Integrating Local Context and Global Cohesiveness for Open Information Extraction

Qi Zhu[1], Xiang Ren[2], Jingbo Shang[1], Yu Zhang[1], Ahmed El-Kishky[1], Jiawei Han[1]

[1]University of Illinois at Urbana-Champaign, IL, USA
[2]University of Southern California, CA, USA
[1]{qiz3, shang7, yuz9, elkishk2, hanj}@illinois.edu [2]xiangren@usc.edu

ABSTRACT

Extracting entities and their relations from text is an important task for understanding massive text corpora. Open information extraction (IE) systems mine relation tuples (i.e., entity arguments and a predicate string to describe their relation) from sentences. These relation tuples are not confined to a predefined schema for the relations of interests. However, current Open IE systems focus on modeling *local* context information in a sentence to extract relation tuples, while ignoring the fact that *global* statistics in a large corpus can be *collectively* leveraged to identify high-quality sentence-level extractions. In this paper, we propose a novel Open IE system, called ReMine, which integrates local context signals and global structural signals in a unified, distant-supervision framework. Leveraging facts from external knowledge bases as supervision, the new system can be applied to many different domains to facilitate sentence-level tuple extractions using corpus-level statistics. Our system operates by solving a joint optimization problem to unify (1) segmenting entity/relation phrases in individual sentences based on local context; and (2) measuring the quality of tuples extracted from individual sentences with a translating-based objective. Learning the two subtasks jointly helps correct errors produced in each subtask so that they can mutually enhance each other. Experiments on two real-world corpora from different domains demonstrate the effectiveness, generality, and robustness of ReMine when compared to state-of-the-art open IE systems.

1 INTRODUCTION

With the emergence of massive text corpora in many domains and languages, the sheer size and rapid growth of this new data poses many challenges understanding and extracting insights from these massive corpora. Information extraction (IE) [31] – extraction of relation tuples in the form of ⟨*head entity*, relation, *tail entity*⟩ – is a key step towards automating knowledge acquisition from text. In Fig. 1, for example, the relation tuple ⟨*Louvre-Lens*, build, *new satellites*⟩ can be extracted from unstructured text s_2 to represent a piece of factual knowledge in structured form. These relation tuples have a variety of downstream applications, such as serving as building blocks for knowledge base construction [11] and facilitating

WSDM '19, Melbourne, VIC, Australia

DOI: 10.1145/3289600.3291030

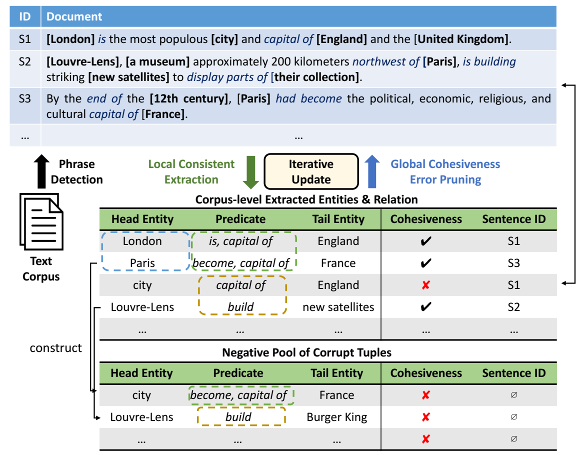


Figure 1: Example of incorporating global cohesiveness view for error pruning. One can infer "London" and "Paris" are similar because they co-occur a lot with the same relation in corpus. By constructing false tuples from extractions, "city" occurs with relation "capital of" in the negative pool more often, then it is unlikely for tuples with "city" and "capital of" to be correct.

question answering systems [13, 35]. While traditional IE systems require people to pre-specify the set of relations of interest, recent studies on *open-domain information extraction* (Open IE) [3, 8, 32] rely on *relation phrases* extracted from text to represent the entity relationship, making it possible to adapt to various domains (*i.e.*, open-domain) and different languages (*i.e.*, language-independent).

Current Open IE systems focus on analyzing the local context within individual sentences to extract entities and their relationships, while ignoring the redundant information that can be collectively referenced across different sentences and documents in the corpus. For example, in Fig. 1, seeing entity phrases "*London*" and "*Paris*" frequently co-occur with similar predicate strings and tail entities in the corpus, one gets to know that they have close semantics (same for "*England*" and "*France*"). This not only helps confirm that ⟨*London*, is capital of, *England*⟩ is a quality tuple as we know ⟨*Paris*, become capital of, *France*⟩ is extracted with high confidence, but this also rules out the tuple ⟨*city*, capital of, *England*⟩ as "*city*" is semantically distant from "capital of". Therefore, the information redundancy in the massive corpus provides clues on whether a candidate relation tuple is consistently used in the corpus, and motivates us to design a principled way of measuring tuple quality (*i.e.*, global cohesiveness).

Furthermore, most existing Open IE systems assume that they have access to entity detection tools (*e.g.*, named entity recognizer (NER), noun phrase (NP) chunker) to extract entity phrases from

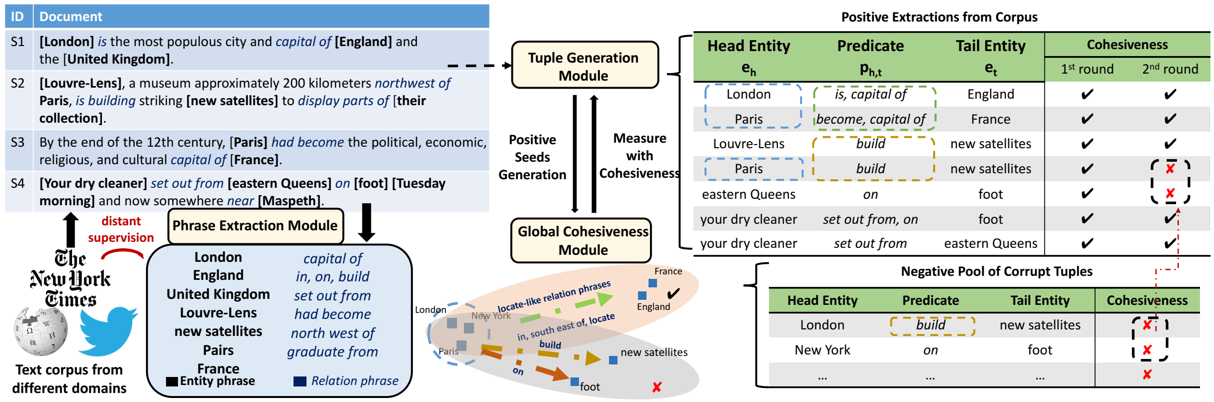


Figure 2: Overview of the ReMine Framework.

sentences, which are then used to form entity pairs for relation tuple extraction [3, 8, 32]. Some systems further rely on dependency parsers to generate syntax parse trees to guide the relation tuple extraction [2, 10, 32]. However, these systems suffer from *error propagation* as the errors in prior parts of the pipeline(*e.g.*, entity recognition) could accumulate by cascading down the pipeline(*e.g.*, to relation tuple extraction), yielding more significant errors. In addition, the NERs and NP chunkers are often pre-trained for general domain and may not work well on a domain-specific corpus (*e.g.*, scientific papers, social media posts).

