INTEGRATING LOCAL CONTEXT AND GLOBAL COHESIVENESS FOR OPEN INFORMATION EXTRACTION

BY

QI ZHU

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Professor Jiawei Han

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, and do not confine to a pre-defined 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 signal and global structural signal in a unified framework with distant supervision. The new system can be efficiently applied to different domains as it uses facts from external knowledge bases as supervision; and can effectively score sentence-level tuple extractions based on corpus-level statistics. Specifically, we design a joint optimization problem to unify (1) segmenting entity/relation phrases in individual sentences based on local context; and (2) measuring the quality of sentence-level extractions with a translating-based objective. Experiments on two real-world corpora from different domains demonstrate the effectiveness and robustness of ReMine when compared to other open IE systems.

To my parents, for their love and support.

ACKNOWLEDGMENTS

I would love to express my great appreciation towards my advisor Professor Jiawei Han of the Department of Computer Science at University of Illinois at Urbana-Champaign. During my two-year master study, he was so kind and patient to guide me in research and live. Being a freshman in research field, I was illuminated and inspired by Professor Han and other data mining group members. He provided me teaching and research assistantship in the past four semesters, so that I could be fully dedicated into my research. During my thesis work, a lot of people in our group helped me a lot, especially Xiang Ren, Yu Zhang and Jingbo Shang.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Massive text corpora are emerging worldwide in different domains and languages. The sheer size of such unstructured data and the rapid growth of new data pose grand challenges on making sense of these massive corpora. Information extraction (IE) [1] – 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. 1.1, for example, the relation tuple (*Louvre-Lens*, build, *new satellites*) can be extracted from sentence S_2 to represent a piece of factual knowledge in text with structured form. Relation tuples so extracted have a variety of downstream applications, including serving as building blocks for knowledge base construction [2] and facilitating question answering systems [3, 4]. While traditional IE systems require people to pre-specify the set of relations of interests, recent studies on *open-domain information extraction* (Open IE) [5, 6, 7] 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).

Prior work can be summarized as sharing two common characteristics: (1) conducting extraction based on local context information; and (2) adopting an incremental system pipeline. Current Open IE systems focus on analyzing the local context within individual sentences to extract entity 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. 1.1, seeing entity phrases "*London*" and "*Paris*" frequently co-occur with similar relation phrase and tail entities in the corpus, one gets to know that they have close semantics (same for "*Great Britain*" and "*France*"). On one hand, this helps confirm that (*Paris*, is in, *France*) is a quality tuple if knowing (*London*, is in , *Great Britain*) is a good tuple. On the other, this helps rule out the tuple (*Paris*, build, *new satellites*) as "*Louvre-Lens*" is semantically distant from "*Paris*". Therefore, the rich information redundancy in the massive corpus motivates us to design an effective way of measuring whether a candidate relation tuple is consistently used across various context (*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 extracted entity phrases from sentences, which are then used to form entity pairs for relation tuple extraction [5, 6, 7]. Some systems further rely on dependency parsers to generate syntax parse tree for guiding the relation tuple extraction [7, 8, 9]. However, these systems suffer from *error propagation* as the errors in prior parts of the pipeline could accumulate cascading down the pipeline, yielding more significant errors. In addition, the NERs and NP chunkers

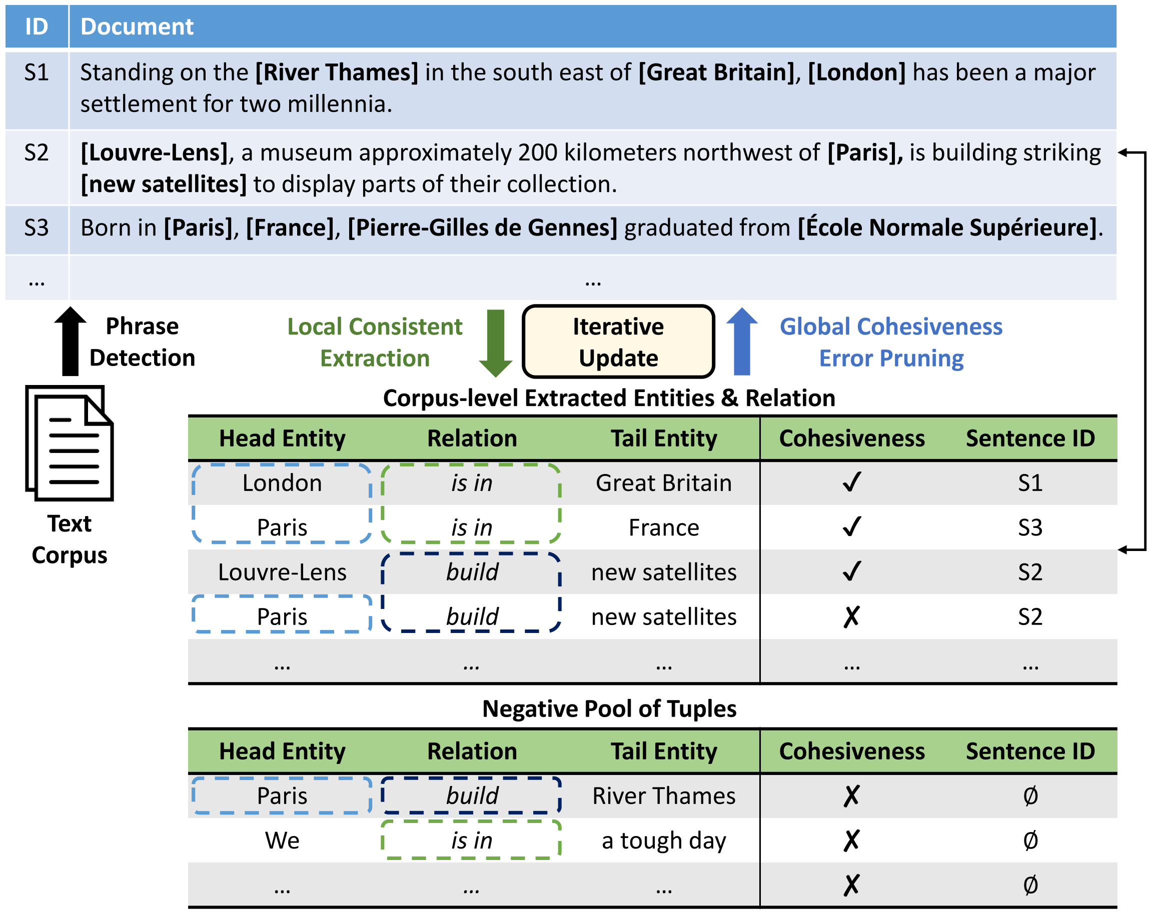


Figure 1.1: Example of incorporating global cohesiveness view for error pruning. "London" and "Paris" are similar because they are head entities of the same relation "is in". When it comes to the relation "build", since "London" and "build" do not co-occur in any tuple in the corpus, then it is unlikely for tuples with "Paris" and "build" to be correct.

