expert created a

$$V \mid VP \mid VW * P$$
V = verb particle? adv?
W = (noun | adj | adv | pron | det)
P = (prep | particle | inf. Marker)

Box 1: A regular expression for the syntactic constraint on relation phrases.

rule-based extractor for 41 relations of interest from open tuples [11]. The extractor obtains a precision of about 0.8 but low recall. All these applications suggest the necessity of classifying open tuples. And this paper is targeted at the problem of open relation classification based on the research of traditional relation classification.

A large number of approaches have been explored for traditional relation classification over the years. Recently, neural network-based approaches have achieved significant improvement over traditional methods based on human-designed features [12]. Earlier neural networks for relation classification are usually based on one-layer Convolutional Neural Networks or recurrent networks. They may fail to perform in exploring the potential representation space at different abstraction levels [13]. The performance of supervised approaches strongly depends on the quality of the designed features. With the recent improvement in deep neural network (DNN), many researchers have presented unsupervised methods for automatic feature learning. Gated recurrent networks with long short-term memory (LSTM) were introduced to relation classification by Xu, Y et al. [14]. And Convolutional Neural Networks (CNNs) were proposed by Zeng to deal with the problem [15]. Additionally, the common Softmax loss function was replaced with a ranking loss in the CNN model in [16]. A negative sampling method based on CNNs was designed in [17]. From the viewpoint of model ensemble, algorithms incorporating CNNs with Recurrent Neural Networks (RNNs) have been proposed for relation classification [18, 19]. Recently, attention mechanism based on LSTM and word vectors was presented by Zhang and Zheng [20]. Additionally, much effort has been invested in relational learning methods that can scale to large knowledge bases. The best performing neural-embedding models are NTN [21] and Bordes models (TransE and TATEC) [22, 23], which extend the traditional relation classification task to semantic relation classification. Different from those approaches built over lexical and distributional word vector features, Siamak B et al. proposed a model using the combination of large commonsense knowledge bases of binary relations for the composite semantic relation classification problem [24].

Nevertheless, these approaches for traditional relation classification are unsuitable for targeting the variety and complexity of open relation types on the Web, since ORE systems are strongly in favor of speed and these approaches are time consuming. Moreover, there is no research reported on classifying open tuples by now, although it is of great importance to the downstream applications. Thus, it is necessary to develop a methodology appropriate to the problem of open tuple classification.

The machine learning algorithms for training a classifier can be organized in two categories: the eager learning and the lazy learning. A well-known disadvantage of eager learning is the high time complexity in the training process, and lazy learning suffers from its high space complexity, since lazy learning must store all the training examples. To overcome the disadvantages of eager and lazy learning but still preserve the benefits of the two models, this paper proposes a semieager learning algorithm to tackle the task for open tuple classification. The proposed algorithm is quite efficient. Its time complexity is $O(n)$ and space complexity is $O(k)$, with n being the number of training instances and k being the number of categories.

Recently, attention mechanism gets a wide range of applications. For NLP, it is capable of automatically concentrating on valuable words for targeted task. Inspired from this, we present a novel method to calculate representation for relation phrases, via word Vector-Sum operation weighted on attention weights. The attention weights are based on the information quantity of the words within the relation phrase.

In summary, the main contributions of this paper are as follows.

(1) A novel semieager learning algorithm is presented to classify open relation tuples. The proposed algorithm offers the benefits of both the eager learning and lazy learning scheme, with its significantly lower computational cost. This algorithm can also be employed in other classification tasks.

(2) Additionally, this paper presents an adapted attention model to transform relation phrases into vectors by using word embedding.

(3) Experiments show that the new algorithms achieve better performance compared to some recently presented methods.

The remainder of the paper is organized as follows. In Section 2, we review related work about open relation extraction and traditional relation classification. The derivation of the proposed semieager learning approach is presented in Section 3, as well as the description of the adapted attention model. In Section 4, we describe in detail the setup of experimental evaluation and the experimental results. Finally, we present the conclusion in Section 5.

2. Related Works

Reverb is the second generation of open relation system. Given a POS-tagged and NP-chunked sentence s as input, the algorithm returns an extraction triple with the form (NP, VP, NP). The VP must satisfy the lexical constraint and syntactic constraint as shown in Box 1, and the NPs are the nearest noun phrases around the VP [5].

In the research area of relation classification, deep neural networks have been widely used in recent years, since deep architectures can automatically learn underlying features. While CNN is not suitable for learning long-distance semantic information [15], Zhang and Wang proposed a bidirectional RNN [25] to learn patterns of relations from raw text data. To overcome problem of vanishing gradient in RNN, SDP-LSTM model was proposed by Yan et al. [14]. Recently, a novel neural network Att-BLSTM was proposed for relation

classification [26]. This model utilizes BLSTM with neural attention mechanism to capture the most important semantic information in a sentence, without using extra knowledge and NLP systems. The Att-BLSTM transforms all words to word vectors, forming a simple but competing model. Similarly, EAtt-BiGRU presents an entity-pair-based attention mechanism for solving relation classification, which utilizes entity pair information as a priori knowledge to adaptively generate attention weights based on word vectors [27].