In this paper, we propose a novel framework, called ReMine, to unify two important yet *complementary* signals for the Open IE problem, *i.e.*, the local context information and global cohesiveness (see also Fig. 2). While most existing Open IE systems focus on analyzing local context and sentence structures for tuple extraction, ReMine further makes use of all the candidate tuples extracted from the entire corpus, to collectively measure whether these candidate tuples are reflecting cohesive semantics. This is done by mapping both entity and relation phrases into the same low-dimensional embedding space, where two entity phrases are similar if they share similar relation phrases and head/tail entity phrases. The entity and relation embeddings so learned can be used to measure the cohesiveness score of a candidate relation tuple. To overcome the error propagation issue, ReMine *jointly* optimizes both the *extraction of entity and relation phrases* and the *global cohesiveness across the corpus*, each being formalized as an objective function so as to quantify the quality scores, respectively.

The major contributions of this paper are as follows.

(1) We propose a novel open IE framework, ReMine, that can extract relation tuples with local context and global cohesiveness.
(2) We develop a context-dependent phrasal segmentation algorithm that can identify high quality phrases of multiple types.
(3) We design a unified objective to measure both tuple quality in a local context and global cohesiveness of candidate tuples.
(4) Extensive experiments on three public datasets demonstrate that ReMine achieves state-of-the-art performance on both entity phrase extraction task as well as Open IE task.

2 PROBLEM DEFINITION

Notations. For any sentence s in a corpus $\mathcal{D}$, a *phrase*, *p*, is defined as single-word or multi-word phrase in s. We further group phrases into three different types, *i.e. entity phrase e*, *relation phrase r* and *background text b*. In Open IE, an *entity phrase* occurs as subject or object in extractions. In practice, entity phrase can be either a named entity of pre-defined types(*e.g.*, *time, location, person, organization*) or other noun phrases. In sentence s_4 of Fig. 2, "Your dry cleaner" is not a named entity, although it is the subject of this sentence and cannot be omitted in relation tuples extraction. Therefore, previous work [12, 32] use pre-trained NP chunkers to identify entity phrases. *Positive entity phrase pairs* E_p^+ is a set of entity pairs that may have textual relations between them. *Relation phrase r* describes relation between an entity phrase pair $(e_h, e_t) \in E_p^+$. Unlike relation extraction tasks, one relation instance can correspond to multiple relation phrases, e.g. *location/country/capital can correspond to ('s capital, capital of, the capital, ...)*. Lastly, *background text* is not a component of relation tuple.

Problem. Let $\mathcal{T}$ denote the extracted relation tuples. Each *relation tuple t* is defined as $\{e_h, p_{h,t}, e_t\}$, where e_h and e_t correspond to head and tail entity arguments and predicate $p = (r_1, r_2, ...r_n)$ may contain multiple relation phrases(*e.g.*, we have two relation phrases: "had become" and "capital of" between ⟨Paris, France⟩ in sentence s_3). Formally, we define the task of Open IE as follows.

Definition 2.1. Given corpus $\mathcal{D}$, the task of Open IE ***aims to****: (1) segment sentence $s \in \mathcal{D}$ to extract entity phrases e, relation phrases r; and (2) output relation tuples $\{e_h, p_{h,t}, e_t\}_{k=1}^{N_t}$.*

3 THE REMINE FRAMEWORK

ReMine aims to jointly address two problems: extracting entity and relation phrases from sentences and generating quality relation tuples. To accomplish this, we must first address three challenges. First, as the phrase boundary and category are unknown, one needs to design a segmentation algorithm to score the quality of segmented phrases and label their categories. Second, as multiple entity phrases may be extracted from a sentence, one needs to identify positive entity phrase pairs and obtain proper relation phrase between them. Third, as tuple extraction based solely on local sentence context may be error-prone, one needs to incorporate corpus-level statistics to help correct errors.

Framework Overview. We propose a framework, called ReMine, that integrates both local context and global structure cohesiveness (see also Fig. 2) to address above challenges. ReMine has three major modules, each focusing on address one challenge mentioned above: (1) phrase extraction module; (2) tuple generation module; and (3) global cohesiveness module. First, to extract quality phrases with different categories, the phrase extraction module trains a robust phrase classifier using existing entity phrases from external knowledge base as “distant supervision” and adjust quality iteratively. Second, the tuple extraction module generates candidate tuples based on sentence's language structure—it adopts widely used local structure patterns [10, 27, 32], including syntactic and lexical patterns over pos tags and dependency parsing tree. Different from previous studies, the module incorporates corpus-level information redundancy. Last, the global cohesiveness module learns entity and relation phrase representation and uses the representation in a score function to rank tuples. By collaborating with each other, the relation tuple generation module and the global cohesiveness module mutually enhance each other's results. Particularly, the relation tuple generation module produces candidate relation tuples(as positive tuples) and feeds them into the global cohesiveness module. By distinguishing positive tuples with constructed negative samples, the global cohesiveness module provides a cohesiveness measure for candidate tuples. The tuple generator further incorporates global cohesiveness into local generation and outputs more precise extractions. ReMine integrates tuple generation and global cohesiveness learning into a joint objective. Upon convergence, the training process results in distinctive and accurate tuples. Overall, ReMine extracts relation tuples as follows, see also Fig. 2:

(1) **Phrase extraction module** conducts context-dependent phrasal segmentation on a target corpus (using distant supervision) , to generate entity phrases, relation phrases and sentence segmentation probability $\mathcal{W}$.
(2) **Tuple generation module** generates positive entity pairs and identifies predicates p between each entity phrase pair (e_h, e_t).
(3) **Global cohesiveness module** learns entity and relation representations $\mathcal{V}$ via a translating objective to capture global structure cohesiveness σ.
(4) **Iteratively update extractions** $\mathcal{T}$ based on both local context information and global structure cohesiveness.

3.1 Phrase Extraction Module

Example 3.1 (Multi-type phrasal segmentation).

[London] ***is*** the most populous **[city]** and ***captital of*** **[England]** and the **[United Kingdom]**.

entity phrases in [], relation phrases in *italic* and all the others are background text.