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 Open IE problem, *i.e.*, the local context information and the global cohesiveness (see also Fig. 1.2). While most existing Open IE systems focus on analyzing local context and linguistic structures for tuple extraction, ReMine further make 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 embeddings space, where two entity phrases are similar if they share similar relation phrases and entity arguments. 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

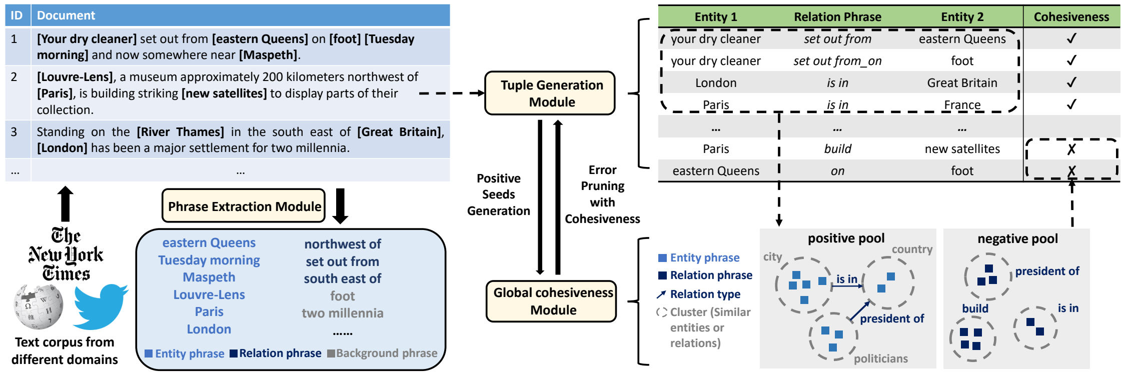


Figure 1.2: Overview of the ReMine Framework.

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.

Specifically, ReMine first identifies entity and relation phrases from local context. In Fig. 1.2, suppose we have a sentence "Your dry cleaner set out from eastern Queens on foot Tuesday morning and now somewhere near Maspeth.". We will first extract three entity phrases, *eastern Queens*, *Tuesday morning*, *Maspeth*, as well as two background phrases *Your dry cleaner*, *foot*. Then, ReMine jointly mines relation tuples and measure extraction with global translating objective. Local consistent text segmentation may generate noisy tuples, such as <your dry cleaner, *set out from*, eastern Queens> and <eastern Queens, *on*, foot>. However, from the global cohesiveness view, we may infer the second tuple as a false positive. Entity phrases like "eastern Queens" are seldom linked by relation phrase *"on"* in extracted tuples. Overall, ReMine will iteratively refine extracted tuples and learn entity and relation representations from corpus level. With careful attention to advantages of linguistic patterns [11, 12] and representation learning [13], this approach benefits from both side. Compared to previous open IE systems, ReMine prune extracted tuples via global cohesiveness and its accuracy is not sensitive to the target domain.

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 can identify high quality phrases of multiple types.
3. We design a unified objective to measure both tuple quality in 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 recognition task as well as Open IE task, when compared with various baseline methods.

CHAPTER 2: RELATED WORK

Information Extraction. Open domain information extraction has been extensively studied in literature. Most of the existing work follows 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 [11]. Carlson and Mitchell *et al.* introduced Never-Ending Language Learning (NELL) based on free-text predicate patterns [14, 15]. ReVerb [12] identified relational phrases via part-of-speech-based regular expressions. Besides part-of-speech tags, recent works start to use more linguistic features, like dependency parsing, to induct long distance relationships [16, 7]. Similarly, ClausIE [8] inducted short but coherent pieces of information along dependency paths, which is typically subject, predicate and optional object with complement. Angeli *et al.* adopt a clause splitter using distant training and mapped predicates to a known relation schema statistically [9]. MinIE [17] further improve the clearness of relation tuples by introducing different 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 on target domain. (2) Although previous efforts achieves 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 inconsistence inferred.

Knowledge Base Population. Knowledge bases (KBs), such as DBpedia [18] and Freebase [19], extract tuples from World Wide Web. However, they are all built upon on existing and specific relation types. Knowledge base population or completion aims at predicting whether tuples not in knowledge base are likely to be true or not. Embedding models [13] has been widely used to learn semantic representation for both entities and relations. By observing each relation may have different semantic meaning, Wang *et al.* [20] project entity vectors to relation-specific hyperplane. Recent research [21, 22] shows that relation path is traversable and contains richer information. People also try to construct web-scale knowledge base using statistical learning and pre-defined rules and predicates [23]. All these approaches start with clean knowledge base tuples, our proposed start from noisy extractions but share similar semantic measures as them. On the opposite, we output comparable clean relation tuples rather than taking tuples as input.

CHAPTER 3: NOTATIONS AND PROBLEM DEFINITION

In general, Open IE is viewed as a two-stage extraction task, where the goal is to first identify entity phrases $\mathcal{E}$, relation phrases $\mathcal{R}$, then select and pair up entity phrases into entity argument pairs, and further extract meaningful relation tuples $\mathcal{T}$ among them.

Entity Phrases. *For any sentence s in a corpus $\mathcal{D}$, entity phrases (e_1, e_2, ..., e_n) $\subset \mathcal{E}$ are defined as token spans in s. An entity phrase usually appears multiple times in corpus,* e.g. *in Fig. 1.2, all strings in blue are entity phrases*

In Open IE, entity phrases are usually not only name entities in pre-defined types, *e.g. time, location, person, organization.* There are phrases that correspond to less known facts, such as the major effort and the final group, which contain information in specific sentence whereas less important in corpus. In Fig. 1.2, we have the sentence “Your dry cleaner set out from eastern Queens on foot Tuesday morning and is now somewhere near Maspeth.” “Your dry cleaner” is not a named entity. However, it is the subject of this sentence and cannot be omitted in relation tuples extraction. Therefore, previous work [12, 7] usually use NP-chunking to identify entity phrases.