Distributed representations of words in a vector space have achieved considerable success in a wide range of NLP tasks [28, 29], including applications to automatic speech recognition and machine translation [30, 31].

Recently, Mikolov et al. introduced two novel algorithms for computing word vectors based on large unlabeled datasets [32, 33]. The first architecture is the continuous bag-of-words model (CBOW), while the second one is named as the Skip-gram model. Given the surrounding words, the CBOW model predicts the current word, and the Skip-gram model predicts the surrounding words based on the current word. Given a word sequence $w_1, w_2, w_3, \ldots, w_T$, which is a sentence or document, the training objective of the CBOW model can be presented as the maximum of the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leqslant j\leqslant c, j\neq 0}\log p\left(w_{t+j} \mid w_t\right) \tag{1}$$

Similarly, the objective of the Skip-gram model is

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leqslant j\leqslant c, j\neq 0}\log p\left(w_t \mid w_{t+j}\right) \tag{2}$$

where $\boldsymbol{c}$ is the size of the training context around the center word w_t. Larger $\boldsymbol{c}$ means more training instances and thus can result in a higher accuracy, but at the cost of training time.

In order to classify the relations with the proposed semieager learning approach, phrase-level feature vectors with the same size must be generated in advance. In this paper, we employ Vector-Sum, Max-Pooling, and an adapted attention model to calculate feature vectors for relation phrases. The adapted attention model is one of our contributions in this paper. The novel model is based on word vector and inspired from the attention mechanism in deep neural network architecture.

3. The Proposed Algorithms

Eager learning methods construct a general, explicit description of the target function when training examples are provided, such as Artificial Neural Networks, Support Vector Machine, and Conditional Random Field algorithm. The disadvantages of eager learning includes the following: (a) the training process for these models usually requires high time cost, up to several hours and even days. For instance, the computational cost for SVMs is $O(n^3)$, with n being the number of training instances; (b) the eager learning approach is bounded to the problem of information loss, which may lead to a high potential risk of overfitting or underfitting; (c) these models are influenced mainly by the global distribution on the whole dataset rather than by the local behavior of unknown prediction targets. However, the local behavior is very important for the convergence of machine learning models [34].

In contrast to eager leaning methods, a delayed, or lazy, learning algorithm simply stores the training examples. Generalizing beyond these examples is postponed until a new instance must be classified. A key advantage of this kind of methods is that instead of estimating the entire instance space these methods can estimate it locally and differently for each new instance to be classified. A typical lazy learning method is k-Nearest Neighbor learning or the locally weighted regression algorithm. In order to make prediction for a new instance, lazy learning will calculate its distance to each training example. Although lazy learning needs no training effort, the time complexity for prediction is $O(n)$, and here n is the number of training examples. This is the main disadvantage of lazy learning.

To sum up, the eager learning approach suffers the problems of concept drifting and information loss, since it computes a global model before seeing the prediction query. And the lazy learning approach suffers from simplistic predicting methods, although it can commit much richer sets of hypotheses (models) from the data. To overcome the disadvantages of eager and lazy learning but still preserve the benefits of the two models, a semieager learning algorithm is proposed in this paper. Unlike the lazy learning algorithm, the proposed model stores only "center point" for each class after the training process. The training time complexity is $O(n)$, and both the time complexity and space complexity for prediction are $O(k)$. Here n is the number of training instances and k is the number of categories. We call this new approach the "SemiE" learning approach

3.1. The Semieager Learning Algorithm. (1) Consider the input space $\mathrm{X} \subset R^n$ is a set of n-dimensional vectors, and the output space is a set of class labels, $\mathrm{Y} = \{c_1, c_2, \ldots, c_k\}$, with $c_i \in Z$. Assume $\mathrm{x} \in X$ is a feature vector and $\mathrm{y} \in Y$ is the corresponding class label. Let P(X, Y) be the joint probability distribution over X and Y, and the training dataset is $\mathrm{T} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where x_i is an instance from X, and y_i is the corresponding class label from Y.

