Learning Dual Retrieval Module for Semi-supervised Relation Extraction

Hongtao Lin
University of Southern California
Los Angeles, California, USA
lin498@usc.edu

Jun Yan
Tsinghua University
Beijing, China
j-yan15@mails.tsinghua.edu.cn

Meng Qu
University of Illinois at Urbana-Champaign
Champaign, Illinois, USA
mengqu2@illinois.edu

Xiang Ren
University of Southern California
Los Angeles, California, USA
xiangren@usc.edu

ABSTRACT

Relation extraction is an important task in structuring content of text data, and becomes especially challenging when learning with weak supervision—where only a limited number of labeled sentences are given and a large number of unlabeled sentences are available. Most existing work exploits unlabeled data based on the ideas of self-training (i.e., bootstrapping a model) and self-ensembling (e.g., ensembling multiple model variants). However, these methods either suffer from the issue of semantic drift, or do not fully capture the problem characteristics of relation extraction. In this paper, we leverage a key insight that *retrieving sentences expressing a relation* is a dual task of *predicting the relation label for a given sentence*—two tasks are complementary to each other and can be optimized jointly for mutual enhancement. To model this intuition, we propose DualRE, a principled framework that introduces a retrieval module which is jointly trained with the original relation prediction module. In this way, high-quality samples selected by the retrieval module from unlabeled data can be used to improve the prediction module, and vice versa. Experimental results[1] on two public datasets as well as case studies demonstrate the effectiveness of the DualRE approach.

KEYWORDS

relation extraction, semi-supervised learning, dual learning

1 INTRODUCTION

Relation extraction (RE) plays a key role in turning massive, unstructured text corpora into structures of factual knowledge, and

---

[1]Code and data can be found at https://github.com/ink-usc/DualRE.

Sentence	Relation
Steve Jobs and Wozniak co-founded **Apple** in 1976.	org:founded_by
He was on Safari in South Africa with his mother, **Trudy**.	per:parents
Trump believes the **United States** has incredible potential.	no_relation

Figure 1: Sentence-level Relation Extraction Task. This task concerns about predicting the relation of two entities mentioned in a sentence. Tokens in orange and blue are subject and object entities in correspondence.

has a wide range of downstream applications in many domains [19, 28, 34]. Given two entities in a sentence, the goal of (sentence-level) relation extraction is to predict the relation expressed between two entities based on their sentence context. For example, in Fig. 1, we aim to classify each sentence into several relation labels, such as "org:founded_by", "per:parents", and "no_relation". While many works have been done on developing supervised relation extraction models [4, 24, 36, 38, 39], they typically require a large amount of human-annotated data (i.e., sentences that are manually assigned with the appropriate relation labels) for model training. Despite its wide applications, the process of manually annotating a large training set with many relation labels is too expensive and error-prone. The rapid emergence of large, domain-specific text corpora (e.g., social media content, customer reviews, scientific publications) calls for methods that can automatically extract entity relationships from text with *less human supervision.*

In recent years there has been a surge of interest in applying *distant supervision* to relation extraction [19, 22, 37, 38]. The general idea is to leverage facts stored in external knowledge bases (KBs) to annotate sentences in an unlabeled corpus as training data—two entities co-occurring in a sentence will be labeled with their KB relations regardless of the specific context. Although distant supervision automates the process of labeling sentences, it suffers from two major limitations. First, not all sentences where two entities co-occur are expressing their KB relations (i.e., context-agnostic label noise). Second, in some application domains the target relations may not be found in existing knowledge bases (e.g., WikiData), making it hard to apply distant supervision for label generation. It is thus desirable to develop methods that can rely on less human-annotated data for training relation extraction models. This motivates the study of *semi-supervised relation extraction* methods [1, 3, 31], which seek to leverage a limited number of labeled
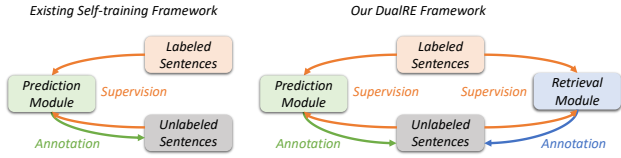
Figure 2: Illustration of the differences between the proposed DualRE framework and existing Self-training framework. Left: Self-training framework uses a single prediction module to annotate unlabeled sentences for generating additional labeled data. Right: DualRE framework leverages a prediction module and a retrieval modules for generating labeled data, where the newly labeled sentences are used to improve both modules. Two modules are jointly optimized and mutually enhance each other.

sentences and a large unlabeled (background) corpus for model training. Such setting has broad use cases, including cold-start knowledge base construction[2] and question answering [11].

In this paper, we study the problem of semi-supervised relation extraction. Towards this goal, there are broadly two kinds of methods, i.e., *self-ensembling* methods, and *self-training* methods. The main idea behind self-ensembling methods [12, 23, 27, 33] is to impose the assumption of "consistency under perturbation"—the prediction on the unlabeled instances should remain unchanged under small perturbations either on the data, or on the model parameters. In contrast, self-training methods [25, 29] aim to incrementally promote high-confidence predictions over unlabeled instances into the labeled training set, and improve the model by re-training with the updated labeled set. However, these methods suffer from the limitations as follows. (1) *Insufficient Supervision*: The process of enforcing prediction consistency between model variants in self-ensembling methods does not involve promoting new labeled instances from the unlabeled set. This makes it hard to improve models when "consistency-based" supervision is relatively weak compared to that from actual instances. (2) *Semantic Drift* [9]: While the self-training methods annotates unlabeled sentences as extra training data, the incremental promotion process can be biased (i.e., provides incorrect predictions), and may accumulate errors (i.e., including falsely-labeled instances into the training set). To overcome aforementioned challenges, we aim to design an approach that can generate high-quality labeled instances (as additional supervision) from the unlabeled corpus in an effective manner.

Intuitively, *retrieving sentences for a given relation* can be seen as a dual problem of *predicting the relation expressed in a sentence* (see the right-hand side of Fig. 2 for illustration). Models for the retrieval problem can help obtain more relevant sentences to a given relation from an auxiliary unlabeled corpus, for improving models for the prediction problem; while models for the prediction problem can predict the relation labels of the unlabeled sentences, and therefore help determine the quality of relevant sentences to collect (under each relation) for benefiting retrieval models. Such an observation of "problem duality" motivates us to solve the primal and dual problems jointly, and leverage the duality to overcome the aforementioned challenges. Specifically, as the prediction and retrieval models annotate/retrieve unlabeled sentences from the

auxiliary corpus, both models provide additional training data to each other and thus help resolve the insufficient supervision issue. Meanwhile, during the joint learning process, both models provide feedback to correct the output of each other, helping generate high-quality labeled data and resolve semantic drift issue.

To model the above ideas, we propose a novel framework, called **DualRE**, for semi-supervised relation extraction. DualRE consists of a prediction module and a retrieval module (see Fig. 3 for illustration of the model design). Given a sentence x, the prediction module aims to determine its relation label y with a neural relation extraction model by approximating $p(y|x)$ (e.g., position-aware LSTM [39]). Meanwhile, the retrieval module employs a neural learning-to-rank model to approximate $p(x|y)$ (e.g., using pointwise or pairwise loss) to facilitate ranking sentences $\{x\}$ to identify relevant ones for a query relation y. During training, the two modules are jointly optimized to provide feedback for each other, by alternating between two processes as follows. In the first process, we fix the prediction module, and use additional training data promoted by both modules to update the retrieval module. Similarly, in the second process, the retrieval module is fixed, and we generate extra training data using both modules and update the prediction module. By doing so, the retrieval module can consistently provide additional supervision to improve the prediction module. As the two modules are jointly leveraged to promote new training data (via referencing the label distribution), it enables the prediction and retrieval module to mutually correct each other and generate high-quality annotation.

We conduct extensive experiments on two public relation extraction datasets to evaluate DualRE[3]. Experimental results show that DualRE consistently outperforms several state-of-the-art methods under different data settings (by varying the size of labeled data and unlabeled data). Detailed case study demonstrates that DualRE is able to select quality instances to guide the model training.