Relation Phrases. *Each relation phrase r conveys semantic correlations between entity arguments. In Fig. 1.2, strings in navy blue are relation phrases. Given entity phrases (e_1, e_2, ..., e_n) in sentence s, relation phrases (r_1, r_2, ..., r_n) $\subset \mathcal{R}$ are extracted between specific entity pairs (e_h, e_t).*

Relation phrases are not exactly relation instances. Specifically, one relation instance can correspond to multiple relation phrases, i.e. *location/country/capital can correspond to ('s capital, capital of, the capital, ...).* More often, relation phrases do not have any corresponding relation instances in knowledge base, like verb phrases. NP (noun phrase) chunking and name entity recognition have been extensively studied by NLP community in the past decades. To the best of our knowledge, we are among the first attempts to extract entity, relation simultaneously under distant supervision.

Relation Tuple. *Relation tuples $\mathcal{T}$ are defined as* $\{e_h, e_t, p\}$*, 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.*

Different from existing Open Information Extraction systems [9, 7], we treat predicate as combination of relation phrase, *e.g.* {Your dry cleaner, foot, (*set_out, on*)}. Unlike the relation types used by external knowledge bases, predicate in relation tuples can be ambiguous and may not align with relation types. We will discuss how to capture semantic drift in one predicate and identify relation phrases $(r_1, r_2, ...r_n)$ in our formulation. Formally, we define the task of Open IE as follows.

(Problem Definition) *Given a corpus $\mathcal{D}$, the task of Open IE aims to extract entity phrases $\mathcal{E}$, relation phrases $\mathcal{R}$ and relation tuples $\{e_h, e_t, p\}_{k=1}^{N_t}$, where entity argument pairs (e_h, e_t) extracted from one sentence are distinctive to each other.*

CHAPTER 4: THE REMINE FRAMEWORK

ReMine aims to jointly address two problems, that is, extracting entity and relation phrases from sentences and generating quality relation tuples. There are three challenges. First, distant supervision may contain *false* seed examples of entity and relation phrases, and thus asks for effective measuring of the quality score for phrase candidates. Second, there exist multiple entity phrases in one sentence. Therefore, selecting entities to form relation tuples may suffer from ambiguity in local context. Third, ranking extracted tuples without referring to the entire corpus may favor with good local structures.

Framework Overview. We proposed a framework, called ReMine, that integrates both local context and global structure cohensiveness (see also Fig. 1.2) to address above challenges. There are three major modules in ReMine: (1) phrase extraction module; (2) relation tuple generation; and (3) global cohesiveness module. To overcome sparse and noisy labels, phrase extraction module trains a robust phrase classifier and adjusts quality from a generative perspective. The relation tuple extraction module generates tuples from sentence structure, which adopts widely used local structure patterns [8, 16, 7], including syntactic and lexically patterns over pos tags and dependency parsing tree. However different from previous studies, the module tries to benefit from information redundancy and mine distinctive extractions with accurate relation phrases. Meanwhile, global cohesiveness module learns entity and relation phrases representation with a score function to rank tuples. Relation tuple generation module and global cohesiveness module are collaborating with each other. Particularly, relation tuple generation module produces coarse positive tuple seeds and feeds them into global cohesiveness module. By distinguishing positive tuples with constructed negative samples, global cohesiveness module provide a cohesiveness measure for tuple generation. Tuple generator further incorporates global cohesiveness into local generation and outputs more precise extractions. ReMine intergrates tuple generation and global cohesiveness learning into a joint optimization framework. They iteratively refine input for each other and eventually obtain clean extractions. Once the training process converges, the tuples are expected to be distinctive and accurate. Overall, ReMine extracts relation tuples as follows, see also Alg. 4.1:

1. **Phrase extraction module** conducts context-dependent phrasal segmentation on target corpus (using distant supervision) , to generate entity phrases $\mathcal{E}$, relation phrases $\mathcal{R}$ and word sequence probability $\mathcal{A}$.
2. **Relation tuple generation module** generates positive entity pairs and identifies predicate p between entity argument pair via tuple generative process.

3. **Global cohesiveness module** learns entity and relation embeddings $\mathcal{V}$ via a translating objective to capture global structure cohesiveness $\mathcal{W}$.
4. **Update sentence-level extractions** $\mathcal{T}$ based on both local context information and global structure cohesiveness.

4.1 PHRASE EXTRACTION MODULE

We explore entity and relation phrase extraction as a multiple type phrasal segmentation task, traditional Open IE use NP-chunking to extract entity phrases, yet not all noun phrases can carry rich information and it requires additional training. Our method uses context-dependent phrasal segmentation to detect and evaluate whether a token span more likely to be an entity phrase, relation phrase or background phrase. Given word sequence $\mathcal{C}$ and corresponding linguistic features $\mathcal{F}$, a 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 $t_i \in \{entity, relation, background\}$, indicating the most possible type of s_i, the joint probability is factorized as:

$$P(\mathcal{S}, \mathcal{C}, \mathcal{F}) = \prod_{t=1}^{n} P(b_{t+1}, w_{[b_t, b_{t+1})} | b_t, \mathcal{F}) \tag{4.1}$$

ReMine generates each segment as follows,
1. Given the start index b_i, generate the end index b_{i+1} according to context-dependent prior Δ, *i.e.* dependency tree pattern prior.

$$P(b_{i+1} | b_i, \mathcal{F}) = \Delta(\mathcal{F}_{[b_i, b_{i+1})]}) \tag{4.2}$$

2. Given the start and end index (b_i, b_{i+1}) of segment s_i, generate a word sequence $w_{[b_i, b_{i+1})}$ according to a multinomial distribution over all segments at the same length.

$$P(w_{[b_i, b_{i+1})} | b_i, b_{i+1}) = P(w_{[b_i, b_{i+1})} | b_{i+1} - b_i) \tag{4.3}$$

3. Finally, we generate a phrase type t_i indicating that category $w_{[b_i, b_{i+1})}$ most likely belongs to and a quality score showing how it likely to be a good phrase $\lceil w \rfloor$.

$$P(\lceil w_{[b_i, b_{i+1})} \rfloor | w_{[b_i, b_{i+1})}) = \max_{t_i} P(t_i | w_{[b_i, b_{i+1})}) \tag{4.4}$$

