
Improving Relation Extraction by Leveraging Knowledge Graph Link Prediction

George Stoica

Carnegie Mellon University
5000 Forbes Ave,
Pittsburgh, PA 15213
gis@cs.cmu.edu

Emmanouil Antonios Platanios

Microsoft Semantic Machines
1 Microsoft Way,
Redmond, WA 98052
emplata@microsoft.com

Barnabás Póczos

Carnegie Mellon University
5000 Forbes Ave,
Pittsburgh, PA 15213
bapoczos@cs.cmu.edu

Abstract

Relation extraction (RE) aims to predict a relation between a subject and an object in a sentence, while knowledge graph link prediction (KGLP) aims to predict a set of objects, O , given a subject and a relation from a knowledge graph. These two problems are closely related as their respective objectives are intertwined: given a sentence containing a subject and an object o , a RE model predicts a relation that can then be used by a KGLP model together with the subject, to predict a set of objects O . Thus, we expect object o to be in set O . In this paper, we leverage this insight by proposing a multi-task learning approach that improves the performance of RE models by jointly training on RE and KGLP tasks. We illustrate the generality of our approach by applying it on several existing RE models and empirically demonstrate how it helps them achieve consistent performance gains.

1 Introduction

Many real-world applications ranging from search engines to conversational agents rely on the ability to uncover new relationships from existing knowledge. Relation extraction (RE) and knowledge graph (KG) link prediction (KGLP) are two closely related tasks that center around inferring new information from existing facts. RE is the task of uncovering the relationship between two entities (termed the subject and object respectively) in a sentence. Similarly, KGLP involves inferring the set of correct answers (i.e., objects) to KG questions consisting of an entity (subject) and relation. These questions are given in triple-form: (SUBJECT, RELATION, ?). To illustrate their relationship, consider the sentence “[John](#) and [Jane](#) are married”, whose subject and object are highlighted in [blue](#) and [red](#) respectively. Given this information, RE models infer the relationship between “[John](#)” and “[Jane](#)” (e.g., “Spouse”). Similarly, KGLP models infer the answers (objects) to the question ([John](#), Spouse, ?). Based on the sentence, the answers must include “[Jane](#)”. Thus, RE models predict the relation between a subject and object, while KGLP models infer the object from the subject and relation.

Several methods have been proposed to boost the performance of RE models by incorporating information from KGLP. However, these approaches typically require KGLP pre-training [32, 29], exhibit constrained parameter sharing [32, 29], or predominately attend over both problems through custom attention mechanisms [3, 11, 35]. Moreover, these frameworks only support a limited class of KGLP models that can be reframed as inferring relations from subject and objects. This constraint excludes recent KGLP methods which perform significantly better, but cannot be reformulated to satisfy the restriction. An ideal framework should support arbitrary RE and KGLP methods, including the significantly more expressive and stronger performing recent KGLP approaches. Additionally, such a framework should enable RE models to benefit from KGLP models with minimal changes to the underlying RE and KGLP methods.

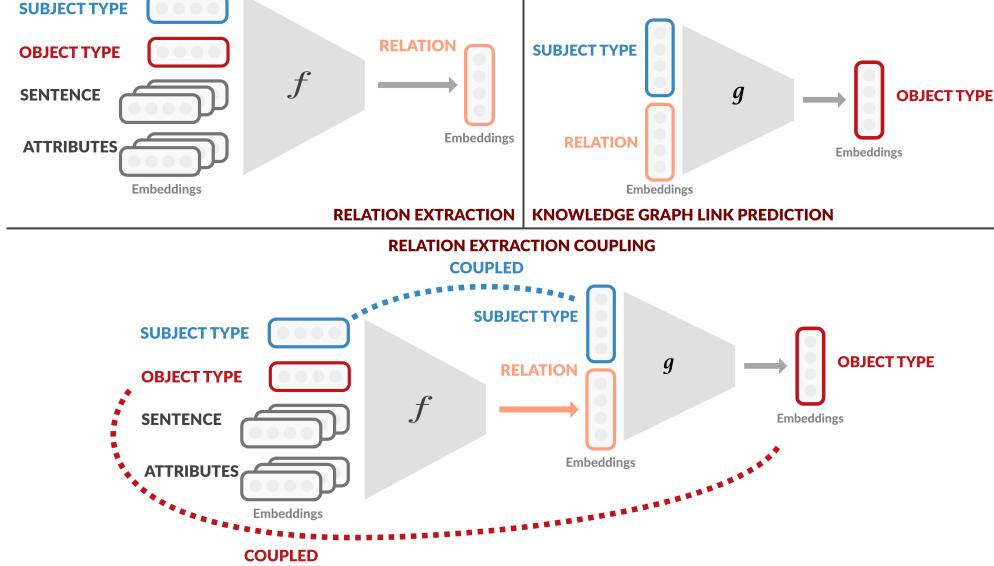


Figure 1: Overview of JRRELP. JRRELP is comprised of three loss terms: the RE loss, the KGLP loss, and the coupling loss. The RE loss is illustrated in the top-left quadrant, the KGLP loss is described by the top-right quadrant, and the bottom half shows the coupling loss.