Assuming $\mathrm{S} = S_1 \cup S_2 \cup \cdots \cup S_k$, with $S_j = \{(x_i, y_i) \mid y_i = c_j\}$, $\mathrm{i, j} = 1, 2, \ldots, \mathrm{N}$, is a partition of set T, the task of the learner is to learn the prior probability

$$\mathrm{P}\left(\mathrm{Y} = c_j\right), \quad \mathrm{j} = 1, 2, \ldots, \mathrm{k} \tag{3}$$

and the conditional probability distribution

$$\mathrm{P}\left(\mathrm{X} = x_i \mid \mathrm{Y} = c_j\right), \quad \text{with } \mathrm{i} = 1, 2, \ldots, \mathrm{N},\ \mathrm{j} = 1, 2, \ldots, \mathrm{k} \tag{4}$$

Thus, we can get the posterior probability

$$\mathrm{P}\left(\mathrm{Y} = \mathrm{c_j} \mid \mathrm{X} = \mathrm{x_i}\right) = \frac{P\left(\mathrm{X} = \mathrm{x_i} \mid Y = c_j\right) P\left(Y = c_j\right)}{\sum_{m=1}^{k} P\left(\mathrm{X} = \mathrm{x_i} \mid Y = c_m\right) P\left(Y = c_m\right)} \tag{5}$$

During prediction, given a certain input x_i, the class label with the highest posterior probability will be outputted:

$$C = \arg\max_j \frac{P\left(\mathrm{X} = \mathrm{x_i} \mid Y = c_j\right) P\left(Y = c_j\right)}{\sum_{m=1}^{k} P\left(\mathrm{X} = \mathrm{x_i} \mid Y = c_m\right) P\left(Y = c_m\right)} \quad (6)$$

Notice that the denominator is a constant independent of c_j. So this item can be dropped, yielding

$$C = \arg\max_j P\left(\mathrm{X} = \mathrm{x_i} \mid Y = c_j\right) P\left(Y = c_j\right) \quad (7)$$

(2) Parameter Estimation. By using the maximum likelihood estimation method, the prior probability can be calculated as

$$\mathrm{P}\left(\mathrm{Y} = \mathrm{c_j}\right) = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} I\left(y_i = c_j\right)}{N}, \quad \mathrm{j} = 1, 2, \ldots, \mathrm{k}. \quad (8)$$

Here N is the number of samples in the training set, and the indicator function is

$$I\left(u = v\right) = \begin{cases} 1, & if\ u = v \\ 0, & otherwise \end{cases} \quad (9)$$

The posterior probability $\mathrm{P}(\mathrm{X} = x_i \mid \mathrm{Y} = c_j)$ is estimated as follows.

According to the central limit theorem, when the number of samples is large enough (N>30), x_i within the training dataset T obeys a normal distribution with μ mean and variance σ^2. Similarly, in each subset $S_j = \{(x_i, y_i) \mid y_i = c_j\}$, x_i obeys a normal distribution with variance σ^2 centered around μ_j, if $|S_j| > 30$. Thus,

$$P\left(\mathrm{X} = \mathrm{x_i} \mid Y = c_j\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(x_i - \mu_j\right)^2\right) \quad (10)$$

Substituting the above equation into (7) yields

$$\begin{aligned} C &= \arg\max_j P\left(\mathrm{X} = \mathrm{x_i} \mid Y = c_j\right) P\left(Y = c_j\right) \\ &= \arg\max_j \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(x_i - \mu_j\right)^2\right) P\left(c_j\right) \end{aligned} \quad (11)$$

The first term in the expression is a constant independent of c_j and therefore can be discarded, yielding

$$C = \arg\max_j \exp\left(-\frac{1}{2\sigma^2}\left(x_i - \mu_j\right)^2\right) P\left(c_j\right) \quad (12)$$

Because $\ln P$ is a monotonic function of P, maximizing $\ln P$ also maximizes P:

$$\begin{aligned} C &= \arg\max_j \ln \exp\left(-\frac{1}{2\sigma^2}\left(x_i - \mu_j\right)^2\right) P\left(c_j\right) \\ &= \arg\max_j \left(\ln P\left(c_j\right) - \frac{1}{2\sigma^2}\left(x_i - \mu_j\right)^2\right) \\ &= \arg\min_j \left(\frac{1}{2\sigma^2}\left(x_i - \mu_j\right)^2 - \ln P\left(c_j\right)\right) \\ &= \arg\min_j \left(\left(x_i - \mu_j\right)^2 - 2\sigma^2 \ln P\left(c_j\right)\right) \end{aligned} \quad (13)$$

Under certain condition that a priori probabilities of all classes are equal, saying that $\mathrm{P}(c_i) = \mathrm{P}(c_j)(i \neq j)$, we can get

$$\mathrm{C} = \arg\min_j \left(x_i - \mu_j\right)^2 \quad (14)$$

where μ_j represents the j-th "class center". Thus, to classify instance x_i, we should determine the class centers for each class. Let $x_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)})$; the following holds:

$$\begin{aligned} \mu_j &= \frac{\sum_i I\left(y_i = c_j\right) x_i}{\sum_i I\left(y_i = c_j\right)} \\ &= \frac{\sum_i I\left(y_i = c_j\right)\left(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}\right)}{\sum_i I\left(y_i = c_j\right)} \end{aligned} \quad (15)$$

where $I(\cdot)$ is the indicator function described in (9).

The semieager algorithm is summarized in Algorithm 1. The inputs of this algorithm include a set of training instances and their class labels. The algorithm outputs frequencies, class centers, and regularization terms for each class, respectively.