Candidate Generation. Phrase Mining [24] had made an assumption that quality phrases

Table 4.1: Entity and relation phrase candidates generation with regular expression patterns on part-of-speech tag

<table>
<tr><th>Pattern</th><th>Examples</th></tr>
<tr><td colspan="2">Entity Phrase Patterns</td></tr>
<tr><td><DT|PP$>?<JJ>*<NN>+</td><td>the state health department</td></tr>
<tr><td><NNP>+<IN>?<NNP>+</td><td>Gov. Tim Pawlenty of Minnesota</td></tr>
<tr><td colspan="2">Relation Phrase Patterns</td></tr>
<tr><td>{V=<VB|VBD|VBG|VBN|VBN|VBP|VBZ>+}</td><td>furnish, work, leave</td></tr>
<tr><td>{V}{P=<NN|JJ|RP|PRP|DT>}</td><td>provided by, retire from</td></tr>
<tr><td>{V}{W=<IN|RP>?*}{P}</td><td>die peacefully at home in</td></tr>
</table>

are frequent n-grams in corpus, while it is not the case when sentence-level extractions are important. To overcome phrase sparsity, several NP chunking rules, see Table 4.1, are adopted to discover infrequent but informative phrase candidates. In our experiments, frequent n-grams and NP chunking rules contribute comparable amount of phrase candidates. These rules have been proved lead to high recall, see Table. 5.2, compared with those only use frequent n-grams [25, 26]

We denote $P(b_{i+1}|b_i, \mathcal{F})$ as $\Delta\mathcal{F}_{[b_i,b_{i+1})]}$, $P(w_{[b_i,b_{i+1})}|b_{i+1}-b_i)$ as θ_u and $\max_{t_i} P(t_i|w_{[b_i,b_{i+1})})$ as $Q(w_{[b_i,b_{i+1})})$. Now we will show how we use Viterbi Training [27] to update Segmentation S and parameters θ, Δ iteratively. In the E-step, given θ and Δ, dynamic programming is used to find the optimized segmentation. Given start index i and end index j,

$$\mathbf{H}_j = \max(\mathbf{H}_j, \mathbf{H}_i \cdot p(i, w_{[i,j)}|j, \mathcal{F})) \tag{4.5}$$

where $\mathbf{H}_i$ the current maximum generation probability ends at i.
In the M-step, we first fixed parameter θ, and update context-dependent prior, $f_{dep} \in \mathbb{N}$ denotes tree pattern id:

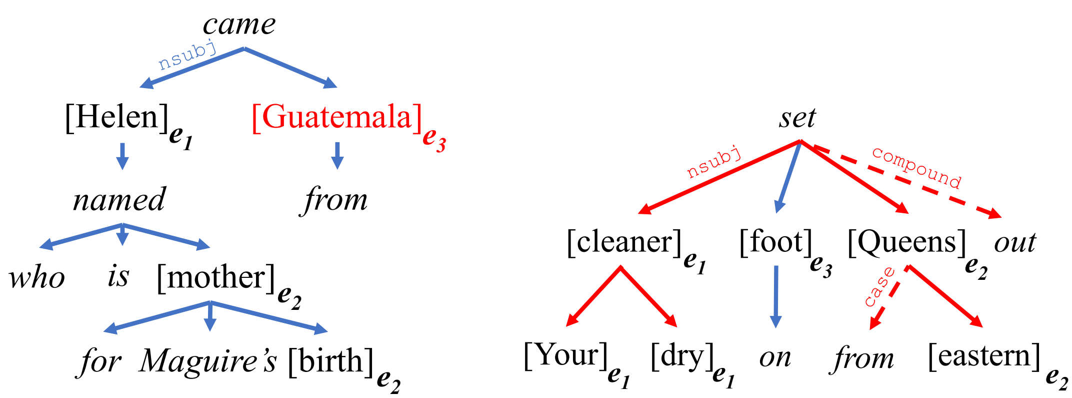
$$\Delta(f_{dep}) = \frac{\sum_{i=1}^{m} \mathbf{1} \cdot (\mathcal{F}_{[b_i,b_{i+1})} = f_{dep})}{\sum_{l=2}^{max_l} \sum_{i=1}^{n-l} \mathcal{F}_{[i,i+l)]}} \tag{4.6}$$

Next when Δ is fixed, optimized solution of θ_u is:

$$\theta_u = \frac{\sum_{i=1}^{m} \mathbf{1} \cdot (w_{[b_i,b_{i+1})} = u)}{\sum_{i=1}^{m} \mathbf{1} \cdot (b_{i+1} - b_i = |u|)} \tag{4.7}$$

4.2 TUPLE GENERATION MODULE

Leveraging information along the dependency path between two given entities has been proved useful for open information extraction [28, 29], as it reduces noise by removing irrelevant semantic phrases or clauses in long sentences with multiple entities. Similar to the goal



(a) Selecting among candidate subject entities **(b)** Finding shortest dependency path

Figure 4.1: Dependency parsing tree of example sentence in Fig. 1.2, *"Your dry cleaner set out from eastern Queens on foot Tuesday morning."* Segmented entities are marked as "[entity_token]$_{e_i}$"

of sentence compression or simplification [30, 31], we assume that the information along the proposed semantic path is sufficient for relation tuple mining.

(Semantic Path) *For any two given entity arguments (head, tail) in the same sentence, semantic path is defined as expanded dependency path* $P_{h,t} = \{p_1, p_2, ..., p_n\}$ *between token span* **h** *and* **t**.

In the previous section, we have introduced how argument candidates are extracted in ReMine. Noticing relation tuples are not linearly aligned, we now present how we generate valuable relation tuples based on those phrases along semantic path, *i.e.*

$$\mathbf{P}(r, e_i, e_j) = \prod_{t=1}^{n} \mathbf{P}(r_{[b_t, b_{t+1})} | w_{[b_t, b_{t+1})}, e_i, e_j) \mathbf{P}(w_{[b_t, b_{t+1})} | b_t, b_t + 1) \tag{4.8}$$

where $b_1, b_2, ..., b_n + 1$ are boundary index between token span (e_i, e_j). Generation Module first generates word sequence according to multinomial distribution inherited from phrase extraction module, then ReMine generates wether it is a good relation between entity i and entity j.