We propose a general framework which ties the RE and KGLP tasks cohesively into a single learning problem. Our architecture, termed **JRRELP**—Jointly Reasoning over Relation Extraction and Link Prediction—has the following desirable properties:

- **Generality:** Our method can be applied to arbitrary RE and KGLP models to boost RE performance. The only assumption JRRELP makes is that both models are trained by minimizing a loss function (which is common across all successful RE and KGLP methods).
- **Effective Information-Sharing:** JRRELP introduces a *cyclical* relationship between model parameters, enabling better information transfer between the learning tasks. Moreover, *all* parameters are shared across both the RE and KGLP tasks.
- **Performance:** JRRELP boosts the performance of all baseline methods used in our evaluation. Additionally, JRRELP-enhanced baselines even match or improve upon the performance of more expressive RE models. For example, we are able to train C-GCN [37] to match TRE [1], even though the latter was proposed as a stronger and significantly more expressive alternative.
- **Efficiency:** JRRELP does not require any task-specific pre-training. It introduces a minimal overhead over the baseline methods (at most 6% slower per batch).

An overview of JRRELP is shown in Figure 1, and is explained in detail in Section 3. Next, we present our proposed method and defer positioning with respect to related work until Section 5.

2 Background

Before presenting our method, we introduce the notation used throughout this paper, and describe the relevant learning tasks. Let \mathcal{D} describe a dataset that contains a collection of sentences. Let $X = [x_1, x_2, \dots, x_n]$ denote a sentence, where x_i represents a one-hot encoding for the i^{th} sentence token (i.e., word). Each sentence contains a subject $s = [x_{s\text{start}}, x_{s\text{start}+1}, \dots, x_{s\text{end}}]$, that is defined as a contiguous span $(s^{\text{start}}, s^{\text{end}})$ over the sentence, and an object $o = [o_{o\text{start}}, o_{o\text{start}+1}, \dots, o_{o\text{end}}]$, that is similarly defined. Subjects and objects are summarized by their *types*, termed s^{type} and o^{type} , respectively. If not already given, these can be extracted by widely used parsing frameworks such as [19]. For example, consider the sentence “John Doe lives in Miami”, where the subject is shown in blue color and the object in red color. In this case, the subject may be tagged as having type **PERSON** and the object may be tagged as having type **CITY**. Several methods [e.g., 36, 37, 9] employ *type-substitution* during data preprocessing: substituting subjects and objects in sentences with their corresponding types. For instance, with type-substitution our example sentence becomes “**SUBJECT-PERSON SUBJECT-PERSON** lives in **OBJECT-CITY**.” For ease of future explanation, we assume that sentences are preprocessed using type-substitution for the remainder of this paper. Each sentence may contain additional structural features such as part-of-speech (POS) tags, named-

entity-recognition (NER) tags, and a dependency parse. Analogous to extracting entity types, these can be generated from parsing frameworks. We denote all such sentence features as members of a set C . Finally, each sentence contains a relation, r , between its subject and object. This may either describe their lack of connection (via a special NoRelation token), or an existing one. For instance, the relation between **John Doe** and **Miami** in our example sentence would be LivesIn. In summary, \mathcal{D} is a set of N tuples: $\mathcal{D} = \{(X_i, C_i, s_i, o_i, s_i^{\text{type}}, o_i^{\text{type}}, r_i)\}_{i=1}^N$, where N is the number of sentences.

2.1 Relation Extraction

Relation extraction (RE) uses X , C , s , and o from \mathcal{D} to infer the relation r between s and o . Note that due to our type-substitution constraint, this is analogous to predicting the relation r between s^{type} and o^{type} . Many successful models that have been proposed to tackle this task involve learning vector embeddings for each component. Specifically, let N_v , N_r , and N_c denote the vocabulary size for sentence tokens, the number of unique relations, and the number of unique attributes in C , computed over the whole training dataset. Additionally, let D_v , D_r , and D_c denote the corresponding embedding sizes. We define $\mathbf{V} \in \mathbb{R}^{D_v \times N_v}$, $\mathbf{R} \in \mathbb{R}^{D_r \times N_r}$, and $\mathbf{A} \in \mathbb{R}^{D_c \times N_c}$ as the vocabulary, relation, and attribute embedding matrices, respectively. Note that \mathbf{V} , \mathbf{R} , and \mathbf{A} are learnable model parameters. Given a sentence, a subject, an object, and its attributes, their respective embedding representations are defined as: $\mathbf{X} = \mathbf{V}X \in \mathbb{R}^{D_v \times n}$, $\mathbf{C} = \mathbf{A}C \in \mathbb{R}^{D_c \times c}$, $\mathbf{s}^{\text{type}} = \mathbf{V}s^{\text{type}} \in \mathbb{R}^{D_v}$, and $\mathbf{o}^{\text{type}} = \mathbf{V}o^{\text{type}} \in \mathbb{R}^{D_v}$, where n is the number of tokens in X and c is the number of attributes in C . Similarly, we define the embedded relation as $\mathbf{r} = \mathbf{R}r \in \mathbb{R}^{D_r}$. Given these embeddings, most successful RE models [e.g., 36, 37, 9, 1, 24] can be formulated as instances of the following model:

$$\mathbf{X} = \mathbf{V}X, \mathbf{C} = \mathbf{A}C, \mathbf{s}^{\text{type}} = \mathbf{V}s^{\text{type}}, \mathbf{o}^{\text{type}} = \mathbf{V}o^{\text{type}}, \quad \text{EMBEDDING} \quad (1)$$

$$\hat{\mathbf{r}} = f(\mathbf{X}, \mathbf{C}, \mathbf{s}^{\text{type}}, \mathbf{o}^{\text{type}}), \quad \text{PREDICTION} \quad (2)$$

$$p(r | \hat{\mathbf{r}}) = \text{Softmax}(\mathbf{R}\hat{\mathbf{r}} + \mathbf{b}), \quad \text{PROBABILITY ESTIMATION} \quad (3)$$

where $\hat{\mathbf{r}}$ is the inferred relation representation from a prediction model f . To demonstrate how multiple RE methods fit under this formulation, we briefly describe the three baseline models used in our experiments.

PA-LSTM. This model was proposed by Zhang et al. [36], and centers around formulating f as the combination of a one-directional long short-term memory (LSTM) network, and a custom position-aware attention mechanism. The sentence attributes it uses are POS and NER tags, as well as SO and OO tags representing the positional offset of each token from the subject and the object respectively. The method first applies the LSTM over the concatenated sentence, POS tag, and NER tag embeddings. A relation $\hat{\mathbf{r}}$ is then predicted by attending the LSTM outputs with a custom position-aware attention mechanism using the SO and OO tag embeddings.

C-GCN. This model was proposed by Zhang et al. [37], and formulates f as a graph-convolution network (GCN) over sentence dependency parse trees. It uses the same sentence attributes as PA-LSTM, and additionally the sentence dependency parse. Similar to PA-LSTM, the method first encodes a concatenation of the sentence, POS tag, and NER tag embeddings using a bi-directional LSTM network. The model then infers relations from these encodings by reasoning over the graph implied by a pruned version of the provided dependency tree parse. In particular, C-GCN computes the least common ancestor (LCA) between s and o , and uses the SO and OO tags to prune the tree around the LCA. Afterwards, C-GCN processes the sentence encodings using a graph convolution network (GCN) defined over the pruned dependency parse tree. The resulting representations are finally processed by a multi-layer perceptron to predict relations.

SpanBERT. This model was proposed by Joshi et al. [15], and is a strong performing BERT [7]-based relation extraction method. SpanBERT extends BERT by pre-training at the span-level. Moreover, the model randomly masks contiguous text spans instead of individual tokens, and adds a span-boundary objective that infers masked spans from surrounding data. In contrast to PA-LSTM and C-GCN, SpanBERT only takes into account the type-substituted sentence in its input to predict relations. f is formulated as its complete architecture, with C masked out. We chose this model because it is a strong performing BERT-based RE model and it is also open-sourced, allowing to easily integrate it in our experimental evaluation pipeline.

Note that PA-LSTM, C-GCN, and SpanBERT are just three of many approaches supported by our abstract RE model formulation. For instance, other transformer-based methods [1, 24, 23] can also be represented by using a different definition for f .

2.2 Knowledge Graph Link Prediction

The objective in knowledge graph link prediction (KGLP) is to infer a set of objects O given a question, $(s, r, ?)$, in the form of a subject-relation-object triple, missing the object. Typically, s and o are nodes in a knowledge graph (KG), while r represents a graph edge. Although \mathcal{D} does not necessarily provide an explicit KG to reason over, it is possible to generate one by assigning unique identifiers for all subjects, relations, and objects. For instance, these may be s^{type} and o^{type} for subjects and objects respectively, and the relation itself. Although we assume that these identifiers are used (as they are available in our training data $\mathcal{D}^{\text{train}} \subset \mathcal{D}$), we emphasize that our method is not limited to datasets with these characteristics. Instead our framework supports any \mathcal{D} that specifies a mapping to a pre-existing KG, or where it is possible to define other unique identifiers. This is a very weak constraint. Therefore, given a sentence with s , o , and r , we can use the subject and object types— s^{type} and o^{type} , respectively—to form a KG whose edges are represented by each r and nodes by each s^{type} and o^{type} . For ease of notation, we assume that each term is a one-hot encoding of the corresponding identifier.