We address entity and relation phrase extraction as a multi-type phrasal segmentation task. Given word sequence C and corresponding linguistic features $\mathcal{F}$ in Table 2, a phrasal segmentation $\mathcal{S} = s_1, s_2, ..., s_n$ is separated by boundary index $B = b_1, b_2, ..., b_{n+1}$. For each segment s_i, there is a type indicator $t_i \in \{e, r, b\}$[1], indicating the most possible type of s_i. In above example 3.1, s_0 = *London*, $t_0 = e$. We factorize the phrasal segmentation probability as:

$$P(C|\mathcal{F}) = \prod_{i=1}^{n} P(b_{i+1}, s_i | b_i, \mathcal{F}) \tag{1}$$

ReMine generates each segment as follows,
1. Given the start index b_i, generate the end index b_{i+1} according to context-free prior $P(b_{i+1} - b_i) = \delta^{|b_{i+1} - b_i|}$, *i.e.* length penalty [21].
2. Given the start and end index (b_i, b_{i+1}) of segment s_i, generate a word sequence s_i according to a multinomial distribution over all segments of the same length.

$$P(s_i | b_i, b_{i+1}) = P(s_i | b_{i+1} - b_i) \tag{2}$$

[1] e:entity phrase, r:relation phrase, b:background text

Table 1: Entity and relation phrase candidates generation with regular expression patterns on part-of-speech tag

Pattern	Examples
Entity Phrase Patterns	
<DT|PP$>?<JJ>*<NN>+	*the state health department*
<NNP>+<IN>?<NNP>+	*Gov. Tim Pawlenty of Minnesota*
Relation Phrase Patterns	
{V=<VB|VB*>+}	*furnish, work, leave*
{V}{P=<NN|JJ|RP|PRP|DT>}	*provided by, retire from*
{V}{W=<IN|RP>?*}{P}	*die peacefully at home in*

3. Finally, we generate a phrase type t_i indicating the type of s_i and a quality score showing how likely it is to be a good phrase $\lceil s \rfloor$.

$$P(\lceil s_i \rfloor | s_i) = \max{}_{t_i} P(t_i | s_i) = Q_{t_i}(s_i) \tag{3}$$

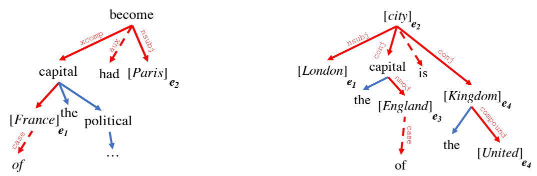
Phrase type t and quality Q are determined by a random forest classifier with robust positive-only distant training [33], which uses phrases in external knowledge base as positive samples and draws a number of phrases from unknown candidates as negative samples. Among all word sequence s_i, we denote unique phrase as u and $P(s_i | b_{i+1} - b_i)$ as θ_u. Similar with [21], we use Viterbi Training [1] to find best segmentation boundary B and parameters θ, δ iteratively. In the E-step, given θ and δ, dynamic programming is used to find the optimized segmentation. In the M-step, we first fix parameter θ, and update context-dependent prior δ. Next when δ is fixed, optimized solution of θ_u is:

$$\theta_u = \frac{\sum_{i=1}^{m} \mathbf{1} \cdot (s_i = u)}{\sum_{i=1}^{m} \mathbf{1} \cdot (b_{i+1} - b_i = |u|)} \tag{4}$$

Phrase Mining [20, 33] makes an assumption that quality phrases can only be frequent n-grams within a corpus. To overcome the phrasal sparsity of this assumption, several NP-chunking rules [12] in Table 1, are adopted to discover infrequent but informative phrase candidates. In experiment 4.2, ReMine has better performance than AutoPhrase [33] as we consider more phrase candidates and multi-type segmentation helps exclude relation phrases and background text better in entity phrase extraction task.

3.2 Tuple Generation Module

Generating Candidate Entity Pairs. For a given sentence s, after phrase segmentation, we have entity phrases $e_1, e_2, ..., e_n$ and relation phrases $r_1, r_2, ..., r_n$. However, it's computationally intractable to explore possible relationships between every entity pair and a large portion of tuples are incorrect among $n(n-1)$ pairs. E_p^+ are candidate entity phrase pairs. Here we heuristically initialize E_p^{+0} by attaching the *nearest* subject e_i(within the sentence) to the object e_j and make an approximation that each entity argument phrase can only be an object once; this also guarantees entity pairs to be distinctive. The nearest subject of e_j is defined as the entity e_i that has the shortest dependency path length to e_j among all other entities. Considering Fig. 3b, we would like to find the subject of entity e_4 : *United Kingdom*, the lengths of the shortest paths between e_4 and e_1, e_2, e_3 are 2,3,1. For those entity candidates with the same distance, see Fig. 3b, both e_1: *London* and e_4: *United Kingdom* is one hop away from e_2: *city*. In this situation, we will prefer the subject with “nsubj” type *i.e.* e_1. If there are still multiple entities, we will choose the closest entity in the original sentence.



(a) Finding semantic path **(b) Positive Entity Pairs Initialization**

Figure 3: Dependency parsing tree of example sentences s_1 and s_3 in Fig. 2, Segmented entities are marked as "[entity_token]$_{e_i}$"

Semantic Path Generation. Once $(e_h, e_t) \in E_p^+$ is determined, the semantic path is defined as the shortest dependency path between two arguments. Compared with using word sequence between (e_h, e_t) directly, the semantic path helps cloud irrelevant information. For example, in Fig. 3a, the semantic path between *"Paris"* and *"France"* of sentence s_3 is marked in red, where word sequence"the political, economic..." is correctly excluded. To preserve integrity of potential relation phrases, we further include particles and preposition along the shortest dependency path as part of the semantic path, which is shown as red dotted line in Fig. 3a.

Definition 3.2. **(Semantic Path)** *For an entity phrase pair (e_h, e_t) in the same sentence, the semantic path is defined as word sequence $SP_{h,t}$ along expanded dependency path.*

Example 3.3 (Generating Relation Tuples). Extracting relation phrases on the semantic path

[Paris] + <u>had become</u> + <u>capital of</u> + **[France]**

We now present how we generate valuable relation tuples according to semantic path, *i.e.*

$$\begin{aligned} \mathbf{P}(r, e_h, e_t) &= \prod_{i=1}^{n} \mathbf{P}(r_i | s_i, e_h, e_t) \mathbf{P}(s_i | b_i, b_{i+1}) \\ \max_{p_{h,t}} \mathbf{P}(r, e_h, e_t) &\Rightarrow \sum_{r_i \in p_{h,t}} \log \sigma(r_i, e_h, e_t) + \log w_i \end{aligned} \tag{5}$$

where $b_1, b_2, ..., b_{n+1}$ are boundary index along semantic path $SP_{h,t}$ of entity phrase pair(e_h, e_t). $\mathbf{P}(s_i | b_i, b_i + 1)$ is inherited from phrase extraction module as sentence segmentation probability w_i, then ReMine judges whether it is a good relation between entity e_h and entity e_t. Notice that the relation phrase boundary $i \in p_{h,t}$ in equation 5 can be derived via dynamic programming since w_i and σ is known for every possible segmentation. In example 3.3, within entity pair ⟨*Paris*, *France*⟩, the semantic path is "had become capital of". Tuple ⟨*Paris*, had become|capital of, *France*⟩ will be generated as both relation phrases had become and capital of are coherent with global cohesiveness measure σ and w_i.