Remarks on semieager learning algorithm:

(a) The training time complexity of the algorithm is $O(n)$, and the prediction time complexity is $O(k)$. It is quite efficient when it is provided as a sufficiently large set of training data.

(b) The SemiE supports Incremental Learning.

(c) $-2\sigma^2 \ln P(c_j)$ is a regularization term. If a data point to be classified has the same distance to all class centers, the class label with the highest frequency will be assigned to it.

(d) It is robust to noisy training examples and to the irrelevant attributes for classification. A small number of noisy instances in a class will not significantly influence the center of the class.

(e) The SemiE is suitable only for the learning tasks with relatively fewer categories (k≪N). Otherwise, if k has the same quantitative level as N, SemiE will degenerate into KNN.

(3) An Example of the SemiE. Figure 1 illustrates the semieager learning algorithm for the case where the instances are points in a two-dimensional space and where the target function is Boolean valued. The positive and negative training examples are shown by "+" and "-", respectively. Both the class centers are marked in red and the noisy points are marked in blue. Individual noisy point has no significant effect on the position of the class center. In the figures, "☆" represents a query data point to be classified. In order to classify the data point, distance values to each class center, $(x_i - \mu_j)^2$, will be calculated, along with the regularization term r_j. Since the frequency values of the two classes are almost identical in Figure 1(a), classification has no relation with the regularization term r_j. Classification will be completely determined by the distance values to the class centers. Since the query data point is closer to the positive examples center in Figure 1(a), the algorithm classifies the data point as a positive example. In Figure 1(b), the negative examples are much more than the positive examples, and the query instance has the same distance to both the two class centers. Thus, the data point to be classified will be assigned to the negative examples according to the regularization term r_j.

Assume the training dataset is $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, with $x_i \in R^n$, $x_i \in Y$, and $Y = \{c_1, c_2, \ldots, c_k\}$ is the set of class labels. Assume $S = S_1 \cup S_2 \cup \cdots \cup S_k$, is a partition of set T, with $S_j = \{(x_i, y_i) \mid y_i = c_j\}$, $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, k$

(1) Training algorithm

(i) $\mu = \dfrac{\sum_{i=1}^{N} x_i}{N}$

(ii) $\sigma^2 = \dfrac{\sum_{i=1}^{N} (x_i - \mu)^2}{N}$

For each c_j within the set Y, and for each training example $\langle x_i, c_j \rangle$

(iii) $P(c_j) \longleftarrow \dfrac{\sum_{i=1}^{N} I(y_i = c_j)}{N}$

(iv) $\mu_j \longleftarrow \dfrac{\sum_i I(y_i = c_j) x_i}{\sum_i I(y_i = c_j)} = \dfrac{\sum_i I(y_i = c_j)(x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)})}{\sum_i I(y_i = c_j)}$

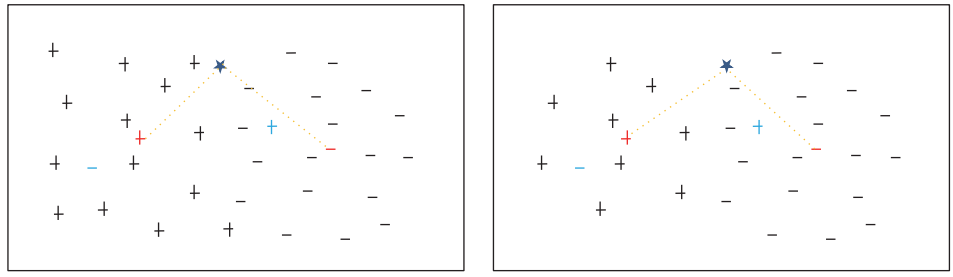
(v) $r_j \longleftarrow -\sigma^2 \ln P(c_j)$

(2) Classification algorithm

Given a instance x_i to be classified

(i) $C \longleftarrow \arg\min_j ((x_i - \mu_j)^2 + r_j)$

Algorithm 1: Algorithm of the semieager learning.



(a) The number of positive examples is identical to that of negative examples. And the query instance is near the positive center

(b) The positive examples are less than the negative examples. But the query instance has the same distance to positive and negative centers

Figure 1: The SemiE. A set of positive and negative training examples is shown, along with a query instance to be classified. The query instance is assigned to the positive examples on the left, whereas the algorithm classifies it as negative on the right.

3.2. Phrases Representation. To classify the relations, it is required to generate phrase-level feature vectors $v \in R^m$ with the same size for each relation phrase. Given a relation phrase with N words $R = w_1 \cdots w_j \cdots w_N$, let v_j be the corresponding word embedding of w_j; we adopt three schemes to calculate phrase representation. The adapted attention model is one of our contributions in this paper.