Due to the *type-substitution* preprocessing step described in Section 2, all types are included in the sentence token vocabulary. Thus, we obtain KG component embeddings by: $s^{\text{type}} = \mathbf{V}s^{\text{type}} \in \mathbb{R}^{D_v}$, $o^{\text{type}} = \mathbf{V}o^{\text{type}} \in \mathbb{R}^{D_v}$, and $r = \mathbf{R}r \in \mathbb{R}^{D_r}$. Multiple existing KGLP methods can be characterized in terms of the following abstract model:

$$\mathbf{s}^{\text{type}} = \mathbf{V}s^{\text{type}}, \mathbf{r} = \mathbf{R}r, \quad \text{EMBEDDING} \quad (4)$$

$$\mathbf{z} = g(\mathbf{s}^{\text{type}}, \mathbf{r}), \quad \text{MERGE} \quad (5)$$

$$p(O | o^{\text{type}}, \mathbf{z}) = \text{Sigmoid}(\mathbf{V}_{o^{\text{type}}} \mathbf{z} + \mathbf{b}), \quad \text{PROBABILITY ESTIMATION} \quad (6)$$

where \mathbf{z} is a merged representation of s^{type} and r . Note that the set of available object embeddings $\mathbf{V}_{o^{\text{type}}} \subset \mathbf{V}$ contains *only* valid (in the type-checking sense) object embeddings. Previous work [25] shows that multiple KGLP methods fit under this formulation. While certain early KGLP methods [4, 33, 18, 14, 28] do not fit under this formulation, we note that they may be accommodated by a simple reconfiguration of Equation 6 to their respective scoring terms. We now provide the definition of ConvE [6] under this formulation, because we use ConvE as our KGLP model in our experiments. While we acknowledge that ConvE is not the current state-of-the-art (SoTA) KGLP approach, it performs very well while using only a fraction of the parameters current SoTA [25, 31] methods require, thus making it more efficient. Moreover, ConvE is an example of a KGLP method which cannot be restructured to infer r from s and o , making it infeasible to use with any of the previous joint RE and KGLP frameworks [e.g., 29, 32]. Note that, our results can only be further enhanced by using a stronger KGLP approach and thus this choice should not affect our conclusions.

ConvE. ConvE is defined by using the following merge function in our abstract model formulation:

$$g(s^{\text{type}}, r) = \text{Conv2D}(\text{Reshape}([s^{\text{type}}; r])), \quad \text{MERGE} \quad (7)$$

where “Conv2D” is a 2D convolution operation and “Reshape([$s^{\text{type}}; r$])” first concatenates s^{type} and r and then reshapes the resulting vector to be a square matrix, so that a convolution operation can be applied to it.

3 Proposed Method

As mentioned in Section 1, the RE and KGLP tasks are tightly coupled. Given a sentence X (e.g., “[Miami](#) is in [Florida](#)”) that contains a subject s (e.g., [Miami](#)) and an object o (e.g., [Florida](#)), the goal of RE is to predict the relation r (e.g., [locatedIn](#)), between s and o , that the sentence describes. Similarly, the goal of KGLP is to infer a set of objects O using r and s , such that the inferred objects correspond to correct subject-relation-object triples, and where $o \in O$ (this is known because the sentence X describes this relationship). Based on this observation, we propose JRRELP, a multi-task learning framework that explicitly accounts for this relationship between RE and KGLP. JRRELP trains a RE model, p_{RE} , that is defined using our abstract formulation from Section 2.1 and a KGLP model, p_{KGLP} , that is defined using our abstract formulation from Section 2.2, jointly, using four key ideas:

1. Parameter Sharing: p_{RE} and p_{KGLP} share all of the embedding parameters. This corresponds to the matrices \mathbf{V} , \mathbf{R} , and \mathbf{A} from Sections 2.1 and 2.2. Moreover, all parameters between RE and KGLP methods are also shared.

2. **Joint Training:** The two models are trained jointly by optimizing a single objective function. This function contains terms that correspond to the RE objective function, the KGLP objective function, as well as a prediction coupling loss term.
3. **Cyclical Coupling:** Our joint loss terms establish a cyclical relationship between the embedding parameters, that tightly couples the RE and KGLP tasks. This is because the RE model uses \mathbf{V} (which includes $\mathbf{V}_{o^{\text{type}}}$) to predict relation representations that are then compared to \mathbf{R} to produce distribution over relations. Reciprocally, the KGLP model uses \mathbf{R} to generate object embeddings that are compared to $\mathbf{V}_{o^{\text{type}}}$ to produce distributions over objects.
4. **Unmodified Evaluation:** JRRELP does not introduce any additional terms when evaluating p_{RE} . Thus, rather than enhancing p_{RE} by increasing its capacity, JRRELP does this by altering its training trajectory.

We now provide details on how each term of the joint training objective function is defined.