3.3 Global Cohesiveness Module

Inevitably, false tuple like ⟨*city*, is capital of, *England*⟩ will be generated by relation tuple generation module as introduced in Sec 3.2, since the nearest subject of *England* in Fig. 3b is *city*. To get rid of such false tuples, current methods use textual patterns [10, 32] to identify it as a false extraction. In contrast, we design global cohesiveness measure using corpus-level statistics, and integrate the measure with the relation tuple generation. To capture the global cohesiveness of relation tuples, we adopt translation-based multi-relational data representation [5].

$$\sigma(p, h, t) = -d(h + p, t); \quad d(h + p, t) = \|v_h + v_p - v_t\| \tag{6}$$

where v_h, v_t are embeddings for head and tail entities, p is the predicate. Such a measure imposes reliable relation tuples on small translating distance between $h+p$ and t. We use L_1 norm in ReMine.

Based on initial positive entity pairs constructed E_p^{+0} and relation tuples $\mathcal{T}$, we construct a pseudo knowledge graph. Particularly, predicate $p_{h,t}$ may contain several relation phrases. Motivated by process of knowledge traverse [15], we average multiple relation phrases embeddings to represent the predicate *i.e.* $v_p = \sum_{i=1}^{n} v_{r_i}/n$.

Example 3.4 (Generating False Tuples). ⟨*Paris*, become capital of, *France*⟩ → ⟨***city***, become capital of, *France*⟩, ⟨*Paris*, become capital of, ***Burger King***⟩

In order to learn a global cohesiveness representation $\mathcal{V}$, we construct correlated negative tuples from positive seeds. For instance, as seen for example 3.4, we see that for a positive tuple, we can generate many incorrect or "negative" tuples.

The cohesiveness measure σ is optimized by maximizing the cohesiveness margin between positive and negative tuples,

$$\max_{\mathcal{V}} \sum_{p,h,t}^{\mathcal{T}} \sum_{p,h',t'}^{\mathcal{T}^-} [\sigma(p, h, t) - \sigma(p, h', t') - \gamma]_- \tag{7}$$

where $[x]_-$ denotes the negative part of x, $\mathcal{T}$ denotes positive relation tuples generated by local relational extraction, γ is the hyper margin, $(p, h', t') \in \mathcal{T}^-$ is composed of training tuples with either **h** or **t** replaced.

3.4 The Joint Optimization Problem

Relation tuple generation in Section 3.2 incorporates cohesiveness similarity σ. Additionally global cohesiveness measure learning depends on extracted tuples $\mathcal{T}$. We now show how local context and global cohesiveness introduced above can be integrated.

Overall Updating Schema. The final objective for update is formulated as the sum of both sub-objectives,

$$\max_{\mathcal{V},\mathcal{T}} O = O_{local} + O_{global} \tag{8}$$

$$O_{local} = \sum_{E_p^+} \log \mathbf{P}(p_{h,t}, e_h, e_t) \tag{9}$$

$$O_{global} = \sum_{p,h,t}^{\mathcal{T}} \sum_{p,h',t'}^{\mathcal{T}^-} [\sigma(p, h, t) - \sigma(p, h', t') - \gamma]_- \tag{10}$$

To maximize the above unified open IE objective, see Alg. 1, we first initialize positive entity pairs E_p^{+0}. Given entity phrase pairs, we perform local optimization, which leads to positive relation tuples $\mathcal{T}$. Note that, at the first round, there is no global representation, so we initialize all $\sigma = 1$ as identical. Then we update global phrase semantic representation via stochastic gradient descent. With both global cohesiveness information and local segmentation results, ReMine updates relation tuples as described in Alg. 1. Overall ReMine solves the integrated problem greedil and it iteratively updates local and global objectives until a stable E_p^+ is reached.

Example 3.5 (Updating Relation Tuples). In sentence s_1,

⟨*city*, is capital of, *England*⟩ → ⟨*London*, is capital of, *England*⟩

Update Positive Entity Pairs and Relation Tuples. Given a semantic representation for each entity **e** and relation **r** and local segmentation between entity pairs, we can update the *Positive Entity Pairs* by finding the most semantically consistent subject e_h for each object $\mathbf{e}_t$ among M_{sp}-nearest neighbors on the dependency

Algorithm 1: The ReMine Algorithm for Joint Optimization

Input: corpus $\mathcal{D}$, sentence S, text features $\mathcal{F}$, convergence threshold t
Output: relation tuples $\mathcal{T}$, semantic representation $\mathcal{V}$, similarity measure σ

generate entity and relation seeds via distant corpus linking ;
phrase extraction module outputs entity phrases, relation phrases, sentence segmentation probability $\mathcal{W}$;
initialize positive E_p^{+0}, cohesiveness measure $\sigma = 1$;
generate relation tuples $\mathcal{T}$ among E_p^{+0} ;
do
  update $\mathcal{V}$, σ in Eq. (10) via **global cohesiveness module** ;
  $E_p^{+n} \leftarrow \emptyset, \Delta_E \leftarrow 0$;
  for *each tuple* $\langle e_h, p_{ht}, e_t \rangle \in \mathcal{T}$ **do**
    construct candidate subject sets **s** of e_t with at most M_{sp} entities;
    $\sigma_* \leftarrow \sigma(e_h, p_{h,t}, e_t)$;
    for *i = 1 to* M_{sp} **do**
      generate $\langle s_i, p_{i,t}, e_t \rangle$ in Eq. (9) via **relation tuple generation module** given $\mathcal{W}$ and $\mathcal{V}$;
      if $\sigma(s_i, p_{i,t}, e_t) > \sigma_*$ **then**
        $\sigma_* \leftarrow \sigma(s_i, p_{i,t}, e_t), e_* \leftarrow s_i$;
      end
    end
    $E_p^{+n} \leftarrow E_p^{+n} \bigcup \langle e_*, e_t \rangle$;
    if $e_* \neq e_h$ **then**
      $\Delta_E \leftarrow \Delta_E + 1$, $\langle e_*, p_{*t}, e_t \rangle \leftarrow \langle e_h, p_{ht}, e_t \rangle$ update $\mathcal{T}$;
    end
  end
while $\frac{\Delta_E}{|E_p^{+n}|} > t$;

Table 2: Features used in the phrase extraction module (Sec. 3.1).