In summary, in this paper we make the following contributions:

- We propose a novel framework for semi-supervised relation extraction, which encourages the (primal) prediction task and the (dual) retrieval task to mutually benefit each other by generating additional training instances from an unlabeled corpus.
- We develop a joint learning algorithm to alternatively optimize the prediction and retrieval modules, such that they can collaborate with each other and improve the quality of annotations.
- We conduct extensive experiments on two public datasets. Experimental results and case studies validate the effectiveness of the proposed framework.

2 BACKGROUND AND PROBLEM

In this section, we introduce the relevant concepts and notations, and formally define semi-supervised relation extraction task.

Entity Mention. An entity mention e is a token span (or word sequence) that refers to a real-world entity. In Fig. 1, the word "Apple" is an entity mention that maps to the company *Apple Inc.*.

---

[2]https://tac.nist.gov/2017/KBP/ColdStart/index.html

[3]Source code: https://github.com/INK-USC/DualRE

Relation Mention and Label. A relation mention consists of two entity mentions and the sentence x they co-occur in (i.e., the two entity mentions are sub-strings of the sentence). Here we consider entity relations as directional connections between two entities. Therefore, we denote the left entity as subject entity $e_s = (x_{s_1}, ..., x_{s_p})$ and the right entity as object entity $e_o = (x_{o_1}, ..., x_{o_q})$, where s_i and o_j are the token indexes of subject and object entities in the sentence, respectively. A relation mention (x, e_s, e_o) can be assigned with a relation label $r \in \mathcal{R}$ (i.e., e_s and e_o express relation r in the sentence x). We use $\mathcal{R}$ to denote the pre-defined set of relations of interests in the downstream application. In particular, undefined or irrelevant relations are assigned with a special label "no_relation".

Problem Definition. In this paper, our goal is to learn a model to effectively predict the relation label between two entities e_s and e_o for a relation mention (x, e_s, e_o). Fig. 1 presents some examples for the task. Each row demonstrates a relation mention with two entities highlighted in blue (subject entity) and red (object entity). For relation mentions that are not expressing target relations of interests (or do not express any relation) like the third example, we assign "no_relation" as a special label in the relation label set. Moreover, the third example also demonstrates the difference between *sentence-level* and *instance-level* relation extraction. Despite the well-known fact that *Donald Trump* is the president of *United States*, which can be extracted from multiple sentences in the corpus, we cannot infer such relation from this specific sentence and thus should return a label that is the most appropriate for the given sentence. Formally, we define the task of sentence-level relation extraction as follows.

Definition 2.1. **(Sentence-level Relation Extraction)** Given a sentence $x = (x_1, x_2, ..., x_n)$, a subject entity e_s and an object entity e_o in the sentence, the task of sentence-level relation extraction is to predict the relation label $r \in \mathcal{R}$ given the sentence and two entities, where $\mathcal{R}$ is a pre-defined set of relations including "no_relation".

Under the semi-supervised learning setting, we aim to learn the relation extraction model using both *labeled relation mentions* (which are manually annotated) and *unlabeled relation mentions* (of which entity mentions are identified by some existing named entity recognition tools). Formally, let $\mathbf{x} = (x, e_s, e_o)$ denote a relation mention and y denote its corresponding relation label r from the set $\mathcal{R}$. We define the task of semi-supervised relation extraction as follows.

Definition 2.2. **(Semi-supervised Relation Extraction)** Given a set of labeled relation mentions $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_L}$ and a set of unlabeled relation mentions $U = \{\mathbf{x}_j\}_{j=1}^{N_U}$, semi-supervised relation extraction task aims to learn a multi-class classification model f that (1) fits the labeled training data L; and (2) captures the information in the unlabeled data U, at the same time.

In the following sections, we stick with the denotation of $\mathbf{x}$ and y when referring to the relation mention (x, e_o, e_s) and relation r.

Our Focus. In this work, we assume entity mentions have been recognized from text using existing tools. In our experiments, the entity mentions are provided by the public datasets [15, 39]. Similarly, the set of relations $\mathcal{R}$ is also given by the datasets. In particular, we focus on the cases that the labeled instances are manually annotated with reliable labels, as compared to the noisy labels obtained
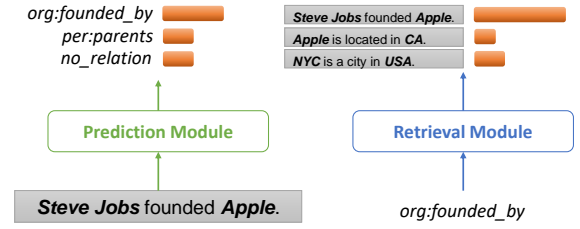

Figure 3: Illustration of the Two Modules in DualRE. Left: The prediction module focuses on the primal task, that is, predicting the relation label y of a sentence x. Right: The retrieval module focuses on the dual task, which aims at retrieving relevant sentences x for a query relation y.

via distant supervision [22]. Here we do not consider applying distant supervision to generate pseudo labels over the unlabeled relation mentions, and leave this as future work.

3 THE DUALRE FRAMEWORK

Framework Overview. This section introduces our dual learning approach to semi-supervised relation extraction. Given that *retrieving sentences expressing a relation* can be seen as a dual task of *predicting the relation expressed in a sentence*, we propose DualRE to take advantage of the dual property of two tasks and overcome the challenges mentioned in the introduction. At a high level, DualRE trains a module for the prediction task and a module for the retrieval task jointly. The retrieval module helps obtain relevant sentences for each relation from an auxiliary unlabeled corpus, which are then used as additional training data for the prediction module. Meanwhile, the prediction module predicts the relation labels for the unlabeled sentences, and determines the relation labels for retrieved sentences, which then serve as training data to improve the retrieval module. Two modules are jointly optimized to mutually enhance each other, generating additional, high-quality labeled data to address the problems of insufficient supervision and semantic drift.

Formally, our framework consists of a prediction module $\mathcal{P}_\theta$ and a retrieval module Q_ϕ, where θ and ϕ are their model parameters. The prediction module focuses on the primal task, i.e., predicting the relation label for a given relation mention. Therefore, for a mention-label pair $(\mathbf{x}, y)$, it models the conditional probability $p_\theta(y|\mathbf{x})$. By contrast, the retrieval module solves the dual task, i.e., retrieving relevant sentences expressing a specific relation, and hence it defines the probability $q_\phi(\mathbf{x}|y)$ for a mention-label pair $(\mathbf{x}, y)$. In practice, since $q_\phi(\mathbf{x}|y) \propto q_\phi(\mathbf{x}, y)$ when given a specific relation y, the retrieval module will focus on estimating $q_\phi(\mathbf{x}, y)$ instead. The overall objective function is given below:

$$\mathbf{O} = \mathbf{O}_P + \mathbf{O}_R + \mathbf{O}_U \tag{1}$$

There are three terms in total. In $\mathbf{O}_P$ and $\mathbf{O}_R$, we use the given labeled sentences to train the prediction module and the retrieval module, respectively. In the unsupervised part $\mathbf{O}_U$, we promote the collaboration of the two modules to annotate unlabeled sentences as additional training data. In the rest of this section, we will introduce the details of each key component in DualRE.

$p_\theta(y|\mathbf{x})$

org:founded_by   0.95
per:parents   0.03
no_relation   0.01

Softmax Classifier

$\mathbf{z}$

RNN / CNN Encoder

| $\mathbf{x}_1$ | $\mathbf{x}_2$ | $\mathbf{x}_3$ | $\mathbf{x}_4$ |

Steve Jobs founded ***Apple***.

$q_\phi(\mathbf{x}|y)$

0.9   ***Steve Jobs*** founded ***Apple***.
0.3   ***NYC*** is a city in ***USA***.
0.2   ***Apple*** is located in ***CA***.