RE Loss. The first term corresponds to the standard loss function used to train the RE model. This loss function is defined as follows (where we use the notation introduced in Section 2.1):

$$\mathcal{L}_{\text{RE}} = \sum_{i=1}^N \text{SCE}(r_i, p_{\text{RE}}(r_i | X_i, C_i, s_i^{\text{type}}, o_i^{\text{type}})), \quad (8)$$

where ‘‘SCE’’ represents the softmax cross-entropy loss function, and p_{RE} is defined as in Equation 3:

$$p_{\text{RE}}(r_i | X_i, C_i, s_i^{\text{type}}, o_i^{\text{type}}) = \text{Softmax}(\mathbf{R}f_{\text{RE}}(\mathbf{X}_i, \mathbf{C}_i, \mathbf{s}_i^{\text{type}}, \mathbf{o}_i^{\text{type}}) + \mathbf{b}_{\text{RE}}), \quad (9)$$

where f_{RE} is the specific prediction function used by our RE model. Although this loss term assumes that a *single* relation exists between a subject and an object in a sentence, it is consistent with the loss term utilized by our baselines and is also appropriate for our widely used benchmark datasets described in Section 4. Additionally, we note that this does not restrict the applicability of JRRELP to single-relation extraction problems. For instance, ‘‘SCE’’ can be substituted for binary-cross entropy (BCE) in the case of having multiple applicable relations.

KGLP Loss. The second term corresponds to a popular loss function which is often used to train KGLP models. This loss function is defined as follows (where we use the notation introduced in Section 2.2):

$$\mathcal{L}_{\text{KGLP}} = \sum_{i=1}^N \text{BCE}(O_i, p_{\text{KGLP}}(O_i | s_i^{\text{type}}, o_i^{\text{type}}, r_i)), \quad (10)$$

where p_{KGLP} is defined as in Equation 6:

$$p_{\text{KGLP}}(O_i | s_i^{\text{type}}, o_i^{\text{type}}, r_i) = \text{Sigmoid}(\mathbf{V}_{o_i^{\text{type}}}g_{\text{KGLP}}(\mathbf{s}_i^{\text{type}}, \mathbf{r}_i) + \mathbf{b}_{\text{KGLP}}), \quad (11)$$

where g_{KGLP} is the specific merge function used by our KGLP model. Note here that O_i is a set of objects that can be constructed automatically given all of the training data and conditioned on s_i^{type} and r_i , as described in Section 2.2. We also acknowledge that certain KGLP methods [4, 33, 18, 14, 28] cannot be represented by this loss term. However, this does not detract from the generality of the proposed framework because they can be accommodated by changing this term to their respective objective functions.

Coupling Loss. The third term penalizes inconsistencies between the predictions of the RE and KGLP models. It is defined as follows:

$$\mathcal{L}_{\text{COUPLING}} = \sum_{i=1}^N \text{BCE}(O_i, p_{\text{COUPLING}}(O_i | X_i, C_i, s_i, o_i, s_i^{\text{type}}, o_i^{\text{type}})), \quad (12)$$

where:

$$p_{\text{COUPLING}}(O_i | \dots) = \text{Sigmoid}(\mathbf{V}_{o_i^{\text{type}}}g_{\text{KGLP}}(\mathbf{s}_i^{\text{type}}, \mathbf{f}_{\text{RE}}(\mathbf{X}_i, \mathbf{C}_i, \mathbf{s}_i^{\text{type}}, \mathbf{o}_i^{\text{type}}) + \mathbf{b}_{\text{KGLP}})), \quad (13)$$

where we have omitted the conditioning variables for brevity. The key difference between this loss term and the KGLP loss term is shown in red color. Specifically, the relations embeddings — \mathbf{r}_i — computed by r_i in the KGLP loss term, are replaced by the predicted relation embeddings $\hat{\mathbf{r}}_i$ from f_{RE} . This term aligns the RE and KGLP methods by making the first compatible with the second, and enhances the overall performance of our framework.

3.1 JRRELP Objective Function

The JRRELP objective function is formed by putting together the above three terms:

$$\mathcal{L}_{\text{JRRELP}} = \mathcal{L}_{\text{RE}} + \lambda_{\text{KGLP}} \mathcal{L}_{\text{KGLP}} + \lambda_{\text{COUPLING}} \mathcal{L}_{\text{COUPLING}}, \quad (14)$$

where $\lambda_{\text{KGLP}} \geq 0$ and $\lambda_{\text{COUPLING}} \geq 0$ are model hyperparameters that need to be tuned properly. We note that, while in principle λ_{KGLP} and $\lambda_{\text{COUPLING}}$ can vary independently, in our experiments we set both to the same value for simplicity and cheaper hyperparameter tuning. Furthermore, we observed no negative impact in performance.

Most importantly, due to the JRRELP parameter sharing and the use of this loss function, our framework introduces a cyclical relationship between the RE and KGLP models that couples them together very tightly. Specifically, the RE model predicts relation embeddings using V that it compares to R to produce distributions over relations. The KGLP model on the other hand predicts object embeddings using R that it compares to V to produce distributions over objects. It is mainly this cyclical relationship along with the coupling loss term that result in both the RE and KGLP models benefiting from each other and serves to enhance the performance and robustness of RE methods. An overview of JRRELP is shown in Figure 1.