(1) Vector-Sum. Because of the various phrase lengths, a Vector-Sum operation is a rational solution. The relation phrases are represented as the normalization of sum of the vectors within the phrases.

$$V = \frac{\sum_{i=1}^{N} v_i}{\left\| \sum_{i=1}^{N} v_i \right\|} \tag{16}$$

(2) Max-Pooling. A pooling approach is often used in the CNN-based models [15]. With the RNN structure, this approach has been used for sentence-level semantic embedding [35]. The element $V^{(i)}$ of V is obtained by using Max-Pooling operation as follows:

$$V^{(i)} = \max_j \left\{ v_j^{(i)} \right\} \quad j = 1, \ldots, N, \; i = 1, \ldots, m \tag{17}$$

Here m is the dimension of the word vector v_j and $v_j^{(i)}$ is the i-th element of v_j.

(3) Attention Model. Recently, attention mechanism is introduced into deep learning and achieves success in a wide range of tasks, such as question answering, machine translations and speech recognition [36], and image captioning [37]. When reading an article, people pay more attention to the words or segments that are valuable for comprehending the text. Similarly, since various words contribute differently to the ultimate objective in NLP tasks, attentive neural networks calculate a series of weighted distribution for words sequence

in text. Inspired by this, we present a novel approach to calculate phrase representation based on the information quantity of the words within the phrase.

$$V = \sum_{i=1}^{N} \beta_i v_i \tag{18}$$

where

$$\beta_i = \frac{-\log_2 p_i}{-\sum_{j=1}^{N} \log_2 p_j} = \frac{\log_2 p_i}{\sum_{j=1}^{N} \log_2 p_j}, \tag{19}$$

with p_i being the probability of the word w_i. Thus, a word with a higher probability will get a lower β_i value.

4. Experimental Results and Analysis

To the best of the authors' knowledge, we are the first to tackle the problem of open relation classification by employing machine learning algorithm in this paper. The proposed semieager learning approach belongs to the class of supervised models. Consequently, one of the key challenges for this research is training data, since there is no large training set available for open relation classification.

Though a multitude of ORE systems have been developed in recent years, a well-defined, generally accepted definition for this task is still missing. In this situation, it is difficult to create a large-scale annotated corpus serving as a gold standard dataset for an objective and reproducible cross-system comparison. As a consequence, ORE systems were predominantly evaluated on small-scale corpora that were created by the researchers themselves. Although some of these corpora (e.g., the Wikipedia and Reverb datasets) are occasionally reused, none of the datasets for assessing the performance of different systems is widely agreed upon. Moreover, because ORE systems rely on unsupervised extraction strategies, these datasets generally consist only of unlabeled natural language sentences that cannot be used for training a supervised model, such as the SemiE.

Fortunately, in the research area of traditional relation classification, there are several benchmarks that consist of sentences which have been manually annotated with relation labels. As the first attempt to classify open relations, we would adopt some of these datasets (with minor modifications) to evaluate our models presented in this paper.

In subsections below, these four components are described in detail. Section 4.1 shows the datasets and evaluation metrics. The training process and the efficiency of our models are discussed in Section 4.2. Section 4.3 describes the experimental results, and the error analysis is presented in Section 4.4.

4.1. Datasets and Evaluation Metrics. There are two datasets which are shown to be closely related to open relation classification, and the annotated examples in these datasets can be easily converted into the form of open tuples. The first one is from the SemEval-2010 Task 8. The second dataset is a revision of MIML-RE annotation dataset, provided by Gabor Angeli et al.

In the SemEval-2010 Task 8, there are 9 relationships (with two directions) and an undirected 'Other' class, resulting in 19 relation classes in total. This dataset contains 10,717 annotated examples, including 8,000 sentences for training and 2,717 for testing. To make the dataset more suitable for our task, we changed the annotated instances into the form of (NP, VP, NP), as described in Section 2 and Box 1. For example, "The $[\text{owl}]_{e1}$ held the mouse in its $[\text{claw}]_{e2}$", was transformed into "$[\text{The owl}]_{e1}$ held the mouse in $[\text{its claw}]_{e2}$." The open relation phrase is "held the mouse in", and the relation class is Component-Whole(e2,e1). Since we aimed at classifying verb-mediated relations, the instances that cannot be converted into the form (NP, VP, NP) were not taken into consideration (Box 1). For example, the Component-Whole(e1,e2) instance "He decided to pad the $[\text{heel}]_{e1}$ of $[\text{shoes}]_{e2}$ with a shock absorbing insole or heel pad." was ignored.

The MIML-RE dataset consists of annotated sentences from the 2010 and 2013 KBP collections, along with a dump of Wikipedia in July 2013. There are a total of 33811 sentences that have been annotated. According to description of KBP task, there are 41 relations in total. We again took into consideration only the instances that can be converted into the form (NP, VP, NP), for example, the relation "per: city-of-death(e_1,e_2), $[\text{Smith}]_{e1}$ died in $[\text{Wilkes-Barre General Hospital}]_{e2}$".