Positive Entity Pairs Initialization. For a given sentence s, after phrase segmentation, we have entity arguments $e_1, e_2, ..., e_n$ and relation arguments $r_1, r_2, ..., r_n$. Notice that good background phrases also recognized as arguments. However, it's infeasible to explore semantic path between every entity pair and a large portion of tuples are incorrect among $N(N-1)$ pairs. Positive entity pairs E_p^+ are entity arguments pair selected. Here we heuristically initialize E_p^{+0} by attaching *nearest* subject e_i to object e_j and make an approximation that each entity argument phrase can only be object once, which also guarantees entity pairs to be distinctive. Nearest subject of e_j is defined as entity e_i that has the shortest dependency path length to e_j among all other entities. Considering Fig. 4.1a, we would like to find subject of entity e_3 : *Guatemala*, length of the shortest path between e_3 and e_1, e_2 are 2,4 respectively. For those entity candidates with the same distance, see Fig. 4.1b, both e_1: *Your dry cleaner* and e_2: *eastern Queens* is one hop away from e_2: *foot*. We will prefer subject with "nsubj" type *i.e.* e_1 then choose closest entity in original sentence if there are still multiple of them.

Semantic Path Generation. Once E_p^+ is determined, the semantic path is therefore along the shortest dependency path between two arguments. For example, in Fig. 4.1b, the semantic path between e_1 and e_4 is marked in red. To preserve integrity of potential relation phrases, particles and preposition along semantic path are added as red dotted line in Fig. 4.1b.

4.3 GLOBAL COHESIVENESS MODULE

During the process of tuple generation, $\mathbf{P}(r_{[b_i,b_{i+1})}|w_{[b_i,b_{i+1})}, e_i, e_j)$ can be seen as cohesiveness measure $\mathcal{S}$ of candidate relation tuple (e_i, r, e_j). In our illustrative example in Fig. 1.1, current methods use textual patterns [7, 8] to identify (Paris, build, new satellites) as a false extraction, while we prune it via global cohesiveness measure. To capture the global cohesiveness of relation tuples, we adopt translating measuring of knowledge base completion [13].

$$S(h, l, t) = \|v_h + v_l - v_t\|; v \in \mathbb{R} \tag{4.9}$$

where v_h, v_t are dense vectors for head and tail entities, l is the predicate. We use L_1 norm in ReMine for efficiency.

Such objective associates entity and relation with dense feature vectors, which can be further applied on relation classification and KB completion. Based on initial positive entity pairs constructed E_p^{+0} and relation tuples, we construct a pseudo knowledge graph. Particularly, predicate l is composed by relation phrases extracted along semantic path $l = (r_1, r_2, ..., r_n)$. We model multiple relation phrases in one predicate as process of knowledge traverse [21] *i.e.* $v_l = \sum_{i=1}^{n} v_{r_i}/n$.

Global Cohesiveness Error Pruning and Learning Objective. In order to learn global cohesiveness representation $\mathcal{V}$ and $\mathcal{W}$, Relation Tuple Generation Module acts as a positive seeds generator. Global Cohesiveness Module will construct correlated negative tuples from positive seeds accordingly, see Fig. 1.2. False tuples like (Paris, build, new satellites) will be reduced by some similar negative tuples like (Paris, build, River Thames). Cohesiveness measure S is optimized as follows,

$$\max \sum_{(h,l,t)\in E^+_{h,l,t}} \sum_{(h',l,t')\in E^-_{h',l,t'}} \|v_h + v_l - v_t\| - \|v'_h + v_l - v'_t\| - \gamma \tag{4.10}$$

where $E^+_{h,l,t}$ denote positive for relation tuples generated by Relation Tuple Generation Module, γ is the hyper margin, $E^-_{h',l,t'}$ is composed of training tuples with either **h** or **t** replaced.

4.4 THE JOINT OPTIMIZATION PROBLEM

We now show how three modules introduced above can be organically integrated. Phrase Extraction Module provide entity and relation seeds for Tuple Generation Module and Global Cohesiveness Module. Relation Tuple Generation Module can provide positive tuples for semantic representation learning, in return, global cohesiveness representation serve as good semantic measure during generation process.

Objective for Local Context. The following objective aims at finding semantic consistent tuples in each sentence s,

$$\max \sum_{(h,t)\in E^+_p} \sum_i \|w_i\| A_i(h,t) \tag{4.11}$$

where $\|w_i\| = S(h,l,t)$, $A_i(h,t) = \mathbf{P}(w_{[b_i,b_{i+1})}|h,t)$. $\mathcal{A}$ and $\mathcal{W}$ are calculated via Phrase Extraction Module and Global Cohesiveness Module respectively. In each sentence, it is a discrete problem to find most consistent tuples regarding given entity pairs and scores. Therefore dynamic programming is deployed to find optimal solution of Relation Tuple Generation Module.

Objective for Global Cohesiveness. With global measuring of relation tuples, we have global objective to associate extracted relation tuples in the corpus $\mathcal{D}$ as below,

$$\max \sum_{w_i\in E^+_{h,l,t}} \sum_{\tilde{w}_i\in E^-_{h',l,t'}} \|w_i\| - \|\tilde{w}_i\| - \gamma \tag{4.12}$$

where $E^+_{h,l,t}$ denote for $(h,t) \in E^+_p$ and predicate l stands for average extracted predicate $l = (r_1, r_2, ..., r_n)$ in between, γ is the hyper margin, $E^-_{h',l,t'}$ is composed of training tuples with either **h** or **t** replaced. Global objective tries to maximize margin between positive extractions similarities and negative one's similarity, which start with current positive extractions and iteratively propagate to more unknown tuples in local optimization.