Note that, even though JRRELP minimizes the joint three-task objective function shown in Equation 14, at test time we only use the RE model to predict relations between subjects and objects. Thus, JRRELP can be thought of as a framework which alters the learning trajectory of an RE model, rather than increase its capacity through using additional model parameters.

4 Experiments

We empirically evaluate the performance of JRRELP over two existing relation extraction baselines on two widely used supervised benchmark datasets. Our primary objective is to measure the importance of a joint RE and KGLP objective in environments where learning over both tasks is restricted *only* to data available in a relation extraction dataset. This serves to simulate how effective JRRELP may be in real-world applications where a pre-existing KG is not available for a given RE task. Additionally, we perform an ablation study to examine the impact each part of JRRELP has on its overall performance.

Datasets. We use the TACRED [36] and SemEval 2010 Task 8 [12] datasets for our experiments, which are commonly used in prior literature [e.g., 36, 37, 9, 24]. Table 1 shows their summary statistics. As mentioned in Section 2, for both datasets we utilize the following sentence attributes: NER tags, POS tags, subject/object offsets, and dependency tree structure. For the KGLP task in JRRELP, we construct the KG by generating $(s^{\text{type}}, r, o^{\text{type}})$ triples automatically, for each training sentence. We then ask questions of the form $(s^{\text{type}}, r, ?)$, where the answer belongs to a set of applicable objects O .

Setup. We perform our experiments on TACRED consistent with prior literature [36, 37, 9]. We use the same type-substitution policy where we replace each subject and object in a sentence with their corresponding NER types. Additionally, we evaluate our models using their micro-averaged F1 scores. Finally, we report the test metrics of the model with the best validation F1 score over five independent runs. While SemEval 2010 Task 8 is traditionally evaluated without type-substitution, Zhang et al. [37] point out that this causes models to overfit to specific entities, and does not test their ability to generalize to unseen data. They address this by masking these entities using their types. Therefore, to examine JRRELP’s generalization capabilities, we perform the same type-substitution procedure, and evaluate on the transformed dataset (denoted as SemEval-MM). Consistent with prior work [36, 37, 9, 1, 24], we report the macro-averaged F1 scores. Because SemEval(-MM) does not contain a validation set, we subsample 800 examples from the training set to use as a validation set.

Models. We illustrate the generality of JRRELP by evaluating it on baselines from both classes of RE approaches:¹ Two sequence-based models (PA-LSTM and SpanBERT), and a graph-based model (C-GCN). We join all three baselines with the KGLP method ConvE. We distinguish between our baselines and their JRRELP variants by boxing their model names (e.g. PA-LSTM is the JRRELP extended version of PA-LSTM). All models can be found in our repository: <https://github.com/gstoica27/JRRELP.git>.

¹Refer to Section 5 for their definitions.

Table 1: Dataset statistics. Here, # Train, # Validation, and # Test denote the number of questions used for training, validation, and testing. # Relations describes the number of distinct relation in each dataset, Avg. Tokens refers to the average number of tokens in each dataset sentence, and % Negatives indicates the percentage of data where there is "no relation" between subjects and objects.

Dataset	# Train	# Validation	# Test	# Relations	Avg. Tokens	% Negatives
TACRED	68,124	22,631	15,509	42	36.4	79.5%
SemEval-MM	8,000	-	2,717	19	19.1	17.4%

Table 2: Results reported by our own experiments are marked by *. The remainder are taken from [1] and [23]. All numbers are expressed as percentages. † denotes experiments performed using additional data other than provided by the respective models. “—” denotes missing results from the respective publications. “SemEval-MM” denotes the Masked-Mention version of the SemEval dataset.

Dataset	Metric	Models								
		C-AGGCN	TRE	BERT _{EM}	PA-LSTM	PA-LSTM	C-GCN	C-GCN	SpanBERT	SpanBERT
TACRED	Precision	73.1	70.1	—	65.7	67.8*	69.9	74.1*	69.2*	74.0*
	Recall	64.2	65.0	—	64.5	65.0*	63.3	61.9*	71.2*	67.3*
	F1	68.2	67.4	71.5†	65.1	66.4*	66.4	67.4*	70.2*	70.8*
SemEval-MM	Precision	—	—	—	75.2	74.8	76.5	76.9	81.2	82.7
	Recall	—	—	—	78.0	80.6	79.5	80.3	86.1	85.2
	F1	—	—	—	76.6	77.6	78.0	78.5	83.6	83.9

Results. We report our overall performance results on TACRED in Table 2. We observe that JRRELP consistently outperforms its baseline variants over their F1 and precision metrics. In particular, we find that JRRELP improves all baseline model performances by at least .6% F1, and yields improvements of up to 4.1% in precision. Furthermore, JRRELP bridges the performance gap between several methods, *without* altering their model capacities. Notably, JRRELP extended PA-LSTM matches the reported C-GCN performance, whose JRRELP variant matches TRE [1] — a significantly more expressive transformer-based approach. These results suggest that the true performance ceiling of reported relation extraction approaches may be significantly higher than their reported results, and that JRRELP serves as a conduit towards achieving these performances. Results on SemEval-MM indicate a similar pattern to TACRED: JRRELP improves performance across all baselines. This illustrates the effectiveness of JRRELP’s framework in environments with little data.