SemEval-2010 dataset defined 18 generally accepted relation types and "Other". In this task, the class Other is used to indicate that the relation between two nominals does not belong to any of the 18 relation classes of interest. Therefore, SemEval-2010 dataset is "complete" for open relation classification. Although the MIML-RE dataset is of no completeness, it still defined up to 41 commonly accepted relation types. And this is practically "sufficient" for our task, to some extent. Thus, we adopted these datasets to reduce the labor costs of manual annotation.

There are mainly two aspects in which MIML-RE is different from SemEval-2010 Task 8: (1) MIML-RE is dominated with entity names (pairs of nouns) which are more sparse than SemEval-2010 Task 8. And there are more target nouns containing more than one word; (2) sentences in MIML-RE are averagely much longer than SemEval-2010 Task 8, as we can notice in Table 1.

The SemEval-2010 Task 8 benchmark is one of the commonly used datasets in traditional relation classification. The performance is evaluated in terms of the F1 score defined by this task. Both the data and the evaluation tool are publicly available (http://semeval2.fbk.eu/semeval2.php?location=data). For evaluation on this dataset, we applied the official scoring script and report the macro F1 score which also served as the official result of the shared task. In the experiments that are conducted on MIML-RE dataset, we adopt three metrics, including precision, recall, and F1 score, to evaluate our models. This dataset is also publicly available (http://nlp.stanford.edu/software/mimlre.shtml) but does not provide official evaluation tools.

4.2. Model Training. In order to classify the open relations with SemiE, it is required to generate phrase-level feature vectors. Three schemes are adopted to calculate representation

Table 1: The distribution of context lengths with two datasets.

Dataset	Context Length			Proportion of Long Context (≥11)
	≤10	11-15	≥16	
SemEval-2010 task-8	6658	3725	334	0.379
MIML-RE	6618	11647	15546	0.804

Table 2: Performance comparison on SemEval-2010 Task 8.

Model	Additional Information	F1	Epochs (Training)
BRNN	Word embeddings	82.8	20
CR-CNN	+PF[1]	84.1	10
SDP-LSTM	+POS+GR+WordNet embeddings	83.7	110
BLSTM	+PF+POS+NER	84.3	50
Att-BLSTM	Word embeddings + PI[2]	84.0	100
EAtt-BiGRU	Word embeddings + PF	84.7	10
SemiE	+ vector_sum	84.1	1
	+ max_polling	84.6	1
	+attention_mode	85.1	1

1: position feature.
2: position indicator, to be inserted into input sentence to specify the boundaries of target entity pair.

Table 3: Performance comparison on MIML-RE.

Model	Additional Information	Precision	Recall	F1	Epochs (Training)
BRNN	Word embeddings+PI[1]	67.4	52.1	58.8	20
CNN	Word embeddings+PI	61.8	49.7	55.1	15
SemiE	+ vector_sum	63.5	53.0	57.8	1
	+ max_polling	62.1	55.4	58.6	1
	+ attention_mode	70.2	52.5	60.1	1

1: position feature.

for the relation phrases within the annotated examples, as described in Section 3.2. In these schemes, we use the released word embedding set GoogleNews-vectors-negative300.bin to produce phrase representations, which is trained by Mikolov's word2vec tool (http://code.google.com/p/word2vec/). For the words that are not contained in pretrained word embedding set, the vectors of them are initialized randomly ranging from -0.25 to 0.25.

As detailed in Algorithm 1, the proposed SemiE learns only two parameters for each class during its training process. In our experiments, all the parameters are learned by making only a single pass over the training corpus. Nevertheless, the ANN-based models have complex structure and massive parameter set. In order to optimize these parameters, it takes about tens or dozens (if not hundreds) of epochs to train the models, with the potential risk of overfitting (see Tables 2 and 3 for details). Accordingly, the SemiE algorithm presented in this paper is significantly more efficient than all of the compared methods.

4.3. Experimental Results. Table 2 illustrates the comparison of our proposed method with some other state-of-the-art relation classification models on SemEval-2010 Task 8.

SDP-LSTM [14] picks up heterogeneous information via the shortest dependency path within entity pair and integrates external linguistic features via multichannel LSTM networks. Based on this, it obtains an F1-score of 83.7%. To get better performance, many relation classification models involve the external lexical knowledge. However, although the proposed SemiE model does not make use of any complicated human-designed features, it still achieves the superior F1-score of 85.1%, when it works with the attention model.

CR-CNN [16] presents a new ranking function to substitute Softmax and pays more attention to the influence of class "Other". This targeted modification achieves F1-score of 84.1%. By contrast, our model does not use any ranking function to finish classification.

All of the three models, BRNN, BLSTM, and Att-BLSTM, are based on the bidirectional RNN architecture. BRNN [25] utilizes the original RNN to extract sentence-level feature with the assistance of Position Indicator and Max-Pooling operation, which achieves F1-score of 82.8%. BLSTM [20] leverages bidirectional LSTM and a piece-wise Max-Pooling to generate sentence-level representation. This model achieves the performance of 84.3% with NLP tools and lexical resources. However, our result is yielded without any extra

features. Att-BLSTM [26] employs attention mechanism for relation classification. EAtt-BiGRU [27] utilizes attention layer with reasonable a priori knowledge. This model generates attention weights depending on corresponding entity pair information and brings better performance than Att-BLSTM. However, our attention mechanism involves only information quantity of the words within the phrase. The proposed approach outperforms these models without any extra features.