Algorithm 4.1: Joint Tuple Mining

Input: Corpus $\mathcal{D}$, Sentence S, Entities $\mathcal{E}$, Relations $\mathcal{R}$, Word sequence probability$\mathcal{A}$
Output: Relation Tuples $\mathcal{T}$, representation $\mathcal{V}$, similarity measure $\mathcal{W}$
initialize positive E_p^+ according Sec. 4.2 ;
initialize similarity measure $w = 1$;
while L *does not convergence* **do**
  for *each entity argument pair* (e_i, e_j) **do**
    identify semantic path $P(e_i, e_j)$;
    $l \Leftarrow$ from relation tuple generation module given $\mathcal{W} and \mathcal{A}$;
    update relation tuple $(e_i, l, e_j) \in \mathcal{T}$;
  end
  $\mathcal{V}, \mathcal{W} \Leftarrow$ update global cohesiveness module;
  update E_p^+ according to $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{W}$;
end

Update Positive Pairs. Given semantic representation for each entity $\mathbf{e}$ and relation $\mathbf{r}$ and local segmentation between entity pairs, we can update the *Positive Entity Pairs* by finding most semantically consistent subject e_h for each object $\mathbf{e}_t$. By optimizing $\mathbf{P}(r, e_h, e_t)$ in Eq. 4.8, we also obtain the relation tuples for updated positive pairs ${E_p^+}^{n+1}$.

$$E_p^+ = \underset{e_h}{\operatorname{argmax}} \mathbf{P}(r, e_h, e_t) \tag{4.13}$$

Overall Updating Schema. From an overall point of view, the final objective for update is formulated as the sum of both sub-objectives:

$$\mathcal{O} = \mathcal{O}_{local} + \mathcal{O}_{global} \tag{4.14}$$

To maximize above unified open IE objective, see Alg. 4.1, we first initialize positive entity pairs ${E_p^+}^0$. Given entity argument pairs, we perform local optimization, which leads to positive relation tuples $E_{h,l,t}^+$. Note that, at the first round, there is no global representation, so we initialize all $w_i = 1$. Then we update global phrase semantic representation via stochastic gradient descent. With both global cohesiveness information and local segmentation result, ReMine updates positive pairs as described in Sec. 4.2. ReMine solves the integrated problem in a greedy manner, first fix $\mathcal{W}$, Relation Tuple Generation Module selects positive $\mathcal{T}$. Then we maximize global objective by updating $\mathcal{W}$ and $\mathcal{V}$. Finally, fix $\mathcal{W}$, update positive pairs is identical to select new set of $\mathcal{A}$. We iteratively updating local and global objectives until the convergence, which will lead to a stable $\mathcal{W}$, $\mathcal{V}$ and E_p^+.

CHAPTER 5: EXPERIMENTS

In this section, we evaluate the performance of the proposed system on two sub-tasks, *i.e.*, entity recognition (with weak supervision) and relation tuple extraction. We compared output of our system with state-of-the-art entity recognition methods and Open IE systems.

5.1 EXPERIMENTAL SETUP

Datasets. We use three datasets in our experiments: (1) NYT [32]: The training corpus consists of 23.6k sentences from ∼294k 1987-2007 New York Times news articles. 395 sentences are manually annotated with entity and relation mentions by authors [32]. (2) Wiki-KBP [33]: The training corpus contains 2.4k sentences sample from ∼780k Wikipedia articles [33] as training corpus and 290 manually annotated sentences as test data. (3) Twitter [34]: The dataset consists of 1.4 million tweets in Los Angeles with entities and/or noun phrases collected from 2014.08.01 to 2014.11.30.

Training Corpora Distant Linking. Our proposed method ReMine mainly have several outcomes, including high-quality entity and relation phrases and relation tuples. For each corpus, we first generate some distant supervision seeds via DBpedia Spotlight service [35] for entity phrases. With entity phrases, we generate relation phrases between each pair of entity mentions via pattern matching. We then followed the procedure introduced in Sec. 4.1, segmenting input corpora into entity phrases, relation phrases and background phrases.

Phrase Features Generation. In order to estimate type and quality, we designed a set of features $\mathcal{F}$ in Table 5.1 that indicates a good phrase and its type. It 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 into different patterns. For example, "work NN" and "work VBP" are two different semantic patterns. Shang et al. [24] show that considering POS tags in quality predictor yields better performance. Compared with previous phrase mining work, we introduce extra linguistic constraints as prior of phrase segment - dependency parsing tree patterns, which usually bring rich context information. For example, for the sentence "Gov. Tim Pawlenty of Minnesota order the ...", ReMine would segment "[Gov. Tim Pawlenty of Minnesota]" together as a whole entity phrase rather than "[Gov. Tim Pawlenty] [of Minnesota]" since the context-dependent prior prefers one complete tree pattern rather than two separate ones. We applied the Stanford CoreNLP [36] tool to get POS tags and dependency parsing tree. We use same external linguistic features

Table 5.1: List of features used in the phrase extraction module (Sec. 4.1).

Feature	Descriptions
popularity	raw frequency, occurrence probability, log occurrence probability
completeness	sub-phrases within long frequent phrases are also informative
concordance	tokens in quality phrases should co-occurs frequently
punctuations	phrase in parenthesis, quote or has dash after
stopwords	first/last token is stopword and stopword ratio
word shape	first capitalized or all capitalized
part-of-speech tags	unigram and bigram POS tags

as other Open IE methods in our experiments.

Compared Methods. Since ReMine approach information extraction task as two stage process, *i.e.* phrase extraction (or entity detection) and relation tuple extraction, we mainly compare ReMine with other baselines for these two tasks.

We compare entity detection results on the test sets with two state-of-the-art sequence labeling and one phrase mining algorithms (1) Ma & Hovy [37]: adopts a Bi-directional LSTM-CNN structure to encode character embeddings and pre-trained word embeddings. (2) Liu. *et al.* [25]: incorporates a neural language model and conducts multi-task learning to guide sequence labeling; as well as (3) AutoPhrase [24]: the state-of-the-art quality phrase mining method with POS-guided phrasal segmentation. We will not compare ReMine with existing Open IE algorithms on entity detection, since they do not explicitly output entity arguments and same entity will have various boundaries in one sentence.

In the relation tuple extraction task, we consider following approaches for comparison: (1) OLLIE [7] utilizes open pattern learning and extracts patterns over dependency path and part-of-speech tags. (2) ClausIE [8] adopts clause patterns to handle long-distance relationships. (3) Stanford OpenIE [9] learns a clause splitter via distant training data. (4) MinIE [17] refines tuple extracted by ClausIE by identifying and removing parts that are considered overly specific. (5) ReMine-R is a variant of our approach with only the relation tuple generation module. In other words, global cohesiveness measure only plays factors on ranking tuples. (6) ReMine[1] is our proposed approach, in which relation tuple generation module collaborates with global cohesiveness module. All Open IE methods, to some extent, requires weak supervision or distant supervision.

Evaluation Setup. Here we describe the setups for two tasks.

(1) Weakly-supervised entity detection. For this task, NYT and Wiki-KBP are used for evaluation, since both two datasets contain manually-annotated entity mentions in test set. The training data is generated through distant supervision described above without

[1]Code and data of this paper are available at https://github.com/GentleZhu/ReMine.

type information. Considering distant supervision may not be as good as gold annotation, we use high confident seeds for ReMine and other baselines, pushing them towards a fair comparison. 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.