Table 3: TACRED F1 results from our ablation study. † denotes experiments conducted without $\mathcal{L}_{\text{COUPLING}}$, and ‡ marks those run without $\mathcal{L}_{\text{KGLP}}$.

Dataset	Metric	Ablation Experiments							
		PALSTM	PA-LSTM	PALSTM	PALSTM	C-GCN	C-GCN	C-GCN	C-CGCN
TACRED	F1	65.1	66.4	65.6	66.3	66.4	67.4	66.8	67.0
SemEval-MM	F1	76.6	77.6	76.8	77.3	78.0	78.5	78.1	78.4

Ablation Experiments. To examine the effects of JRRELP’s $\mathcal{L}_{\text{KGLP}}$ and $\mathcal{L}_{\text{COUPLING}}$ over the traditional relation extraction objective, \mathcal{L}_{RE} , we perform an ablation study with each term removed on methods from both RE approach classes: sequence-based (PALSTM) and graph-based (C-GCN). Table 3 shows the F1 results. Metrics for each dataset are reported in the same manner as previous results. All ablation performances illustrate the importance of $\mathcal{L}_{\text{KGLP}}$ and $\mathcal{L}_{\text{COUPLING}}$ as part of JRRELP’s framework, as their respective models are worse than the full JRRELP architecture: they exhibit performance drops up to .8% F1 respectively. Moreover, we observe the largest performance drop from the removal of $\mathcal{L}_{\text{COUPLING}}$ — which removes JRRELP’s consistency constraint between RE and KGLP models. This highlights importance of establishing this relationship while training to achieve strong performance.

5 Related Work

There are three areas of research that are related to the method we propose in this paper. In this section, we discuss related work in each area and position JRRELP appropriately.

Relation Extraction. Existing RE approaches can be classified in two categories: sequence-based, and graph-based methods. Given a sentence in the form of a sequence of tokens, sequence-based models infer relations by applying recurrent neural networks [38, 36], convolutional neural networks [34, 21, 30], or transformers [1, 24, 15, 23]. In addition to the sentence, graph-based methods use the structural characteristics of the sentence dependency tree to achieve strong performance. [22] apply an n-ary Tree-LSTM [26] over a split dependency tree, while [37, 9] employ a graph-convolution network (GCN) over the dependency tree.

Knowledge Graph Link Prediction. Existing KGLP approaches broadly fall under two model classes: single-hop and multi-hop. Given a subject and a relation, single-hop models infer a set of objects by mapping the subject and relation respectively to unique learnable finite dimensional vectors (embeddings) and jointly transforming them to produce an object set. These approaches can be translational [4] over the embeddings, multiplicative [33, 28], or a combination of the two [6, 18, 14, 2, 25, 31]. On the other hand, multi-hop approaches determine object sets by finding paths in the KG connecting subjects to the objects, and primarily consist of path-ranking methods [16, 8, 20, 10, 27, 5, 17].

Joint Frameworks. Several approaches [32, 11, 29, 35, 3] have explored using the additional supervision provided by a KG to benefit relation extraction model performance. Of these, we believe [32, 11, 29] are most similar to our work. [32] proposes a framework which utilizes a KGLP model, TransE [4], as an additional re-ranking term when evaluating an RE model. While employing TransE as a re-ranker improves performance, their framework trains TransE and the respective RE approach separately without parameter sharing. This only allows very restricted information sharing during evaluation. [11] proposes a dual-attention framework for jointly learning KGLP and RE tasks by computing a weight distribution over training data and shares parameters between tasks. However, like [32], [11] limits KGLP model selection to those which can be reformulated as inferring relations from subjects and objects. This excludes a large number of recent methods [6, 2, 5, 17, 25, 31] which cannot be reframed in this way. [29] also presents a joint framework, LFDS, for training relation extraction approaches via KGLP objectives. In particular, the architecture introduces a similar objective to $\mathcal{L}_{COUPLING}$, but can only support the same class of KGLP methods as in [32, 11]. Moreover, LFDS requires KGLP pre-training, and does not share core parameters such as relation representations between RE and KGLP methods. This can create domain-shift between the two respective models and impact performance.

JRRELP improves upon previous literature by providing a single joint objective which simultaneously addresses all their aforementioned limitations. First, JRRELP proposes an abstract framework which supports many RE and KGLP methods through three standard-based loss terms. Second, JRRELP shares all its parameters between KGLP and RE tasks, and establishes a novel cyclical learning structure over core parameters. Third, RE and KGLP tasks are jointly trained without any problem-specific pretraining required, enabling tasks to benefit from each other simultaneously during training. Fourth, JRRELP’s structure facilitates support for RE and KGLP methods with minimal implementation changes: only requiring their respective substitutions into f and g .