The next experiment compares with some recent models proposed by Zhang and Wang [25] with MIML-RE dataset. The experimental results are presented in Table 3.

From Table 3, we could find that when it works with Vector-Sum or Max-Pooling, the proposed SemiE gets similar F1 scores with the BRNN and CNN models but obtains a good balance of precision and recall. However, comparing to BRNN and CNN, SemiE+attention_model achieves a noticeable boost in precision, which leads to the best results in terms of F1 score on the MIML-RE dataset.

A particular advantage of the RNN model is that it can tackle long-distance patterns more effectively, compared to the CNN model. Nevertheless, Table 3 shows that the proposed SemiE+attention_mode significantly outperforms the RNN model. This is due to the large proportion of long contexts in the MIML-RE data. Thus, we could draw the conclusion that the SemiE+attention model is more suitable for long context relations than the RNN model.

4.4. Error Analysis. To better understand the merits and demerits of our models, we performed a detailed analysis of the classification errors occurring in the experiments.

The first type of errors is caused by the SemiE itself. As discussed in Section 3, SemiE takes an assumption that training examples obey a normal distribution. Nevertheless, if the number of samples is not large enough (e.g., less than 30), this assumption does not hold. By analysing the statistics of the annotated relation types of the SemEval-2010 Task 8 dataset, we found the relation type with the largest number of instances is Cause–Effect (1003 instances) and the smallest one is Instrument–Agency (504 instances). Because the number of training examples in each relation type is large enough, it is reasonable to take the assumption of normal distribution. As a consequence, the proposed SemiE achieves better performance on this dataset. In contrast, the training instances are not balanced across relation types in the MIML-RE dataset. For instance, the relation type *employee-of* contains 1978 annotated training examples, but there are only 14 annotated examples in the relation type *schools-attended*. We find that 65% of the classification errors occurred in relations with fewer annotated examples, which leads to a dramatic drop of F1 score on this dataset.

The second type of errors is caused by the relation phrase embeddings. The three most basic characteristics of a sequence are arguably its length, the items within it, and their order [38]. Somewhat surprisingly, the simple vector_sum model can encode a fair amount of information with regard to length, word content, and word order, although this model does not attempt to preserve word order information. The attention_model proposed in this paper is based on the vector_sum but more reasonable than it. Nevertheless, we note that when encoding longer phrases the proposed attention_model (and vector_sum) tends to lose more order information, which may cause additional classification errors.

5. Conclusion

Open relation extraction has been a growing field of research in the last few years. But there is no research presented in the literature for open relation classification to our knowledge. In this paper, we proposed a semieager learning approach, named SemiE, to deal with the problem in open relation classification. Unlike the eager learning approach to construct complex models, the proposed SemiE is much simpler and quite efficient but still preserves the benefits of both the eager learning and lazy learning approaches. To classify the relations with SemiE, it is necessary to generate phrase-level feature vectors with the same size. Although Vector-Sum and Max-Pooling can be employed, we still present an adapted attention model, inspired from the attention mechanism in deep neural network architecture.

Experimental results on two benchmark datasets demonstrated that the semieager learning approach can achieve better results than the newly presented approaches. And especially with the attention model the SemiE model exhibits clear advantages for sentences with long-distance relations.

Data Availability

The datasets supporting the results in this article are publicly available at http://semeval2.fbk.eu/semeval2.php?location=data and http://nlp.stanford.edu/software/mimlre.shtml.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by a research grant from the National Natural Science Foundation of China [No. 2016ZDA055].

References

[1] J. Kim and D. Moldovan, "Acquisition of semantic patterns for information extraction from corpora," in *Proceedings of the 9th IEEE Conference on Artificial Intelligence for Applications*, pp. 171–176, Orlando, FL, USA.

[2] E. Riloff, "Automatically generating extraction patterns from untagged text," in *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 1044–1049, August 1996.

[3] S. Soderland, "Learning information extraction rules for semi-structured and free text," *Machine Learning*, vol. 34, no. 1, pp. 233–272, 1999.

[4] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007*, pp. 2670–2676, India, January 2007.

[5] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for Open Information Extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2011*, pp. 1535–1545, UK, July 2011.

[6] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam, "Open information extraction: The second generation," in *Proceedings of 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011*, pp. 3–10, Spain, July 2011.

[7] Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni, "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, EMNLP-CoNLL 2012*, pp. 523–534, Republic of Korea, July 2012.

[8] Harinder Pal and Mausam, "Demonyms and Compound Relational Nouns in Nominal Open IE," in *Proceedings of the Workshop on Automated Knowledge Base Construction (AKBC) at NAACL*, 2016.