Feature	Descriptions
popularity	raw frequency, occurrence probability
completeness	whether can be interpreted as a complete semantic unit
concordance	tokens in quality phrases should co-occurs frequently
punctuation	phrase in parenthesis, quote or has dash after
stopwords	first/last token is stopword and stopword ratio
word shape	first capitalized or all capitalized
POS tags	unigram and bigram POS tags

parsing tree. By optimizing $\mathbf{P}(r, e_h, e_t)$ in Eq. 5, we also obtain the relation tuples for updated positive pairs E_p^{+n+1}.

$$E_p^+ = \underset{e_h}{\operatorname{argmax}}\, \mathbf{P}(p_{h,t}, e_h, e_t) \tag{11}$$

In example 3.5 and Fig. 3b, false tuple ⟨*city*, is capital of, *England*⟩ will be updated as ⟨*London*, is capital of, *England*⟩. Seeing *London* and *Paris* share lots of predicate and tail entities, the updated tuple is more cohesive with others *e.g.* ⟨*Paris*, become capital of, *France*⟩.

4 EXPERIMENTS

For thorough evaluation of the proposed approach, we test the performance of ReMine system from two aspects, *i.e.*, quality of the extracted entity phrases (*i.e.*, entity phrase extraction with distant training), and quality of the extracted relation tuples (*i.e.*, output of the Open IE system). In particular, we compare ReMine with state-of-the-art Open IE systems to validate our three claims: (1) the domain-independent framework performs consistently well on different domains, (2) global structure cohesiveness improves performance of Open IE, and (3) the proposed iterative updating algorithm is effective and scalable.

4.1 Experimental Setup

Datasets. We use three public datasets[2] from different domains in our experiments: (1) NYT [29]: a corpus consisting of 23.6k sentences from ~294k 1987-2007 New York Times news articles. 395 sentences are manually annotated with entities and their relationships. (2) Wiki-KBP [19]: The training corpus contains 2.4k sentences sampled from ~780k Wikipedia articles [19] as the training corpus and 290 manually annotated sentences as test data. (3) Twitter [38]: consists of 1.4 million tweets from Los Angeles with entities and/or noun phrases collected from 2014.08.01 to 2014.11.30.

Distant Supervision for Generating Training Data. For each corpus, we apply the entity linking tool DBpedia Spotlight [3] [9] to recognize DBpedia entities in sentences and use them as "seed" entity phrases. With seed entity phrases, we generate relation phrases between each pair of entity mentions via pattern matching (see Sec. 3.2), forming the seed relation tuples. These seed tuples are used as distant supervision for training segmentation algorithm (thus "distant training"). We then follow the procedure introduced in Sec. 3.1 to segment sentences into entity and relation phrases.

Phrase Features Generation. In order to estimate quality and catgeory of phrases, we use features $\mathcal{F}$ in Table 2. These features can be grouped into several different categories, *i.e.* statistic features, token-wise features and POS features. ReMine treats phrases with multiple POS tag sequences as different patterns. For example, "work NN" and "work VBP" are two different semantic patterns. We applied the Stanford CoreNLP [25] tool to get POS tags and dependency parsing trees.

Compared Methods. For the entity phrase extraction task, NYT and Wiki-KBP are used for evaluation, since both datasets contain annotated entity mentions in test set. We adopt the sequence labeling evaluation setup [24], and compare ReMine's entity phrase extraction module with two state-of-the-art sequence labeling methods and one distantly-supervised phrase mining method on the test sets: (1) **Ma & Hovy** [24]: adopts a Bi-directional LSTM-CNN structure to encode character embeddings and pre-trained word embeddings; (2) **Liu. *et al.*** [22]: incorporates a neural language model and conducts multi-task learning to guide sequence labeling; and (3) **AutoPhrase** [33]: the state-of-the-art quality phrase mining method with POS-guided phrasal segmentation.

For the relation tuple extraction task, we consider following Open IE baselines for comparison: (1) **OLLIE** [32] utilizes open pattern learning and extracts patterns over the dependency path and part-of-speech tags. (2) **ClausIE** [10] adopts clause patterns to handle long-distance relationships. (3) **Stanford OpenIE** [2] learns a clause splitter via distant training data. (4) **MinIE** [14] refines tuple extracted by ClausIE by identifying and removing parts that are considered overly specific. (5) **ReMine-L** is a base model of our approach with only local context. We only plot precision@300 in Fig. 4 as no ranking measure is deployed. (6) **ReMine-G** extend ReMine-L by ranking tuples via global cohesiveness without updating entity phrase pairs and any further iterations. (7) **ReMine** is our proposed approach, in which relation tuple generation module collaborates with global cohesiveness module.

Parameters Settings. For baselines of entity phrase extraction task, we tune all the models using the same validation set. In the testing of ReMine and its variants, we set hypermargin $\gamma = 1$, maximal phrase length $\epsilon = 6$, number of candidate subject entity phrase for each tail entity $M_{sp} = 6$ and learning rate of the global cohesiveness module $\alpha = 10^{-3}$. The dimension of global cohesiveness

[2]Codes and datasets can be downloaded at https://github.com/GentleZhu/ReMine

[3]https://github.com/dbpedia-spotlight/dbpedia-spotlight

representation k is 100. We stop further joint optimization if the ratio **t** of updated tuples is smaller than 10^{-3}.

Table 3: Performance comparison with state-of-the-art entity phrase extraction algorithms for the weakly-supervised entity phrase extraction task.

Methods	NYT [29]			Wiki-KBP [19]		
	F1	Prec	Rec	F1	Prec	Rec
AutoPhrase [33]	0.531	0.543	0.519	0.416	0.529	0.343
Ma & Hovy [24]	0.664	0.704	0.629	0.324	0.629	0.218
Liu. *et al.* [22]	**0.676**	**0.704**	0.650	0.337	0.629	0.230
ReMine	0.648	0.524	**0.849**	**0.515**	**0.636**	**0.432**

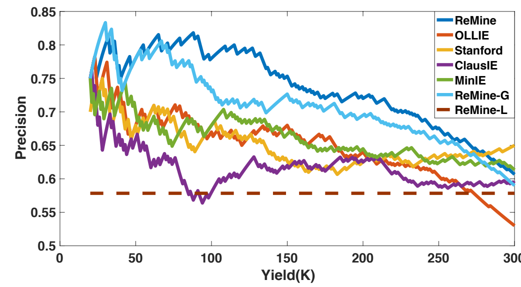
Cut-off Threshold for Extraction Output. The number of tuple extractions from different systems can vary a lot. For example, for the first 100 sentences in the NYT test set, both ReMine and OLLIE get about 300 tuples. In contrast, Stanford OpenIE returns more than 1,000 tuples. However, many paraphrased extractions can be found within them. Since each extracted tuple is also assigned with a confidence score, we select 300 tuples for both datasets with the highest scores for each open IE system to report the performance. By selecting 100 sentences from NYT test set and 300 tweets from Twitter test set, we believe ~3 tuples per sentence in News domain and ~1 tuple per sentence in Twitter are reasonable for a fair comparison. A more detailed study can be found in Sec. 5.