Learning-to-rank Layer

$\mathbf{z}$ $\mathbf{y}$

RNN / CNN Encoder   Lookup Table

| $\mathbf{x}_1$ | $\mathbf{x}_2$ | $\mathbf{x}_3$ | $\mathbf{x}_4$ |

org:founded_by

Steve Jobs founded ***Apple***.
Apple is located in ***CA***.
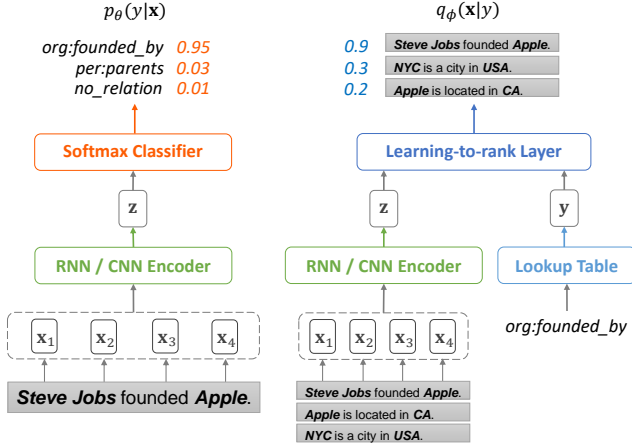***NYC*** is a city in ***USA***.

Figure 4: Model Design of the Two Modules in DualRE. Left: The prediction module encodes the relation mention and outputs a probability distribution over all relations. Right: The retrieval module ranks the relation mentions given a query by predicting a relevance score $q_\phi(\mathbf{x}, y)$. Retrieving instances given a relation is essentially taking the top instances from the ranking list. Note that this figure does not illustrate how two modules mutually enhance each other using unlabeled data.

3.1 Relation Prediction Module

In the DualRE framework, the prediction module $\mathcal{P}_\theta$ focuses on the primal task, that is, predicting the relation expressed in a relation mention, as shown in Fig. 4. We build the prediction module by first leveraging an encoder to encode the relation mention, and then adopting a softmax classifier to perform relation classification.

Relation Mention Encoder. The goal of relation mention encoders is to encode relation mentions into latent vectors, which can serve as features for prediction. A variety of encoders have been proposed for relation mention encoding, such as Long Short-Term Memory (LSTM) encoders [17], Piecewise Convolutional Neural Network (PCNN) encoders [37], and Position-aware Recurrent Neural Network (PRNN) encoders [39]. Among them, PRNN outperforms all other encoders on relation extraction, and therefore we use PRNN as our default encoder, but our framework can also adopt other encoders.

Formally, given a relation mention, PRNN considers the POS tagging, NER tagging, the position of each token (with regard to object and subject entities) as features. Then an LSTM cell is leveraged to learn the hidden representation of each word. Finally, the attention mechanism is utilized to weighted aggregate different hidden representations as the final encoding of the relation mention.

Objective Function. After encoding a relation mention, we feed it into a softmax classifier to predict the relation label of the mention. The prediction module can be trained using the given labeled data, in which we leverage the cross-entropy objective function, which is given below:

$$\mathbf{O}_P = \mathbb{E}_{(\mathbf{x}, y) \in L}[\log p_\theta(y|\mathbf{x})] \tag{2}$$

Given a relation mention $\mathbf{x}$ and its relation label y, the prediction module seeks to maximize the probability of y for the mention $\mathbf{x}$.

3.2 Sentence Retrieval Module

The intended behavior of retrieval module $\mathcal{Q}_\phi$ is to select a set of relevant mentions given specific relations. In information retrieval domain, this is similar to the problem of retrieving a set of documents given a query. In our case, "document" and "query" correspond to relation mention and relation, respectively.

Learning-to-rank [20] is a type of model that aims to rank the documents according to its relevance to a query, thus we can select the top-ranking ones as retrieved documents. Learning-to-rank formulates such problems using joint probability $q_\phi(\mathbf{x}, y)$, which is in proportion to $q_\phi(\mathbf{x}|y)$ when the probability of observing query $q(y)$ is fixed. Similar to the prediction module, we can train the retrieval module by leveraging the labeled data and optimizing the following objective function:

$$\mathbf{O}_R = \mathbb{E}_{(\mathbf{x}, y) \in L}[\log q_\phi(\mathbf{x}, y)] \tag{3}$$

However, directly optimizing $q_\phi(\mathbf{x}, y)$ is intractable, since we have to ensure $\sum_{(\mathbf{x}, y)} q_\phi(\mathbf{x}, y) = 1$, which requires traversing over all possible pairs of $(\mathbf{x}, y)$ to calculate the partition function. Therefore, following existing learning-to-rank models, we seek not to enforce $q_\phi(\mathbf{x}, y)$ as a valid distribution. Instead, we let $q_\phi(\mathbf{x}, y)$ output a score to estimate the degree of *relevance* of the pair $(\mathbf{x}, y)$. In literature, there are mainly two types of approaches to train such retrieval models, i.e., the pointwise approach [8, 18] and the pairwise approach [5, 6], and we tried both in our experiment. Next, we introduce these approaches respectively.

Pointwise Approach. Pointwise learning-to-rank models assume that each query-document pair is assigned a score that reflects their degree of relevance, which can be used to determine the rank for each document given a query.

In the task of retrieving relation mentions, for each relation mention $\mathbf{x}$, we have a label y to indicate which relation it belongs to. Alternatively, we can regard such supervision signal as a binary score (0 or 1) that indicates whether a relation mention $\mathbf{x}$ is relevant to a specific relation $y \in \mathcal{R}$. In other words, each relation mention in the labeled dataset is paired up with all relations, with a binary label to indicate if they are relevant.

The model design for the learning-to-rank retrieval module is illustrated in Fig. 4. Given a relation y and a set of relation mentions $\{\mathbf{x}\}$ to be retrieved, we first encode each relation mention $\mathbf{x}$ as a vector $\mathbf{z}$. Each relation y is mapped to its relation embedding $\mathbf{y}$ according to a relation embedding matrix R, which has the same dimension as $\mathbf{z}$. In the learning-to-rank layer, pointwise approach takes the inner product $q_\phi(\mathbf{x}|y) = \sigma(\mathbf{z}^\top \mathbf{y})$ as the relevance score, where $\sigma(\cdot)$ denotes the sigmoid activation function.

$$\mathbb{E}_{(\mathbf{x}, y) \in L}[\log \sigma(\mathbf{z}^\top \mathbf{y})] + \mathbb{E}_{(\mathbf{x}, y') \notin L}[\log(1 - \sigma(\mathbf{z}^\top \mathbf{y}'))], \tag{4}$$

where $(\mathbf{x}, y)$ is a pair in the labeled dataset L, $(\mathbf{x}, y')$ pairs up the relation mention $\mathbf{x}$ with incorrect relation y', $\mathbf{z}$ is the mention encoding for $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{y}'$ are embeddings of relation y and y'.

Pairwise Approach. Pairwise learning-to-rank models assume that a relative order can be determined between the retrieved documents. For a set of documents, we can compare any pair of them

and predict their relative order. We expect the model predictions (i.e. the set of relative orders) are consistent (no conflicting results such as $a > b, b > c$ but $c < a$), and thus the absolute order of each document in the ranked list can be derived.

The core part of pairwise learning-to-rank models is the design of objective function that can characterize the relative order of paired documents. We adopt the approach in RankNet [5], which proposes an objective function to minimize the number of inversions in ranking. An inversion happens when we rank a lower rated document above a higher rated one in a ranked list.

In the task of retrieving relation mentions, given a relation, we aim to rank all relevant mentions higher than the irrelevant ones. Similar to the design of pointwise learning-to-rank, we also encode each relation mention $\mathbf{x}$ and relation y into vector representation $\mathbf{z}$ and $\mathbf{y}$ respectively. The only difference is how the objective function in the learning-to-rank layer (as shown in Fig. 4) is calculated. We approximate O_R in Eq. 3 using the following term:

$$\mathbb{E}_{(\mathbf{x},y),(\mathbf{x}',y')\in L\times L}[r(\mathbf{x},\mathbf{x}',y)\log\sigma(\mathbf{z}^T\mathbf{y}-\mathbf{z}'^T\mathbf{y})], \quad (5)$$

where the order score $r(\mathbf{x},\mathbf{x}',y)$ expresses the relative order of two relation mentions, given the relation y. Consider $\mathbf{x}$ as a positive instance for relation y, the order score will return 1 if $\mathbf{x}'$ is a negative instance. During optimization process, this indicates that we attribute $\mathbf{z}^T y$ a higher score than $\mathbf{z}'^T y$. If $\mathbf{x}'$ happens to be another positive instance, we cannot make assertion about their relative order, then the order score returns 1/2. During the optimization process, this indicates that we assume the score $\mathbf{z}^T y$ and $\mathbf{z}'^T y$ are close.