(2) Relation tuple extraction. We aim to compare performance on both normal and short text, so we choose NYT and Twitter dataset for evaluation. For the relation tuple extraction 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 infeasible to know all the "correct" tuples.

5.2 EXPERIMENTS AND PERFORMANCE STUDY

1. Performance on Entity Detection. Although pre-trained models [37, 25] yield extremely high precision in standard sequence labeling tasks *e.g.* NER, POS Tagging, NP Chunking. Regarding open domain extractions, lots of domain-specific entity types are involved, thus, we re-trained these models with ReMine using same supervision. Table 5.2 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.

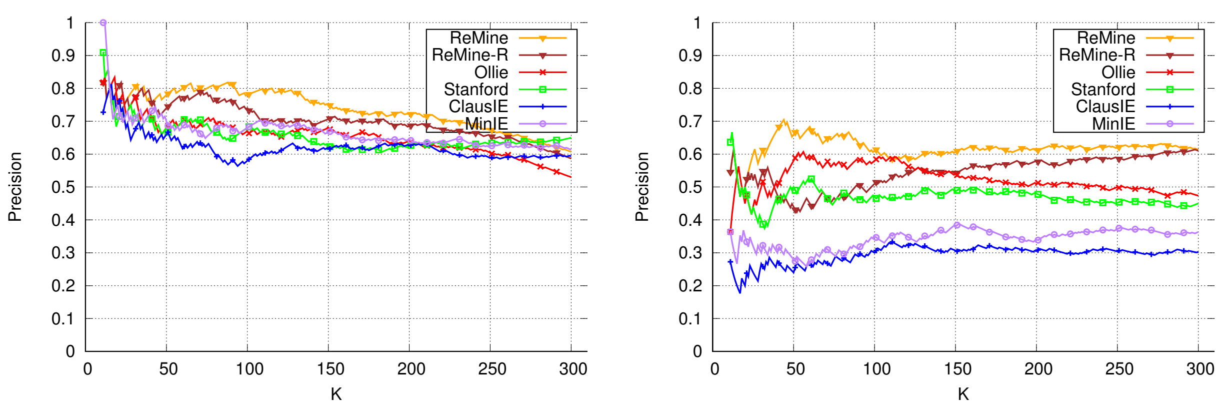
Table 5.2: Performance comparison with state-of-the-art entity recognition algorithms for the weakly-supervised entity detection task.

Methods	**NYT** [32]			**Wiki-KBP** [33]		
	F1	Prec	Rec	F1	Prec	Rec
AutoPhrase [24]	0.531	0.543	0.519	0.416	0.529	0.343
Ma & Hovy [37]	0.664	0.704	0.629	0.324	0.629	0.218
Liu. *et al.* [25]	**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**

2. Performance on Relation Tuple Extraction. Open IE systems can extract information tuples from open domain corpus. We compared ReMine with its own ablation ReMine-R as well as 4 other Open IE systems mentioned above. We manually labeled the extractions got from these extractors. Each extraction was labeled by two independent annotators for 2 rounds. Both annotators are highly proficient and literate in English. The two annotators are asked to evaluate without knowing which model produced the results, eliminating potential bias in evaluation. One extraction is treated as correct only if both labelers think it is correct. Similar to the settings in previous studies [8], we ignore 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 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 in the word segmentation level. Besides, to avoid redundancy, if any component of the tuples contains more than one sentence (*e.g.* (*"we"*, *"like"*, *"coding but he doesn't"*)), the tuples will be labeled as incorrect. We measured the agreement between the two labelers using Cohen's Kappa value. The scores are 0.79 and 0.73 for the NYT dataset and the Twitter dataset respectively.

Table 5.3: Performance of different methods on both datasets.

Dataset	Methods	P@100	P@200	MAP	NDCG@100	NDCG@200	MRR
NYT	ClausIE [8]	0.580	0.625	0.623	0.575	0.667	0.019
	Stanford [9]	0.660	0.585	0.630	0.655	0.726	0.023
	OLLIE [7]	0.670	0.640	0.683	0.684	0.775	**0.028**
	MinIE [17]	0.680	0.645	0.687	0.724	0.723	0.027
	ReMine-R	0.740	0.685	0.726	0.757	0.776	0.027
	ReMine	**0.780**	**0.720**	**0.760**	**0.787**	**0.791**	0.027
Twitter	ClausIE [8]	0.300	0.305	0.308	0.332	0.545	0.021
	Stanford [9]	0.390	0.410	0.415	0.413	0.557	0.023
	OLLIE [7]	0.580	0.510	0.525	0.519	0.626	0.017
	MinIE [17]	0.350	0.340	0.361	0.362	0.541	**0.025**
	ReMine-R	0.510	0.580	0.561	0.522	0.610	0.021
	ReMine	**0.610**	**0.610**	**0.627**	**0.615**	**0.651**	0.022



(a) NYT **(b)** Twitter

Figure 5.1: The Precision@K curves of different Open IE systems on NYT and Twitter datasets.

Among all the Open IE system described above, ReMine and OLLIE extract a relatively small number of tuples. For example, for the first 100 sentences in the NYT test set, both

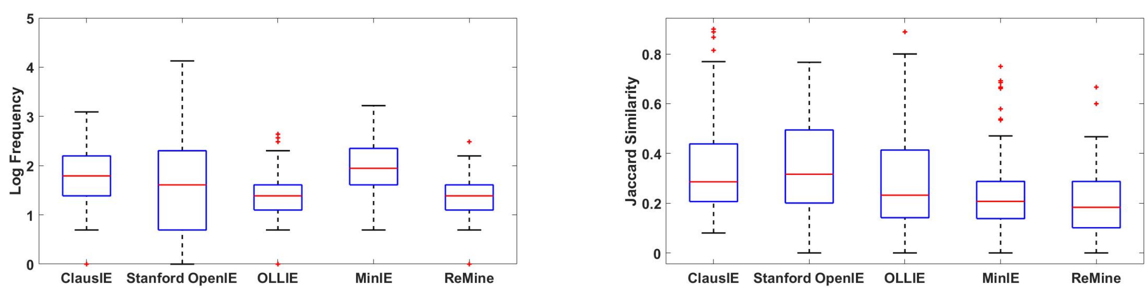
ReMine and OLLIE get about 300 tuples. In contrast, Stanford OpenIE returns more than 1,000 tuples. It may be unfair if we directly plot the $P@k$ curves to compare those methods and ignore the tuple numbers. For example, imagine there is a system returning N tuples. It is not difficult to paraphrase each of them and get another N tuples. If we use the whole $2N$ tuples to plot a $P@k$ curve, we are essentially "stretching" the original curve to a longer one. Since $P@k$ curves are usually monotone decreasing, we will have a "higher" curve after the paraphrase. To alleviate this problem, since each extracted tuple is also assigned a confidence score, we select 300 tuples with the highest scores for each Open IE system to plot the curves. The results are shown in Figure 5.1 and Table 5.3.