Annotation of Ground-truth Data. We manually labeled the top-300 tuple extraction results obtained from all compared methods via pooling method (*i.e.*, high-confidence tuples by each system are pooled together as the candidate set). Each extracted tuple in the candidate set was labeled by two independent annotators. A tuple is labeled as positive only if both labelers agree on its correctness. All tuples with conflicting labels results were filtered. Similar to [10], we ignored the context of the extracted tuples during labeling. For example, both (*"we", "hate", "it"*) and (*"he", "has", "father"*) will be treated as correct as long as they meet the fact described in the sentence. However, tuples cannot be read smoothly will be labeled as incorrect propositions. For example, (*"he", "is", "is the professor"*) and (*"he", "is", "the professor and"*) will not be counted since they have mistakes at the word segmentation level. The Cohen's Kappa value between the two labelers are 0.79 and 0.73 for the NYT dataset and the Twitter dataset respectively.

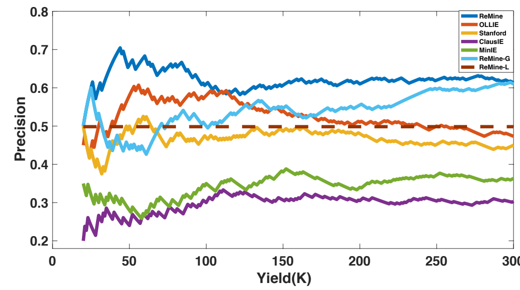
Evaluation Metrics. We use Precision (*i.e.* how many entities we get are correct), Recall (*i.e.* how many correct entities do we get), and F1-score to evaluate the performances on entity phrase extraction task, same as other sequence labeling studies [24]. For the Open IE task, since each tuple obtained by ReMine and other benchmark methods will also be assigned a confidence score. We rank all the tuples according to their confidence scores. Based on the ranking list, we use the following four measures: *P@k* is the precision at rank *k*. *MAP* is mean average precision of the whole ranking list. *NDCG@k* is the normalized discounted cumulative gain at rank k. *MRR* is the mean reciprocal rank of the whole ranking list. Note that we do not use recall in this task because it is impossible to collect all the "true" tuples.

4.2 Experiments and Performance Study

1. Performance on Entity Phrase Extraction. The training data is generated through distant supervision described above without type information. Regarding open domain extractions, we train



(a) NYT



(b) Twitter

Figure 4: The Precision@K curves of different open IE systems on NYT and Twitter datasets.

baseline models using the same distant supervision as ReMine, to push them towards a fair comparison. Table 3 demonstrates the comparison result over all datasets. In the Wiki-KBP dataset, ReMine evidently outperforms all the other baselines. In the NYT dataset, ReMine has a rather high recall and is on par with the two neural network models on F1-score.

2. Performance on Relation Tuple Extraction. On NYT and twitter test set, we compare ReMine with its variants ReMine-L and ReMine-G as well as four baseline open IE systems mentioned above. The results are shown in Figure 4 and Table 4.

"*Does* ReMine *perform consistently well on different domains?*"

According to the curves in Figure 4a and 4b, ReMine achieves the best performance among all open IE systems. All methods experience performance drop in Twitter, while ReMine declines less than any other methods on the rank-based measures. In the NYT dataset, all the systems except OLLIE have similar overall precision (*i.e. P*@300). But ReMine has a "higher" curve since most tuples obtained by Stanford OpenIE and ClausIE will be assigned score 1. Therefore we may not rank them in a very rational way. In contrast, the scores of different tuples obtained by ReMine-G and ReMine are usually distinct from each other. In Table 4, ReMine also consistently performs the best . In the Twitter dataset, ReMine shows its power in dealing with short and noisy text. Both ClausIE and MinIE have a rather low score since there are lots of non-standard language usages and grammatical errors in tweets. In twitter, dependency parsing attaches more wrong arguments and labels than usual. All methods investigated depend on dependency parsing to varying degrees, while clause-based methods rely heavily on it and may not achieve a satisfying performance.

"*Does global cohesiveness improve quality of open IE?*"

Model-wise, we believe global cohesiveness helps open IE from two aspects: (1) ranking tuples (2) updating entity phrase pairs.

Table 4: **Performance comparison with state-of-the-art Open IE systems on two datasets from different domains, using Precision@K, Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR).**

Methods	**NYT** [29]						**Twitter** [38]					
	P@100	P@200	MAP	NDCG@100	NDCG@200	MRR	P@100	P@200	MAP	NDCG@100	NDCG@200	MRR
ClausIE	0.580	0.625	0.623	0.575	0.667	0.019	0.300	0.305	0.308	0.332	0.545	0.021
Stanford	0.680	0.625	0.665	0.689	0.654	0.023	0.390	0.410	0.415	0.413	0.557	0.023
OLLIE	0.670	0.640	0.683	0.684	0.775	**0.028**	0.580	0.510	0.525	0.519	0.626	0.017
MinIE	0.680	0.645	0.687	0.724	0.723	0.027	0.350	0.340	0.361	0.362	0.541	**0.025**
ReMine-G	0.730	0.695	0.734	0.751	0.783	0.027	0.510	0.580	0.561	0.522	0.610	0.021
ReMine	**0.780**	**0.720**	**0.760**	**0.787**	**0.791**	0.027	**0.610**	**0.610**	**0.627**	**0.615**	**0.651**	0.022

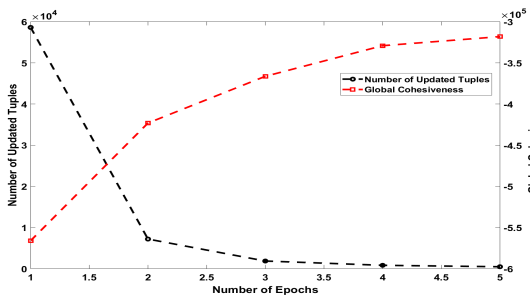


Figure 5: The number of updated tuples and global cohesiveness against the number of epochs for the proposed ReMine**.**

From Figure 4 and Table 4, we find ReMine outperforms ReMine-G and ReMine-L on each evaluation metric on both datasets. In particular, ReMine-G differs from ReMine-L only on extraction scores, since global cohesiveness σ provides better ranking performance (P@300) over random (ReMine-L). The gain between ReMine and ReMine-G clearly shows the updated extractions have better quality in general.