3.3 Interaction Between the Two Modules

So far, we only introduced the formulation of prediction and retrieval modules and how they can be trained in a supervised way (by maximizing O_P and O_R). In the semi-supervised setting, we try to incorporate unlabeled data, which leads to the introduction of O_U in overall objective function. In DualRE, the formulation of this unsupervised objective function involves both prediction and retrieval modules. Intuitively, we hope to leverage the unlabeled data under the collaboration between both modules.

Formally, in $\mathbf{O}_U$, we seek to maximize the log-likelihood function of the unlabeled relation mentions:

$$\mathbf{O}_U = \mathbb{E}_{\mathbf{x}\in U}[\log q_\phi(\mathbf{x})]$$
$$\geq \mathbb{E}_{\mathbf{x}\in U, y\sim p_\theta(y|\mathbf{x})}[\log\frac{q_\phi(\mathbf{x},y)}{p_\theta(y|\mathbf{x})}] \quad (6)$$

The above inequality is derived from the EM algorithm [10]. The two sides are equal when $p_\theta(y|\mathbf{x}) = q_\phi(y|\mathbf{x})$. By maximizing the lower bound of O_U, we incorporate both modules in the optimization process. As we will see in Sec. 4, we approach the optimization process by sampling from unlabeled data and accepting the instances that both modules agree on. These newly retrieved instances are regarded as pseudo-labeled data and enhance the training of both modules. Such process effectively leverages the selection quality of retrieved instances and boosts the overall model performance.

4 JOINT LEARNING OF PREDICTION AND RETRIEVAL MODULES

In Sec. 3, we introduce the formulation for three components (O_P, O_R and O_U) of the overall objective function. This section will present the details on how we approach the optimization of the overall objective function O, and the learning algorithm.

4.1 The EM-based Joint Optimization

To optimize the objective function O, we follow the EM algorithm to alternatively optimize the prediction and retrieval modules.

E Step. In this step, we update the prediction module $\mathcal{P}_\theta$ by fixing the retrieval module Q_ϕ. This corresponds to the minimization of KL divergence $KL(p_\theta(y|\mathbf{x})||q_\phi(y|\mathbf{x}))$. Directly minimizing this objective function is difficult, thus we follow the wake-sleep algorithm [16] to minimize the reversed KL divergence $KL(q_\phi(y|\mathbf{x})||p_\theta(y|\mathbf{x}))$, which yields the same optimal solution, i.e., $p_\theta(y|\mathbf{x}) = q_\phi(y|\mathbf{x})$. By integrating Eq. 2 and Eq. 6 as the overall objective function, and taking derivative with respect to θ, the gradient for θ is calculated as:

$$\nabla_\theta\mathbf{O} = \mathbb{E}_{(\mathbf{x},y)\in L}[\nabla_\theta\log p_\theta(y|\mathbf{x})]$$
$$+ \mathbb{E}_{\mathbf{x}\in U, y\sim q_\phi(y|\mathbf{x})}[\nabla_\theta\log p_\theta(y|\mathbf{x})], \quad (7)$$

where the gradient is calculated from both the labeled data and additional data annotated by the retrieval module Q_ϕ. However, only leveraging the retrieval module for annotation can be biased, so we may expect both modules to mutually correct each other. Towards this goal, we notice that $\mathbb{E}_{p_\theta(y|\mathbf{x})}[\nabla_\theta\log p_\theta(y|\mathbf{x})] = 0$ [4], and therefore the gradient can be modified as:

$$\nabla_\theta\mathbf{O} = \mathbb{E}_{(\mathbf{x},y)\in L}[\nabla_\theta\log p_\theta(y|\mathbf{x})]$$
$$+ \mathbb{E}_{\mathbf{x}\in U, y\sim(q_\phi(y|\mathbf{x})+p_\theta(y|\mathbf{x}))}[\nabla_\theta\log p_\theta(y|\mathbf{x})] \quad (8)$$

Although the gradient remains the same as in Eq. 7, the new gradient enables us to leverage both modules to collect additional labeled sentences, which usually has lower noise, and therefore leads to better performance.

M Step. In this step, we update the retrieval module Q_ϕ by fixing the prediction module $\mathcal{P}_\theta$. The gradient with respect to ϕ can be similarly calculated as:

$$\nabla_\phi\mathbf{O} = \mathbb{E}_{(\mathbf{x},y)\in L}[\nabla_\phi\log q_\phi(\mathbf{x},y)]$$
$$+ \mathbb{E}_{(\mathbf{x},y)\sim(p_\theta(\mathbf{x},y)+q_\phi(\mathbf{x},y))}[\nabla_\phi\log q_\phi(\mathbf{x},y)] \quad (9)$$

Again, directly calculating $\log q_\phi(\mathbf{x},y)$ is infeasible, and therefore in practice we use the pointwise term (Eq. 4) or the pairwise objective function (Eq. 5) to replace $\log q_\phi(\mathbf{x},y)$.

To summarize, the model alternates between optimizing the objective function with regard to ϕ and θ, which is referred to as E and M step. During each step, we incorporate both labeled and unlabeled data. Both E and M step require sampling annotated instances from unlabeled data. Noticing that samples from distribution $\mathbf{x} \in U, y \sim (q_\phi(y|\mathbf{x}) + p_\theta(y|\mathbf{x}))$ is equivalent to sampling from distribution $(\mathbf{x},y) \sim (p_\theta(\mathbf{x},y) + q_\phi(\mathbf{x},y))$, we can keep a single version of sampled instances and apply to both steps.

---

[4] $\mathbb{E}_{p_\theta(y|\mathbf{x})}[\nabla_\theta\log p_\theta(y|\mathbf{x})] = \mathbb{E}_{p_\theta(y|\mathbf{x})}[\frac{\nabla_\theta p_\theta(y|\mathbf{x})}{p_\theta(y|\mathbf{x})}] = \sum \nabla_\theta p_\theta(y|\mathbf{x}) = \nabla_\theta\sum p_\theta(y|\mathbf{x}) = \nabla_\theta 1 = 0$

Next, we introduce an empirical sampling method that effectively generates high-quality training relation mentions (as additional supervision) to improve the performance.

4.2 Instance Selection

Empirically, sampling from distribution $p_\theta(\mathbf{x}, y) + q_\phi(\mathbf{x}, y)$ can be approximated by taking the intersection set of predicted results from both modules.

In prediction module $\mathcal{P}_\theta$, we first select new instances from distribution $p_\theta(\mathbf{x}, y)$. If we assume a uniform distribution for relation mentions $p(\mathbf{x})$, we are essentially sampling from distribution $p_\theta(y|\mathbf{x}) \propto p_\theta(\mathbf{x}, y)$. Given the unlabeled instance $\mathbf{x}$ and its predicted relation y, we rank these instances according to the probability of predicting y (i.e., $p(y|\mathbf{x})$) and take top-k instances from the ranked list. This is similar to what we do in self-training models.

In retrieval module Q_ϕ, we sample k instances from distribution $q_\phi(\mathbf{x}, y) = q_\phi(\mathbf{x}|y)q(y)$. The first term $q_\phi(\mathbf{x}|y)$ aligns with the intended behavior for the retrieval module: given a relation y, the retrieval module traverses through each unlabeled instance $\mathbf{x}$ and gives out a relevance score $q_\phi(\mathbf{x}, y) \propto q_\phi(\mathbf{x}|y)$. Therefore we can construct a ranked list for each relation y, in which top-k_y instances are retrieved for this relation. We aggregate the instances from each relation to construct the retrieved set. The second term $q(y)$ can be regarded as prior knowledge that constrains the relation distribution for retrieved instances. Given that we retrieved k instances which follows the reference relation distribution $q(y)$, we can determine the number of instances retrieved for each relation $k_y = kq(y)$.