6 Conclusion

We propose JRRELP, a novel framework that improves upon existing relation extraction approaches by leveraging insights from the complementary problem of knowledge graph link prediction. JRRELP bridges these two tasks through an abstract multi-task learning framework that jointly learns RE and KGLP problems by unconstrained parameter sharing. We exhibit this generality by extending three diverse relation extraction methods, and improve their performances.

References

- [1] Christoph Alt, Marc Hübner, and Leonhard Hennig. Improving relation extraction by pre-trained language representations. *CoRR*, abs/1906.03088, 2019. URL <http://arxiv.org/abs/1906.03088>.
- [2] Ivana Balazevic, Carl Allen, and Timothy Hospedales. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural*

Language Processing (EMNLP-IJCNLP), pages 5185–5194, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1522. URL <https://www.aclweb.org/anthology/D19-1522>.

- [3] Iz Beltagy, Kyle Lo, and Waleed Ammar. Improving distant supervision with maxpooled attention and sentence-level supervision. *CoRR*, abs/1810.12956, 2018. URL <http://arxiv.org/abs/1810.12956>.
- [4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.
- [5] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2018.
- [6] Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, pages 1811–1818, February 2018. URL <https://arxiv.org/abs/1707.01476>.
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL <http://arxiv.org/abs/1810.04805>.
- [8] Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 833–838, 2013.
- [9] Zhijiang Guo, Yan Zhang, and Wei Lu. Attention guided graph convolutional networks for relation extraction. *CoRR*, abs/1906.07510, 2019. URL <http://arxiv.org/abs/1906.07510>.
- [10] Kelvin Guu, John Miller, and Percy Liang. Traversing knowledge graphs in vector space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 318–327, 2015.
- [11] Xu Han, Zhiyuan Liu, and Maosong Sun. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *AAAI*, 2018.
- [12] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/S10-1006>.
- [13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [14] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696, 2015.
- [15] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529, 2019. URL <http://arxiv.org/abs/1907.10529>.
- [16] Ni Lao, Tom Mitchell, and William W Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 529–539. Association for Computational Linguistics, 2011.
- [17] Xi Victoria Lin, Richard Socher, and Caiming Xiong. Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3243–3253, 2018.
- [18] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI*

- Conference on Artificial Intelligence*, AAAI’15, pages 2181–2187. AAAI Press, 2015. ISBN 0-262-51129-0. URL <http://dl.acm.org/citation.cfm?id=2886521.2886624>.
- [19] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. URL <http://www.aclweb.org/anthology/P/P14/P14-5010>.
 - [20] Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. Compositional vector space models for knowledge base completion. In *ACL*, 2015.
 - [21] Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado, June 2015. Association for Computational Linguistics. doi: 10.3115/v1/W15-1506. URL <https://www.aclweb.org/anthology/W15-1506>.
 - [22] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen tau Yih. Cross-sentence n-ary relation extraction with graph lstms, 2017.
 - [23] Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations, 2019.
 - [24] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. *CoRR*, abs/1906.03158, 2019. URL <http://arxiv.org/abs/1906.03158>.
 - [25] George Stoica*, Otilia Stretcu*, Emmanouil Antonios Platanios*, Barnabás Póczos, and Tom M. Mitchell. Contextual Parameter Generation for Knowledge Graph Link Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.
 - [26] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, abs/1503.00075, 2015. URL <http://arxiv.org/abs/1503.00075>.
 - [27] Kristina Toutanova, Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. Compositional learning of embeddings for relation paths in knowledge base and text. In *ACL*, 2016.
 - [28] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning (ICML)*, volume 48, pages 2071–2080, 2016.
 - [29] Guanying Wang, Wen Zhang, Ruoxu Wang, Yalin Zhou, Xi Chen, Wei Zhang, Hai Zhu, and Huajun Chen. Label-free distant supervision for relation extraction via knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2246–2255, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1248. URL <https://www.aclweb.org/anthology/D18-1248>.
 - [30] Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1123. URL <https://www.aclweb.org/anthology/P16-1123>.
 - [31] R. Wang, B. Li, S. Hu, W. Du, and M. Zhang. Knowledge graph embedding via graph attenuated attention networks. *IEEE Access*, 8:5212–5224, 2020.
 - [32] Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1366–1371, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/D13-1136>.
 - [33] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations (ICLR)*, 2015.

- [34] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL <https://www.aclweb.org/anthology/C14-1220>.
- [35] Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. *CoRR*, abs/1903.01306, 2019. URL <http://arxiv.org/abs/1903.01306>.
- [36] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1004. URL <https://www.aclweb.org/anthology/D17-1004>.
- [37] Yuhao Zhang, Peng Qi, and Christopher D. Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1244. URL <https://www.aclweb.org/anthology/D18-1244>.
- [38] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-2034. URL <https://www.aclweb.org/anthology/P16-2034>.