"*Is the joint optimization effective and scalable?*"

In Fig. 5, We plot out the number of updated tuples and global cohesiveness objective on NYT dataset. The number of updated tuples reflects how global cohesiveness influences the tuple genration module. The convergence of global cohesiveness indicates the joint optimization leads to cleaner and more coherent extractions. Suppose that corpus $\mathcal{D}$ has N_d words. The time cost of phrase extraction module is $O(\epsilon N_d)$ with the assumption that maximal length of a phrase is a constant ϵ. The tuple generation module examines M_{sp} candidate head entities for each entity phrase and takes ϵM_{sp} time to perform tuple generation as maximal semantic path is bounded by M_{sp}. In total, it takes $O(\epsilon M_{sp}^2 N_d)$ time. The global cohesiveness module requires $O(N_r k + N_e k)$, where N_r,N_e are number of entity and relation phrases and k is the embedding dimension. N_r and N_e is bounded by N_d. By omitting constants, the computational complexity of joint optimization is $O(N_d)$. Furthermore, each component of ReMine is paralleled as the independence between each document.

5 CASE STUDY

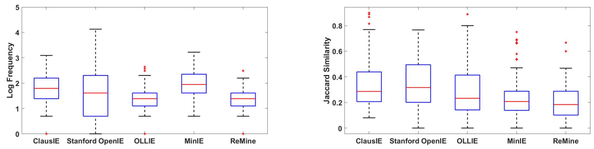
Clearness and correctness on extractions. In Table. 5, we show the extraction samples of the NYT sentence *"Gov. Tim Pawlenty of Minnesota ordered the state health department this month to monitor day-to-day operations at the Minneapolis Veterans Home after state inspectors found that three men had died there in the previous month because of neglect or medical errors."*. We can see that

Table 5: Extraction samples of one sentence in the NYT dataset using different methods. "T" means correct tuples and "F" means incorrect ones. *The tuple is too complicated to clearly explain one proposition. #The tuple cannot read smoothly. †The tuple is logically wrong.

	ClausIE	
R_1	*("Gov. Tim Pawlenty of Minnesota", "ordered", "the state health department this month to monitor day-to-day operations after state inspectors found that three men had died there in the previous month because of neglect or medical errors")*	F*
R_2	*("Gov. Tim Pawlenty of Minnesota", "ordered", "the state health department this month to monitor day-to-day operations")*	T
	Stanford OpenIE	
R_3	*("Gov. Tim Pawlenty", "ordered", "state health department")*	T
R_4	*("Gov. Tim Pawlenty", "monitor", "operations")*	F†
R_5	*("three men", "died there because of", "neglect")*	T
R_6	*("men", "died in", "month")*	F#
	OLLIE	
R_7	*("Gov. Tim Pawlenty of Minnesota", "ordered the state health department in", "this month")*	T
R_8	*("three men", "had died there in", "the previous month")*	T
R_9	*("Gov. Tim Pawlenty of Minnesota", "had died because of", "neglect errors")*	F†
	MinIE	
R_{10}	*("Tim Pawlenty", "is", "Gov.")*	T
R_{11}	*("Tim Pawlenty of Minnesota", "ordered state health department", "this month")*	T
R_{12}	*("QUANT_S_1 men", "had died because of", "neglect errors")*	F†
	ReMine	
R_{13}	*("Gov. Tim Pawlenty of Minnesota", "order", "the state health department")*	T
R_{14}	*("Gov. Tim Pawlenty of Minnesota", "order_to_monitor", "day-to-day operation")*	T
R_{15}	*("Gov. Tim Pawlenty of Minnesota", "order_to_monitor_at", "Minneapolis Veterans Home")*	T
R_{16}	*("three man", "have_die_there", "medical error")*	F#

all the extractors share consensus on that *"Gov. Tim Pawlenty of Minnesota ordered the state health department"* (R_2, R_3, R_7, R_{11} and R_{13}). But some other actions do not belong to "Tim Pawlenty". Both Stanford OpenIE and OLLIE make mistakes on that (R_4 and R_9). In contrast, ClausIE has no logical mistakes in the samples. However, the objective component of R_1 is too complicated to illustrate one proposition clearly. As we mentioned above, these kinds of tuples will be labeled as incorrect ones. R_{15} is the only correct tuple to identify the location *"Minneapolis Veterans Home"*, and ReMine also carefully selects the words to form the predicate *"order_to_monitor_at"* to prevent excessively long relation phrase.

Distinctiveness of tuple generation. In our formulation, we try to cover every entity detected in the target sentence while avoiding extracting duplicate tuples. In Fig. 6a, we show the distribution



(a) Number of tuples **(b) Jaccard similarity**

Figure 6: Distribution over number of extractions and distinctiveness of extractions for different Open IE systems.

of the number of extractions obtained by each Open IE system on the first 100 sentences in NYT dataset. We can see that OLLIE's and ReMine's distributions are relatively balanced. In contrast, Stanford OpenIE returns extractions with a large variance. Among 1054 tuples it extracted, there are 228 tuples belonging to a single sentence and 157 belonging to another. This is despite the latter sentence only containing 39 words. This reminds us that the number of extractions may not be a good alternative to "recall". A more direct way to examine distinctiveness is calculating average Jaccard similarity between extractions from same sentence. We present the Jaccard similarity distribution of different systems at Fig. 6b, we can clearly see MinIE and ReMine extract the most distinctive facts.

Effectiveness of global evidence. Corpus-level cohesiveness can help reduce local error while generating relation tuples. Especially on the twitter dataset, local linguistic structure fails to attach correct argument initially whereby global cohesiveness module corrects those extractions. In table 6, ReMine rejects entity pair (*Liberador, Hollywood*) which is not compatible with the predicate "@". This is because in the twitter corpus, it is more common to see *Person @ Place*. Therefore ReMine attaches Hollywood to Dudamel.

Table 6: Different entity pairs discovered by ReMine **and** ReMine-G**, where blue ones are incorrect extractions.**

Dudamel conduct his score from Liberador#BeastMode @Hollywood Bowl

ReMine-G	ReMine
(Dudamel; "conduct"; Liberador)	*(Dudamel; "conduct"; Liberador)*
(Dudamel; "conduct...from"; #BeastMode)	*(Dudamel; "conduct... @"; Hollywood Bowl)*
(Liberador, "@", Hollywood Bowl)	

6 RELATED WORK

Open Information Extraction. Open domain information extraction has been extensively studied in literature. Most of the existing work follow two lines of work, that is, pattern based methods or clause based methods. Pattern based information extraction can be as early as Hearst patterns like "NP_0 such as $\{NP_1, NP_2, ...\}$" for hyponymy relation extraction [16]. Carlson and Mitchell *et al.* introduced Never-Ending Language Learning (NELL) based on free-text predicate patterns [7, 26]. ReVerb [12] identified relational phrases via part-of-speech-based regular expressions. Besides part-of-speech tags, recent works have started to use more linguistic features, such as dependency parsing, to induce long distance relationships [27, 32]. Similarly, ClausIE [10] inducted short but coherent pieces of information along dependency paths, which is typically subject, predicate and optional object with complement. Angeli *et al.* adopts a clause splitter using distant training and statistically maps predicate to known relation schemas [2]. MinIE [14] removes overly-specific constituents and captures implicit relations in ClausIE by introducing several statistical measures like polarity, modality, attribution, and quantities. Compared with these works, this paper differs in several aspects: (1) previous work relies on external tools for phrase extraction, which may suffer from domain-shift and sparsity problem, while we provide an End-to-End solution towards Open IE. (2) Although previous efforts achieve comparable high precision and reasonable coverage on extraction results, they all focus on local linguistic context. The correctness of extracted facts are evaluated purely on local context, however, large corpus can exclude false extractions from inferred inconsistencies.