Both modules collect a set of training instances, and the intersection of them are regarded as the final set. Intuitively, this is similar to majority voting or ensembling that considers multiple models when doing prediction.

4.3 Instance Promotion

To further eliminate the influence of incorrect annotations in retrieved instances, we experiment with two strategies to incorporate these instances in the training phase.

Equal promotion. Equal promotion treats the retrieved instances and original training data equally when calculating the loss function. This is the typical strategy for self-training models. The objective functions derived in Eq. 8 and 9 reflect this strategy.

Weighted promotion. In this strategy, each instance is assigned a weight π reflecting the confidence of its annotation, which is intended to be low when it's wrongly labeled. The weights are incorporated in training phase when promoting different instances. Considering Eq. 7 when training the prediction module, the first term is intended for promoting instances in the labeled set, which is assigned a weight of 1 and the formula remains unchanged. The second term is modified as follows:

$$\mathbb{E}_{(\mathbf{x}, y) \sim (p_\theta(\mathbf{x}, y) + q_\phi(\mathbf{x}, y))}[\pi_\mathbf{x}^p \nabla_\phi \log q_\phi(\mathbf{x}, y)], \quad (10)$$

where $\pi_\mathbf{x}^p$ is the weight for pseudo-labeled instance $(\mathbf{x}, y)$. Similarly, we introduce the weight $\pi_\mathbf{x}^q$ in objective function O_R for the retrieval module. In our experiment, we parameterize the weights as $\pi_\mathbf{x}^p = p_\theta(y|\mathbf{x})^\alpha$ and $\pi_\mathbf{x}^q = q_\phi(\mathbf{x}, y)^\beta$, in which α and β are hyperparmeters to tune in our model.

Algorithm 1: DualRE Learning Algorithm.

Input: Labeled data $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_L}$, unlabeled data $U = \{\mathbf{x}_j\}_{j=1}^{N_U}$, the amount of data to retrieve each in each iteration k.

Initialize: $L_U \leftarrow \emptyset$.

$P_\theta, Q_\phi \leftarrow$ Pretrain prediction and retrieval module using L.

while $U \neq \emptyset$ **and** *not converge* **do**

   $L' \leftarrow$ Retrieve k annotated instances $(\mathbf{x}, y)$ from U (Sec. 4.2).

   Remove instances L' from U and add them to L_U.

   // Update prediction module:

   Optimize P_θ using data from both L and L_U (Eq. 8).

   // Update retrieval module:

   Optimize Q_ϕ using data from both L and L_U (Eq. 9).

end

Table 1: Statistics for SemEval and TACRED dataset. We present the number of relation mentions in the train/dev/test sets. In particular, % Neg. implies the percentage of "negative" relation mentions (i.e., relation mentions that are labeled as "no_relation").

Dataset	# Train	# Dev	# Test	# Relations	% Neg.
SemEval [15]	7,199	800	1,864	19	17.40
TACRED [39]	75,049	25,763	18,659	42	78.68

4.4 The Joint Learning Algorithm

We summarize the training algorithm for DualRE in Algo. 1. The algorithm is basically an approximation of EM algorithm as described in Sec. 4.1. The whole process works in an iterative manner. In each iteration, we retrieve a new set of annotated instances using both prediction and retrieval modules. According to Sec. 4.2, this is an empirical practice to sample from distribution $p_\theta(\mathbf{x}, y) + q_\phi(\mathbf{x}, y)$. The retrieved instances, along with previous training data can be used for updating the prediction and retrieval module, which correspond to the E step and M step in the EM algorithm. The iterative process stops when our model converges or the unlabeled data is exhausted.

5 EXPERIMENTS

This section first introduces datasets and experimental settings, and then presents performance comparison results with baseline methods as well as case studies of various model components in order to validate different design choices in DualRE.

5.1 Dataset

We choose two public relation extraction datasets with different characteristics to validate the effectiveness of our proposed method. (1) **SemEval** [15]. SemEval 2010 Task 8 provides a standard benchmark dataset which is widely adopted for evaluating relation extraction systems. It has around 10,000 relation mentions in total (where entity mentions are given), with 19 relations (including "no_relation"). (2) **TACRED** [39]. The TAC Relation Extraction Dataset is a large-scale crowd-sourced relation extraction dataset

Table 2: Performance comparison on SemEval [15] with various amounts of labeled data and 50% unlabeled data. We report the mean and standard deviation of the evaluation metrics by conducting 5 runs of training and testing using different random seeds. DualRE outperforms all the baseline methods.

Methods / % Labeled Data	5%			10%			30%		
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
LSTM [17]	25.31 ± 2.16	20.72 ± 3.91	22.71 ± 3.31	36.91 ± 6.10	29.85 ± 7.25	32.92 ± 6.71	64.83 ± 0.69	63.03 ± 0.67	63.91 ± 0.66
PCNN [37]	42.95 ± 4.69	40.79 ± 4.59	41.84 ± 4.63	53.78 ± 1.51	49.11 ± 2.22	51.32 ± 1.74	64.54 ± 0.58	62.98 ± 0.48	63.75 ± 0.33
PRNN [39]	56.16 ± 1.32	54.87 ± 1.49	55.49 ± 0.90	61.70 ± 1.16	63.61 ± 2.07	62.63 ± 1.42	69.66 ± 2.19	68.76 ± 2.60	69.14 ± 1.02
Mean-Teacher (PRNN) [33]	53.71 ± 4.43	49.54 ± 3.29	51.51 ± 3.58	62.43 ± 1.28	60.34 ± 0.62	61.36 ± 0.75	68.65 ± 0.64	69.84 ± 0.65	69.24 ± 0.56
Self-Training (PRNN) [29]	56.47 ± 1.11	56.14 ± 1.33	56.30 ± 0.96	64.27 ± 2.37	63.48 ± 2.02	63.79 ± 0.28	68.95 ± 0.68	72.63 ± 0.82	70.74 ± 0.58
RE-Ensemble (PRNN)	58.77 ± 0.58	58.50 ± 0.97	58.63 ± 0.62	65.10 ± 0.84	64.57 ± 0.54	64.83 ± 0.61	70.26 ± 0.92	73.20 ± 1.22	71.69 ± 0.47
DualRE-Pairwise (PRNN)	59.76 ± 0.47	63.36 ± 0.77	**61.51 ± 0.56**	64.39 ± 0.75	67.70 ± 0.80	66.00 ± 0.48	70.05 ± 0.53	74.83 ± 0.88	72.36 ± 0.60
DualRE-Pointwise (PRNN)	58.73 ± 1.50	62.23 ± 1.93	60.43 ± 1.67	64.50 ± 1.14	67.67 ± 1.66	**66.03 ± 1.00**	70.03 ± 0.74	74.87 ± 0.75	**72.36 ± 0.35**
RE-Gold (PRNN w. gold labels)[*]	72.57 ± 1.47	74.65 ± 1.98	73.56 ± 0.31	71.40 ± 1.42	76.72 ± 0.64	73.95 ± 0.50	72.98 ± 0.96	78.86 ± 0.76	75.80 ± 0.24

following the TAC KBP relation schema[5]. The corpora are collected from all the prior TAC KBP shared tasks. It has more than 100,000 relation mentions with relations categorized into 42 classes (including "no_relation").

Table 1 shows the overall statistics for these two datasets, which are significantly different in terms of dataset size, number of relations, and percentage of negative instances. As one can observe from the table, TACRED is far more complicated than SemEval, since it has more types of relations and more skewed distribution between positive and negative instances. We conduct experiments on both datasets so as to validate the effectiveness of our DualRE model learned with datasets of different characteristics.

5.2 Compared Methods

As the proposed DualRE framework is general and can be integrated with various supervised "base" model of relation extraction, we first compare several widely used supervised models, then adopt the best performing one as the base model and compare DualRE with other representative semi-supervised methods. All models mentioned below are re-implemented and integrated into a unified interface for efficient experiment conducting.