[9] Mausam, "Open information extraction systems and downstream applications," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016*, pp. 4074–4077, USA, July 2016.

[10] A. Zouaq, M. Gagnon, and L. Jean-Louis, "An assessment of open relation extraction systems for the semantic web," *Information Systems*, vol. 71, pp. 228–239, 2017.

[11] S. Soderland, J. Gilmer, and R. Bart, "Open information extraction to KBP relations in 3 hours," in *Proceedings of the Sixth Text Analysis Conference*, 2013.

[12] P. Qin, W. Xu, and J. Guo, "An empirical convolutional neural network approach for semantic relation classification," *Neurocomputing*, vol. 190, pp. 1–9, 2016.

[13] Y. Xu, R. Jia, L. Mou et al., "Improved relation classification by deep recurrent neural networks with data augmentation," 2016, https://arxiv.org/abs/1601.03651.

[14] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pp. 1785–1794, Portugal, September 2015.

[15] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of the 25th International Conference on Computational Linguistics, COLING 2014*, pp. 2335–2344, Ireland, August 2014.

[16] C. dos Santos, B. Xiang, and B. Zhou, "Classifying Relations by Ranking with Convolutional Neural Networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 626–634.

[17] K. Xu, Y. Feng, S. Huang, and D. Zhao, "Semantic relation classification via convolutional neural networks with simple negative sampling," 2015, https://arxiv.org/abs/1506.07650.

[18] Y. Liu, F. Wei, S. Li et al., "A dependency-based neural network for relation classification," 2015, https://arxiv.org/abs/1507.04646.

[19] T. H. Nguyen and R. Grishman, "Combining neural networks and log-linear models to improve relation extraction," 2015, https://arxiv.org/abs/1511.05926.

[20] S. Zhang, Z. Dequan, H. Xinchen, and M. Yang, "Bidirectional long short-term memory networks for relation classification," in *Proceedings of the International Conference on Neural Information processing*, pp. 216–227, 2017.

[21] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, "Reasoning with neural tensor networks for knowledge base completion," *Advances in Neural Information Processing Systems*, pp. 926–934, 2013.

[22] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in Neural Information Processing Systems*, pp. 2787–2795, 2013.

[23] A. Garcia-Duran, A. Bordes, N. Usunier, and Y. Grandvalet, "Combining two and three-way embedding models for link prediction in knowledge bases," *Journal of Artificial Intelligence Research*, vol. 55, pp. 715–742, 2016.

[24] S. Barzegar, A. Freitas, S. Handschuh, and B. Davis, "Composite Semantic Relation Classification," in *Proceedings of the International Conference on Applications of Natural Language to Information Systems*, pp. 406–417, 2017.

[25] D. Zhang and D. Wang, "Relation classification via recurrent neural network," 2015, https://arxiv.org/abs/1508.01006.

[26] P. Zhou, W. Shi, J. Tian et al., "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, pp. 207–212, Germany, August 2016.

[27] P. Qin, W. Xu, and J. Guo, "Designing an adaptive attention mechanism for relation classification," in *Proceedings of the 2017 International Joint Conference on Neural Networks, IJCNN 2017*, pp. 4356–4362, USA, May 2017.

[28] R. Collobert and J. Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 160–167, ACM, July 2008.

[29] P. D. Turney, "Distributional semantics beyond words: Supervised learning of analogy and paraphrase," *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 353–366, 2013.

[30] H. Schwenk, "Continuous space language models," *Computer Speech and Language*, vol. 21, no. 3, pp. 492–518, 2007.

[31] T. Mikolov, *Statistical Language Models Based on Neural Networks [Ph.D. thesis]*, Brno University of Technology, 2012.

[32] Tomas. Mikolov and Kai. Chen, "Efficient estimation of word representations in vector space," in *Proceedings of the Workshop at Int. Conf. on Learning Representations*, pp. 65–76, 2013.

[33] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic regularities in continuous spaceword representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2013*, pp. 746–751, USA, June 2013.

[34] A. Zakai and Y. Ritov, "Consistency and localizability," *Journal of Machine Learning Research (JMLR)*, vol. 10, pp. 827–856, 2009.

[35] H. Palangi, L. Deng, Y. Shen et al., "Deep Sentence embedding using long short-term memory networks: Analysis and application to information retrieval," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 24, no. 4, pp. 694–707, 2016.

[36] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Advances in Neural Information Processing Systems*, pp. 577–585, 2015.

[37] K. Xu, J. Ba, R. Kiros et al., "Show, attend and tell: Neural image caption generation with visual attention," 2015, vol. 2, no. 3 https://arxiv.org/abs/1502.03044.

[38] Y. Adi, E. Kermany, Y. Belinkov et al., "Fine-grained analysis of sentence embeddings using auxiliary prediction tasks," in *Proceedings of the ICLR Conference Track*, 2017.