Knowledge Base Embedding and Completion. Knowledge bases (KBs), such as DBpedia [4] and Freebase [17], extract tuples from World Wide Web. Knowledge base population or completion aims at predicting whether tuples not in knowledge base are likely to be true or not. Previous works attempted to construct web-scale knowledge base using statistical learning and pre-defined rules and predicates [28]. Recently, embedding models [5, 18, 30, 34] have been widely used to learn semantic representation for both entities and relations. By observing each relation may have different semantic meaning, Wang *et al.* [37] projected entity vectors to relation-specific hyperplane. Further research [15, 23] shows that embedding techniques can support composite query(*i.e.* asking about multiple relations) on knowledge graph. All previouos knowledge graph embedding methods start with existing knowledge base tuples, while our proposed global cohesiveness representation starts from noisy extractions. There is another line of work trying to combining KB relations and textual relations [36] or model unstructured and structured data by universal schema [29]. However, they are all built upon on existing and specific relation types. Although we shared similar semantic measures as these work, ReMine uses KB embeddings to measure quality of extracted relation tuples and improve Open IE in a multi-tasking way.

7 CONCLUSION

This paper studies the task of open information extraction and proposes a principled framework, ReMine, to unify local contextual information and global structural cohesiveness for effective extraction of relation tuples. The local objective is jointly learned together with a translating-based objective to enforce structural cohesiveness, such that corpus-level statistics are incorporated for boosting high-quality tuples extracted from individual sentences. Experiments on two real-world corpora of different domains demonstrate that ReMine system achieves superior precision when outputting same number of extractions, compared with several state-of-the-art open IE systems. Interesting future work can be (1) On-The-Fly knowledge graph construction from relation tuples; (2) applying ReMine to downstream applications *e.g.* open domain Question Answering.

8 ACKNOWLEDGEMENTS

Research was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), National Science Foundation IIS 16-18481, IIS 17-04532, and IIS-17-41317, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov). Xiang Ren's research has been supported in part by National Science Foundation SMA 18-29268. We thank Frank F. Xu and Ellen Wu for valuable feedback and discussions.

REFERENCES

[1] A. Allahverdyan and A. Galstyan. Comparative analysis of viterbi training and maximum likelihood estimation for hmms. In *NIPS*, 2011.

[2] G. Angeli, M. J. Premkumar, and C. D. Manning. Leveraging linguistic structure for open domain information extraction. In *ACL*, 2015.

[3] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, 2007.

[4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, 2008.

[5] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013.

[6] R. C. Bunescu and R. J. Mooney. Learning to extract relations from the web using minimal supervision. In *ACL*, 2007.

[7] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, 2010.

[8] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka Jr, and T. M. Mitchell. Coupled semi-supervised learning for information extraction. In *WSDM*, 2010.

[9] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *I-Semantics*, 2013.

[10] L. Del Corro and R. Gemulla. Clausie: clause-based open information extraction. In *WWW*, 2013.

[11] X. L. Dong, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *KDD*, 2014.

[12] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *EMNLP*, 2011.

[13] A. Fader, L. Zettlemoyer, and O. Etzioni. Open question answering over curated and extracted knowledge bases. *KDD*, 2014.

[14] K. Gashteovski, R. Gemulla, and L. Del Corro. Minie: minimizing facts in open information extraction. In *EMNLP*, 2017.

[15] K. Guu, J. Miller, and P. Liang. Traversing knowledge graphs in vector space. In *EMNLP*, 2015.

[16] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *ACL*, 1992.

[17] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[18] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 15, pages 2181–2187, 2015.

[19] X. Ling and D. S. Weld. Fine-grained entity recognition. In *AAAI*, 2012.

[20] J. Liu, J. Shang, and J. Han. Phrase mining from massive text and its applications. volume 9.

[21] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han. Mining quality phrases from massive text corpora. In *SIGMOD*, 2015.

[22] L. Liu, J. Shang, F. F. Xu, X. Ren, H. Gui, J. Peng, and J. Han. Empower sequence labeling with task-aware neural language model. *arXiv preprint arXiv:1709.04109*, 2017.

[23] Y. Luo, Q. Wang, B. Wang, and L. Guo. Context-dependent knowledge graph embedding. In *EMNLP*, 2015.

[24] X. Ma and E. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*, 2016.

[25] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, 2014.

[26] T. M. Mitchell, W. W. Cohen, E. R. Hruschka Jr, P. P. Talukdar, J. Betteridge, A. Carlson, B. D. Mishra, M. Gardner, B. Kisiel, J. Krishnamurthy, et al. Never ending learning. In *AAAI*, 2015.

[27] N. Nakashole, G. Weikum, and F. Suchanek. Patty: A taxonomy of relational patterns with semantic types. In *EMNLP-CoNLL*, 2012.

[28] F. Niu, C. Zhang, C. Ré, and J. W. Shavlik. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. In *VLDS*, 2012.

[29] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin. Relation extraction with matrix factorization and universal schemas. In *NAACL*, 2013.

[30] T. Rocktäschel, S. Singh, and S. Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1119–1129, 2015.

[31] S. Sarawagi. Information extraction. *Foundations and trends in databases*, 1(3):261–377, 2008.

[32] M. Schmitz, R. Bart, S. Soderland, O. Etzioni, et al. Open language learning for information extraction. In *EMNLP-CoNLL*, 2012.

[33] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss, and J. Han. Automated phrase mining from massive text corpora. *arXiv preprint arXiv:1702.04457*, 2017.

[34] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013.

[35] H. Sun, H. Ma, W.-t. Yih, C.-T. Tsai, J. Liu, and M.-W. Chang. Open domain question answering via semantic enrichment. In *WWW*, 2015.

[36] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, 2015.

[37] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 2014.

[38] C. Zhang, G. Zhou, Q. Yuan, H. Zhuang, Y. Zheng, L. Kaplan, S. Wang, and J. Han. Geoburst: Real-time local event detection in geo-tagged tweet streams. In *SIGIR*, 2016.