For base models proposed for supervised relation extraction, we select the following models and train them only on the labeled dataset (i.e., set L): (1) **LSTM** [17]: The Long Short-Term Memory (LSTM) network is a variant of basic Recurrent Neural Network (RNN). We can simply adopt LSTM without attention mechanism to encode the word sequence and classify it into predefined relation categories based on the sentence representation. (2) **PCNN** [37]: Different from the encoding part of LSTM RE model, Piecewise Convolutional Neural Network (PCNN) uses CNN with piecewise max pooling to encode the word sequence. Position embeddings are used to provide the model with the positions of subject and object entities in the sentence. (3) **PRNN** [39]: Integrating position-aware attention mechanism, the Position-aware Recurrent Neural Network (PRNN) is proposed specifically for supervised relation extraction and has achieved state-of-the-art performance on TACRED dataset. Details of PRNN model are described in Sec. 3.1.

As PRNN has achieved state-of-the-art performance in prior work, we construct our DualRE model based on PRNN, and compare with other semi-supervised classification methods which also adopt PRNN as the base model for fair evaluation. For semi-supervised

baselines, we select several representative methods from different categories of semi-supervised learning approaches (See Sec. 6 for details), shown as follows. (4) **Self-Training** [29]: Self-Training is a semi-supervised method that uses a single model's predictions on unlabeled data to iteratively re-train and improve the model itself. In each iteration, the model is first used to label the unlabeled instances, then the most confident pseudo-labeled instances are added to the labeled set, which are further used to re-train the model in the next iteration. The iterative training procedure stops when the unlabeled data is exhausted. (5) **Mean-Teacher** [33]: Mean-Teacher belongs to the category of self-ensembling methods. It is first proposed for semi-supervised image classification task. The main idea is to encourage different variants of the model to make consistent predictions over similar inputs. A perturbation-based loss, and the original training loss, are jointly optimized to utilize the information provided by unlabeled data. In our implementation, the dropout strategy [13, 30] is applied to generate the perturbed input.

In addition to the existing semi-supervised methods introduced above, we propose a variant of DualRE model as another baseline model, to further validate the effectiveness of adopting a dual retrieval model in contrast to ensembling two prediction models with identical configuration, i.e., both modules use PRNN as the same base model but with different model parameters. (6) **RE-Ensemble**: By replacing the retrieval module in the proposed DualRE framework with the same prediction module (with a new set of model parameters), we design the RE-Ensemble model that selects the promoted instances based on the agreement of two prediction modules in each iteration. The structure and size of these two models are exactly the same, but they are independently initialized at random to make sure they are not identical.

For our proposed DualRE method, we focus on studying the performance of two model variants based on different learning-to-rank objectives in the retrieval module. The models are trained on both the labeled data and the unlabeled data, as detailed below: (1) **DualRE-Pointwise**: A DualRE model that uses pointwise learning-to-rank loss [7] in the retrieval module. (2) **DualRE-Pairwise**: A DualRE model that uses pairwise learning-to-rank loss [5] in the retrieval module.

Finally, we present another model to demonstrate the upper bound of the performance for all the semi-supervised methods. This model is learned using both the labeled dataset and the "unlabeled"

dataset. **RE-Gold**: It trains the PRNN base model using both the labeled relation mentions and the gold labels of relation mentions from the unlabeled set. This represents the setting that labels over all unlabeled instances can be perfectly predicted, which gives an upper bound performance.

5.3 Experimental Settings

Data Preparation. To study the performance of semi-supervised relation extraction models with various labeled and unlabeled dataset sizes, we sample from the original training set and construct various sizes of labeled and unlabeled set. More specifically, for SemEval dataset, we sample 5%, 10% and 30% training data as labeled sets, the ratios amounts to 3%, 10% and 15% in TACRED. In both datasets, we sample 50% training data as unlabeled set, whose ground-truth labels are unavailable to models. We perform stratified sampling to ensure the label distribution remains unchanged. Duplication in labeled and unlabeled set is avoided as well. Note that since SemEval does not have official development set, we sample 10% of the training set as the development set, before the above data preparation process.

Data Processing. Since the SemEval dataset does not contain named entity (NE) tags for each relation mention, which is essential for our PRNN base model, we apply the Stanford CoreNLP toolkit [21] to extract the needed information.

Hyperparameter Tuning. For base models (LSTM, PCNN, PRNN), we follow the default hyperparameter settings used in previous works [37, 39]. For fair comparison, both DualRE and other baseline methods are built on top of the same base models. For Mean-Teacher, we perform grid search for weight of consistency loss (select from $\{0.1, 1, 10\}$) and number of labeled instances in a mini-batch with size 50 (select from $\{1, 3, 5\}$). For DualRE, we perform greedy search for retrieval module dropout (select from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$), score weight α, β (select from $\{0.5, 1, 2\}$) and label distribution (select from top-k to top-$7k$ distribution and true distribution). The best hyperparameters are chosen according to model performance on the development set.

For semi-supervised models that promote instances in an iterative manner (Self-Training, RE-Ensemble and DualRE), we fix the number of retrieved instances per iteration k for fair comparison. Since DualRE model takes the joint selection of prediction and retrieval modules, retrieving a fixed amount of instances k is not guaranteed. To solve this problem, we starts with a retrieval upper bound $k' = k$ for both modules to select instances. We iteratively increase the upper bound ($k' = 1.25k'$) until we can sample k unique instances from the joint set. In our experiments, k is set to 10% of the number of original unlabeled instances, which means that the unlabeled data is exhausted after 10 iterations.

Evaluation Metric. Following previous work in relation extraction [19, 39], we consider F_1 score as the main metric while precision and recall serve as auxiliary metrics. Note that the correct prediction for no_relation is ignored during the calculation.

5.4 Experiments and Performance Study

Table 2 and 3 show the model performance on both SemEval and TACRED dataset when 50% unlabeled data and various labeled data are available. We conduct several runs of training and testing using different random seeds and report the mean and standard deviation of the results.

Comparing DualRE with Different Methods and Its Variants. Comparing the performance of different models in Table 2 and 3, the first conclusion we draw is that PRNN outperforms other base models in almost all the data points, which demonstrates the effectiveness of the position-aware encoder with the attention mechanism. This justifies the reason for choosing PRNN as the base model for all semi-supervised models.

Secondly, moving on to semi-supervised baseline models, we observe a consistent performance gain when comparing RE-Ensemble with Self-Training. This is a strong evidence that ensemble of single models can boost the performance. However, Mean-Teacher as another self-ensembling method, does not have consistent gain over other models: It shows a clear win in TACRED dataset with 3% labeled data, but falls far behind in all other data points. The reason behind such phenomenon is still under investigation, but Mean-Teacher is clearly not as stable and strong as other semi-supervised baseline models.

Thirdly, our DualRE-Pointwise model outperforms all the baseline models except on TACRED with 3% labeled data. Its overall performance shows the effectiveness of DualRE framework. From the results, we attribute the performance gain to two design choices: (1) Ensemble method. Both DualRE-Pointwise and RE-Ensemble apply such strategy. By adding another prediction module, RE-Ensemble can already outperform its counterpart (Self-Training). (2) Introduction of dual task. Instead of replicating another prediction module as in RE-Ensemble, DualRE-Pointwise introduces the dual retrieval module. Two modules are jointly trained and complement each other. Thus we observe a consistent performance gain of DualRE-Pointwise over RE-Ensemble.

Lastly, speaking of the variant of DualRE, DualRE-Pairwise has comparable or even better performance as DualRE-Pointwise on the SemEval dataset but not on TACRED. Since DualRE-Pointwise generally performs better than DualRE-Pairwise, we use "DualRE" to refer to DualRE-Pointwise model in the following sections.

Performance on Different Datasets. One can clearly observe that the advantage of DualRE-Pairwise model over other baseline models becomes smaller on the TACRED dataset, comparing to SemEval. Considering the complexity and imbalance distribution in TACRED dataset over SemEval, it is more difficult to learn a strong retrieval module, especially for those rare relations. Thus DualRE models are not significantly better than other models.