According to the curves in Figure 5.1a and 5.1b, ReMine achieves the best performance among all Open IE systems. In the NYT dataset, all the systems except OLLIE actually 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-R and ReMine are usually distinct from each other. In Table 5.3, ReMine also consistently performs the best according to the rank-based measures. In the Twitter dataset, both ClausIE and MinIE have a rather low score since there are lots of non-standard language usages and grammatic errors in tweets. Therefore clause-based methods may not achieve a satisfying performance. In contrast, ReMine shows its power in dealing with short and noisy text.

CHAPTER 6: CASE STUDY

Compared with other Open IE systems, ReMine's generation module produces distinctive results and global cohesiveness module refines extractions. We will show some case studies to reveal overall quality of extractions and effectiveness of specific component.

Clearness and correctness on extractions. In Table. 6.1, 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 could see that 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 logic mistakes in the samples. However, the objective component of R_1 is too complicated to illustrate one proposition clearly. As we mentioned above, this kind 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.



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

Figure 6.1: Distribution over number of extractions and distinctiveness of extractions for different Open IE systems.

Distinctiveness of tuple generation. In our formulation, we try to cover every entity detected in the target sentence while avoid extracting duplicate tuples. In Fig. 6.1a, we show the distribution 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 belong to a single sentence and 157 belong to another. In fact, the latter sentence has only 39 words. This reminds us that

Table 6.1: 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#

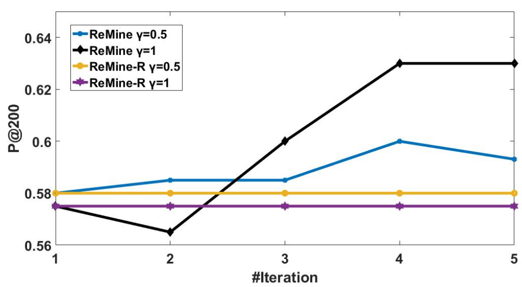
the number of extractions may not be a good alternative of "recall". A more direct way to examine distinctiveness of our extractions is calculating average Jaccard similarity between extractions from same sentence. We present the Jaccard similarity distribution of different systems at Fig. 6.1b, we can clearly see MinIE and ReMine extracts most distinctive facts as they both consider not to be overly specific.

Effectiveness of global evidence. Corpus-level cohesiveness can help reduce local error while generating relation tuples. Especially on twitter set, local linguistic structure fails to attach argument correctly at the first place whereas global cohesiveness module corrects those extractions. In table 6.2, ReMine rejects entity pair (*Liberador*, *Hollywood*) which is not compatible with the predicate "@". This is because in the twitter corpus, it is more common

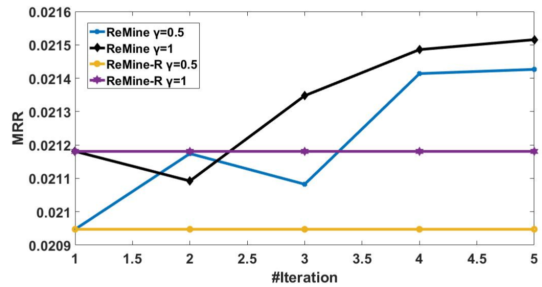
Table 6.2: Different entity pairs discovered by ReMine and ReMine-R, where blue ones are incorrect extractions.

Dudamel conduct his score from Liberador#BeastMode @Hollywood Bowl	
ReMine-R	ReMine
(Dudamel, "conduct", Liberador) *(Dudamel, "conduct...from", #BeastMode)* *(Liberador, "@", Hollywood Bowl)*	*(Dudamel, "conduct", Liberador)* *(Dudamel, "conduct... @", Hollywood Bowl)*

to see *Person @ Place*. Therefore ReMine attaches Hollywood to Dudamel. Comparing our approach with its variant (ReMine-R), we see that by considering global cohesiveness measure, ReMine achieves higher P@200 and MRR by ranking the same set of extractions in Fig. 6.2. We also report ranking performances of the global cohesiveness module, which is not sensitive to the choice of margin γ.



(a) Precision@200



(b) Mean Reciprocal Rank

Figure 6.2: Performances w.r.t number of iterations and γ for ReMine and ReMine-R

CHAPTER 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. ReMine leverages distant supervision in conjunction with existing knowledge bases to provide automatically-labeled sentence and guide the entity and relation segmentation. The local objective is further 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. We develop a joint optimization algorithm to efficiently solve the proposed unified objective function and can output quality extractions by taking into account both local and global information. 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. As a byproduct, ReMine also demonstrates competitive performance on detecting mentions of entities from text when compared to several named entity recognition algorithms.

REFERENCES

[1] S. Sarawagi, "Information extraction," *Foundations and trends in databases*, vol. 1, no. 3, pp. 261–377, 2008.

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

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

[4] 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.

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

[6] 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.

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

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

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

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

[11] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *ACL*, 1992.

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

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

[14] 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.

[15] 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.

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

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

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

[19] 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*, vol. 6, no. 2, pp. 167–195, 2015.

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

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

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

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

[24] 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.

[25] 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.

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

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

[28] Y. Xu, M.-Y. Kim, K. Quinn, R. Goebel, and D. Barbosa, "Open information extraction with tree kernels," in *HLT-NAACL*, 2013.

[29] P. Gamallo, M. Garcia, and S. Fernández-Lanza, "Dependency-based open information extraction," in *ROBUS-UNSUP*, 2012.

[30] X. Zhang and M. Lapata, "Sentence simplification with deep reinforcement learning," in *EMNLP*, 2017.

[31] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," in *HLT-NAACL*, 2016.

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

[33] X. Ling and D. S. Weld, "Fine-grained entity recognition." in *AAAI*, 2012.

[34] 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.

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

[36] 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.

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