Performance on Different Amounts of Labeled Data. From Table 2 and 3, when we fix the amount of unlabeled data to 50% of the original training set, we can observe that increasing the amount of labeled data effectively improves the model performance. Therefore, adding more labeled data (if accessible) is an efficient approach to boost the performance.

Performance on Different Amounts of Unlabeled Data. In specific domains where human-annotated training instances are expensive to obtain, we aim to leverage a large amount of unlabeled data to boost the performance of our model given limited amount of labeled data. Therefore, for semi-supervised learning, we may wonder whether more unlabeled data will always help. To investigate such a problem, we fix the amount of labeled data and compare

Table 3: Performance comparison on TACRED [39] with various amounts of labeled data and 50% unlabeled data. We report the mean and standard deviation of the evaluation metrics by conducting 3 runs of training and testing using different random seeds. DualRE outperforms all the baseline methods except Mean-Teacher at one data point.

| Methods / % Labeled Data	3%			10%			15%		
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
LSTM [17]	40.76 ± 6.62	23.05 ± 4.48	28.62 ± 3.01	50.56 ± 0.98	43.09 ± 1.26	46.51 ± 0.90	55.11 ± 0.56	44.41 ± 0.58	49.18 ± 0.52
PCNN [37]	58.16 ± 7.74	37.49 ± 6.16	44.64 ± 1.59	64.32 ± 7.78	42.06 ± 4.94	50.16 ± 1.15	67.10 ± 0.44	42.88 ± 0.42	52.32 ± 0.30
PRNN [39]	48.93 ± 5.72	33.05 ± 2.19	39.16 ± 0.90	53.44 ± 2.82	51.77 ± 1.88	52.49 ± 0.64	58.88 ± 2.32	51.30 ± 2.04	54.76 ± 0.93
Mean-Teacher (PRNN) [33]	53.08 ± 3.55	41.81 ± 0.61	**46.74 ± 1.70**	58.53 ± 2.56	50.08 ± 1.14	53.94 ± 0.91	57.90 ± 1.09	52.64 ± 0.97	55.13 ± 0.05
Self-Training (PRNN) [29]	49.89 ± 1.05	39.23 ± 2.26	43.86 ± 1.26	56.54 ± 0.72	53.00 ± 0.49	54.71 ± 0.09	60.09 ± 0.43	54.77 ± 0.55	57.31 ± 0.47
RE-Ensemble (PRNN)	56.48 ± 0.95	36.90 ± 0.91	44.62 ± 0.39	61.26 ± 0.58	52.51 ± 0.56	55.54 ± 0.29	60.76 ± 0.78	55.00 ± 1.04	57.72 ± 0.38
DualRE-Pairwise (PRNN)	58.97 ± 0.96	34.55 ± 1.18	43.55 ± 0.67	63.10 ± 0.94	48.91 ± 0.93	55.09 ± 0.25	60.99 ± 1.39	54.04 ± 0.46	57.30 ± 0.81
DualRE-Pointwise (PRNN)	52.76 ± 2.58	38.99 ± 2.08	44.73 ± 0.66	61.61 ± 1.30	52.30 ± 0.89	**56.56 ± 0.42**	60.66 ± 1.57	56.65 ± 0.37	**58.58 ± 0.69**
RE-Gold (PRNN w. gold labels)[*]	64.38 ± 0.46	60.35 ± 0.81	62.30 ± 0.29	65.88 ± 0.66	61.65 ± 0.42	63.70 ± 0.53	66.95 ± 2.78	59.97 ± 3.12	63.13 ± 0.56
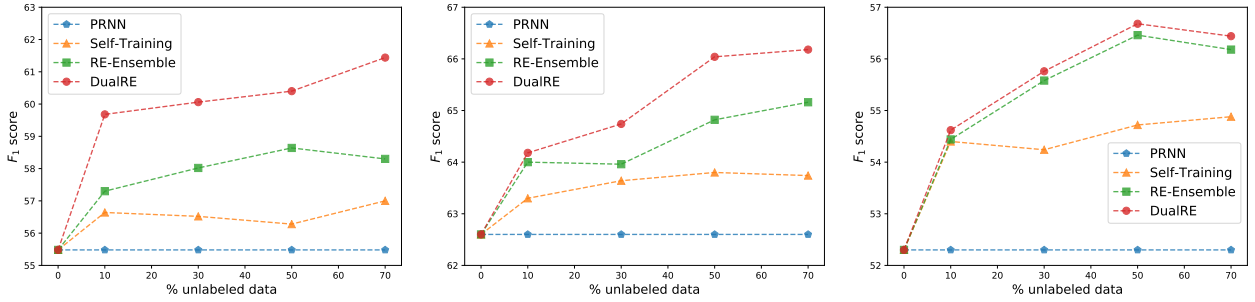
Figure 5: Performance w.r.t. the amount of unlabeled data. Left: 5% labeled data on SemEval; Middle: 10% labeled data on SemEval; Right: 10% labeled data on TACRED. In most cases, increasing unlabeled data results in better performance, but too much unlabeled data may hurt the performance.

Table 4: Comparison of Different Weighing Methods for Instance Promotion. We test with 10% labeled data and 50% unlabeled data on the SemEval dataset. Weighted promotion leads to significant improvement as compared with equal promotion.

Instance Weight	F_1 (Dev)	F_1 (Test)
α=0, β=0 (Equal Promotion)	62.40 ± 0.52	64.42 ± 0.82
α=1, β=1	63.84 ± 0.23	65.98 ± 0.46
α=1, β=2	64.22 ± 0.26	66.02 ± 0.15
α=0.5, β=2	**64.28 ± 0.46**	**66.04 ± 1.00**

the performance under different amounts of unlabeled data. More specifically, we investigate the model performance in SemEval and TACRED dataset, with 5% or 10% labeled data presented. In our experiment, 10%, 30%, 50% and 70% unlabeled data are available.

We summarize the results of DualRE and other methods in Fig. 5. One can observe that most models benefit from larger amount of unlabeled data in the range of 10% to 50%. However, increasing unlabeled data to 70% doesn't result in further improvement for all models. Half of them perform even worse with more unlabeled data. Therefore, it is not necessarily beneficial for semi-supervised models to take in as much unlabeled data as possible.

5.5 Case Study

Effectiveness of Weighted Promotion. We experiment with two different ways of assigning weights to pseudo-labeled instances
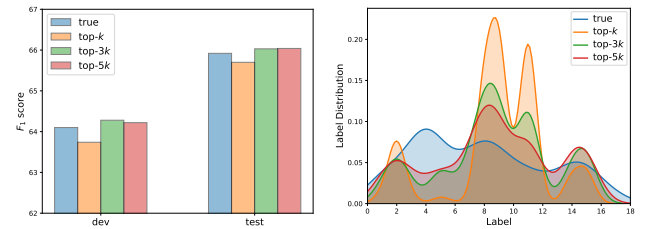

Figure 6: Case Study on Retrieval Module in DualRE. Left: Comparison of using different label distributions as the retrieval query; Right: A visualization of different label distributions in the first iteration. Both are evaluated with 10% labeled data and 50% unlabeled data on the SemEval dataset.

while training DualRE model. Specifically, different combinations of α and β are explored, according to Section 4.3 . Assigning α=0 and β=0, for instance, is to promote all instances equally in training. Results in Table 4 indicate that by using weighted promotion, we can get better results than equal promotion.

Choice of Reference Relation Distribution. In DualRE, when selecting instances to promote, the retrieval module takes in a reference relation distribution $q(y)$ as described in Sec. 4.2. Practically, we assign $q(y)$ with label distribution from available dataset. Two options are explored: (1) **true**: Use the label distribution from the labeled set L. (2) **top-nk**: Use the label distribution from the most confident $n \times k$ instances given by the prediction module.
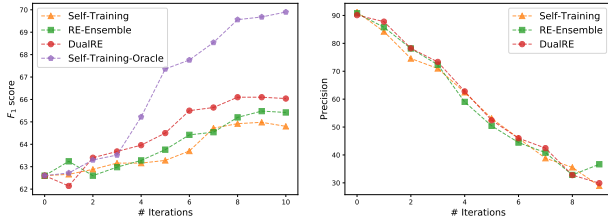
Figure 7: Analysis on Quality of Retrieved Instances. Left: Convergence curve of test F_1 score for different methods; Right: Precision of the relation mentions selected in each iteration. Both are evaluated with 10% labeled data and 50% unlabeled data on the SemEval dataset.

According to Fig. 6, it can be concluded that the best performance is achieved at top-$3k$, whose distribution lies between true and top-k. Intuitively, the retrieved instances work better if they follow the true distribution. Due to the unbalanced distribution, however, the long-tail relations usually have fewer instances and are harder to learn. The true relation distribution forces the model to attend to these relations and selection quality goes down. On the other hand, top-k distribution is too biased towards frequent relations. Retrieving more of these instances does not add much information for models to learn. Therefore, Top-$3k$ distribution can be regarded as a combination of above two strategies and works better.

Quality of Retrieved Instances. To investigate the underlying reason for performance gain in DualRE, we hypothesize that the quality (i.e. precision) of retrieved instances is higher in DualRE. In the study, we utilize the ground-truth labels of the "unlabeled data" to evaluate the precision of retrieved instances.

In our study, we evaluate the self training methods (Self-Training, RE-Ensemble and DualRE) using test set F_1 score in each iteration together with the precision of its promoted pseudo-labeled instances. Additionally, we construct an oracle model "Self-Training-Oracle" to identify the upper bound of self-training methods. Similar to Self-Training, we rank the instances by their probability of predicted label and select new instances in each iteration. But here we assume that the model can perfectly select those correctly pseudo-labeled instances and add them to the training set in each iteration.

Experimental results are shown in Fig. 7. Comparing results on DualRE and other semi-supervised models, the precision for retrieved instances in each iteration are generally higher in DualRE, which indicates that DualRE can select instances with better quality. This directly contributes to the performance gain in F_1 score in each iteration. To measure the upper bound of semi-supervised models that fall in the self-training category, the performance of Self-Training-Oracle indicates that the potential for such models are high enough to be comparable with RE-Gold (refer to Table 2). But this upper bound is extremely hard to approach because the model does not have additional supervision to guide the instance selection process.

6 RELATED WORK

Relation extraction. Our work is related to sentence-level relation extraction, which aims at predicting the relation label between two entities in a sentence. Traditional relation extraction methods [1, 26] seek to find discriminative textual pattern s with some labeled sentences, and further leverage these patterns for relation extraction. Nevertheless, the learned patterns may have difficulties matching with unseen sentences, as patterns are composed of discrete words. Recent studies leverage deep neural networks such as Long Short-Term Memory Networks (LSTM) [17] and Convolutional Neural Networks (CNN) [37] for relation extraction, achieving much better performance. However, all these methods are fully-supervised, which require a large amount of labeled data to train effective models. By contrast, our proposed approach is semi-supervised, which requires much fewer labeled sentences during training.

Semi-supervised learning. In the literature of semi-supervised learning, there are some methods which can be applied to our problem. One major category is the self-ensembling method [12], which assumes that the predicted labels of the unlabeled instances should remain unchanged under small perturbations on the instances or the model parameters. In this way, they guarantee the local consistency of the prediction and hence improve the performance. However, when applied to our problem, such a training framework relies solely on local consistency as supervision signal, which is weak compared to labeled data, therefore the improvement is marginal. Another category is the self-training method [29], which iteratively promotes instances from unlabeled data. Self-training methods impose a stronger supervision signal and generally achieve better performance. However, it suffers from semantic drift [9], which accumulates systematic errors from prediction model in each iteration. Co-training [2] is an attempt to eliminate this problem. It promotes unlabeled data using two models that are trained on different views of data, which essentially relies on the assumptions that two different sets of features are independent and sufficient for prediction. This assumption is not easy to follow in the relation extraction task, in which relation mention is the only source of input. Our DualRE model, instead, introduces a dual retrieval module to correct the bias from prediction module. Therefore we are able to collect extra training data with higher quality to avoid semantic drifting.

Dual learning. Our work is also related to dual learning [14, 32, 35], which can be generalized to the concept of jointly learning the primal and dual models to boost the performance of each other. For example, English-French translation and French-English translation [14], image classification and image generation [35], question answering and question generation [32] can be treated together to leverage the duality of two tasks. Different from their supervised settings, we apply the concept of dual model in semi-supervised settings and propose an algorithm that effectively utilize the unlabeled data. We are also the first to perform dual model learning in the relation extraction task.

7 CONCLUSION

In this paper, we studied the problem of relation extraction in the semi-supervised setting, where a few labeled sentences and an auxiliary unlabeled corpus are provided. We proposed a principled framework consisting of a prediction module and a retrieval module, which focus on the primal prediction task and the dual retrieval task respectively. During training, we encouraged them to mutually

enhance each other through annotating unlabeled sentences as additional training data. A joint optimization method was proposed for optimizing our framework, which alternates between updating the prediction module and updating the retrieval module. Experimental results on two public datasets proved the effectiveness of our proposed framework. Interesting future work directions include switching the retrieval module to text generative models, extending our framework to deal with various text classification tasks, and studying its connection to the variational auto-encoder method.

8 ACKNOWLEDGEMENT

REFERENCES

[1] Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *ACM DL'00*. 85–94.
[2] Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT'98*. 92–100.
[3] Sergey Brin. 1998. Extracting patterns and relations from the world wide web. In *International Workshop on The World Wide Web and Databases*. Springer, 172–183.
[4] Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *HLT-EMNLP'05*. 724–731.
[5] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 89–96.
[6] Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. 2006. Adapting ranking SVM to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 186–193.
[7] David Cossock and Tong Zhang. 2006. Subset ranking using regression. In *International Conference on Computational Learning Theory*. Springer, 605–619.
[8] Koby Crammer and Yoram Singer. 2002. Pranking with ranking. In *Advances in neural information processing systems*. 641–647.
[9] James R Curran, Tara Murphy, and Bernhard Scholz. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *PACLING'07*. 172–180.
[10] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977), 1–38.
[11] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*.
[12] Geoffrey French, Michal Mackiewicz, and Mark Fisher. 2017. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208* (2017).
[13] Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*. 1019–1027.
[14] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*. 820–828.
[15] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*. Association for Computational Linguistics, 94–99.
[16] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. 1995. The" wake-sleep" algorithm for unsupervised neural networks. *Science* 268, 5214 (1995), 1158–1161.
[17] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[18] Ping Li, Qiang Wu, and Christopher J Burges. 2008. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in neural information processing systems*. 897–904.
[19] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural Relation Extraction with Selective Attention over Instances.. In *ACL'16*. 2124–2133.
[20] Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331.
[21] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 55–60.
[22] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data.. In *ACL-IJCNLP'09*. 1003–1011.
[23] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence* (2018).
[24] Raymond J Mooney and Razvan C Bunescu. 2006. Subsequence kernels for relation extraction. In *NIPS'06*. MIT Press, 171–178.
[25] Gerhard Paass. 1993. Assessing and improving neural network predictions by the bootstrap algorithm. In *Advances in Neural Information Processing Systems*. 196–203.
[26] Meng Qu, Xiang Ren, and Jiawei Han. 2017. Automatic Synonym Discovery with Knowledge Bases. In *KDD'17*. 997–1005.
[27] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. 2015. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*. 3546–3554.
[28] Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. CoType: Joint extraction of typed entities and relations with knowledge bases.. In *WWW'17*. 1015–1024.
[29] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. 2005. Semi-Supervised Self-Training of Object Detection Models.. In *WACV/MOTION*. 29–36.
[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
[31] Ang Sun and Ralph Grishman. 2012. Active learning for relation type extension with local and global data views. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 1105–1112.
[32] Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027* (2017).
[33] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*. 1195–1204.
[34] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing Text for Joint Embedding of Text and Knowledge Bases.. In *EMNLP'15*. 1499–1509.
[35] Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. 2017. Dual supervised learning. *arXiv preprint arXiv:1707.00415* (2017).
[36] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research* 3, Feb (2003), 1083–1106.
[37] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1753–1762.
[38] Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Incorporating relation paths in neural relation extraction. In *EMNLP'17*. 1769–1778.
[39] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 35–45.