Open Information Extraction with Meta-pattern Discovery in Biomedical Literature

Xuan Wang
University of Illinois at Urbana-Champaign
xwang174@illinois.edu

Yu Zhang
University of Illinois at Urbana-Champaign
yuz9@illinois.edu

Qi Li
University of Illinois at Urbana-Champaign
qili5@illinois.edu

Yinyin Chen
University of Illinois at Urbana-Champaign
ychen409@illinois.edu

Jiawei Han
University of Illinois at Urbana-Champaign
hanj@illinois.edu

ABSTRACT

Biomedical open information extraction (BioOpenIE) is a novel paradigm to automatically extract structured information from unstructured text with no or little supervision. It does not require any pre-specified relation types but aims to extract all the relation tuples from the corpus. A major challenge for open information extraction (OpenIE) is that it produces massive surface-name formed relation tuples that cannot be directly used for downstream applications. We propose a novel framework CPIE (Clause+Pattern-guided Information Extraction) that incorporates clause extraction and meta-pattern discovery to extract structured relation tuples with little supervision. Compared with previous OpenIE methods, CPIE produces massive but more structured output that can be directly used for downstream applications. We first detect short clauses from input sentences. Then we extract quality textual patterns and perform synonymous pattern grouping to identify relation types. Last, we obtain the corresponding relation tuples by matching each quality pattern in the text. Experiments show that CPIE achieves the highest precision in comparison with state-of-the-art OpenIE baselines, and also keeps the distinctiveness and simplicity of the extracted relation tuples. CPIE shows great potential in effectively dealing with real-world biomedical literature with complicated sentence structures and rich information.

CCS CONCEPTS

• Information systems → Data mining; • Applied computing → Life and medical sciences;

KEYWORDS

open information extraction; pattern mining; biomedical information extraction; text mining

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM-BCB’18, August 29-September 1, 2018, Washington, DC, USA.
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5794-4/18/08...
https://doi.org/10.1145/3233547.3233594

1 INTRODUCTION

Biomedical information extraction (BioIE) aims to automatically extract structured semantic information from unstructured text. It includes tasks such as named entity recognition, relation extraction, and event detection. BioIE has been successfully applied to many downstream applications such as clinical decision support [12] and question answering [4], biological pathway and network analysis [32], integrative biology [35], biocuration [36, 40, 42] and pharmacovigilance [17]. Machine learning (ML) models are widely adopted in current BioIE systems. For example, in a chemical-induced disease (CID) relation extraction task in the recent BioCreative V challenge [41], systems are proposed using models such as support vector machine [33], feature kernels [24] and convolutional neural network [15]. However, ML models rely on human annotation for training data generation, which is time and labor consuming. Moreover, the reliance on human annotation further limits the systems to certain pre-specified relation types and makes it unable to further extend the systems to new relation types.

Open information extraction (OpenIE), a novel information extraction paradigm, begins to attract great attention in the BioIE domain. OpenIE does not require any pre-specified relation types but aims to extract all the relation tuples from a corpus with no or little human supervision. Two major types of method are proposed for OpenIE: clause-based methods and pattern-based methods. Clause-based methods use linguistic features or sentence structures to induct long-distance relationships from sentences [2, 10, 13, 37]. For example, ClausIE [10] performs clause type analysis based on dependency parsing, chunking, and POS tagging to extract all the possible clauses from the input sentences [10]. In the “Clause extraction” step in Figure 1, eight different clauses can be extracted from the input sentence by ClausIE. The clause-based method extracts massive subject-verb-object relation tuples from the corpus in the surface-name form. Pattern-based methods exploit entity type
information and frequent pattern mining to extract entity-attribute-value (EAV) tuples from input sentences [20, 23, 29]. For example, MetaPAD [20] uses context-aware segmentation and synonymous pattern grouping to extract textural meta-patterns and the value tuples from the input sentence [20]. A meta-pattern is a relationship between entity types. In the “Meta-pattern extraction” step in Figure 1, a meta-pattern “CHEMICAL decreased susceptibility to DISEASE in young SPECIES” can be extracted, which is a relation between three entity types. These meta-patterns can further be used to extract corresponding relation tuples from the corpus.

Both types of OpenIE methods have their pros and cons. Clause-based methods are good at resolving long and complicated sentence structures. However, the massive subject-verb-object relation tuples they extract are in purely surface-name form, which cannot be directly used for downstream applications. Pattern-based methods are easy to be extended to any n-ary relations and produce more structured output with entity type information. However, when the whole sentence has a complicated structure, the tokens between the entities can be lengthy, resulting in the sparsity of relation patterns extracted by the pattern-based method. OpenIE methods have been successfully applied in general domain such as extracting information from newspapers. However, they achieved limited success when applying to the biomedical domain due to some specific challenges with the biomedical text:

(1) Sentences are usually long with complicated structures. The entities within one relation may be far apart in a sentence, and one sentence may contain more than one relation type and more than one relation tuple. This greatly limits the performance of current pattern-based methods.

(2) The entity type information is important for both relation type selection/consolidation and relation instance interpretation. This is not considered by clause-based methods.

(3) High-ary relationships exist in sentences, which can provide more complete and accurate information than binary ones. This is also not considered by clause-based methods.

To address the above challenges, we propose a novel framework CPIE: Clause+Pattern-guided Information Extraction as shown in Figure 1. CPIE combines the merits of clause-based and pattern-based methods, extracting both relation types (meta-pattern synonymous groups) and relation tuples. The framework requires no supervision for clause extraction, meta-pattern extraction and relation tuple extraction, only a few positive examples for quality meta-pattern selection. Compared with previous OpenIE methods, CPIE produces massive but more structured output that can be directly used for downstream applications. We first resolve the long and complicated sentence structures by extracting short clauses from the input sentences. Then we perform meta-pattern extraction on the short clauses. Quality meta-patterns are selected and synonymous meta-patterns are grouped together as a single relation type. Then quality meta-patterns are used to extract relation tuples from the input sentences. Experiments show that our method achieves the highest precision in comparison with state-of-the-art OpenIE baselines, and also keeps the distinctiveness and simplicity of the extracted relation tuples. Case studies also show the power of CPIE in effectively dealing with real-world biomedical literature with complicated sentence structures and rich information.

2 RELATED WORK

Open-Domain Information Extraction. OpenIE aims to find new extraction paradigms and extract large sets of relational tuples from a corpus with no or little human supervision. OpenIE has been extensively studied in the NLP area. The task of OpenIE was first introduced by the seminal work of Banko et al. [3]. After their TextRunner system, most of the existing work follows two lines: clause-based methods and pattern-based methods.

For clause-based analysis, linguistic features, like dependency parsing results, are used to induce long-distance relationships. For example, ReVerb [11] identifies relational phrases via part-of-speech-based regular expressions. Ollie [37] further expands the syntactic scope of relation phrases and allows additional context information such as attribution and clausal modifiers. Similarly, ClausIE [10] infers short but coherent pieces of information along dependency paths, which is typically subject, predicate and optional object with complement. Stanford OpenIE [2] adopts a clause splitter using distant training and mapped predicates to a known relation schema statistically. MiniIE [13] further improves the clearness of relation tuples by introducing different statistical measures like polarity, modality, attribution, and quantities. ReMine [44] integrates local context and global cohesiveness to reduce the amount of uninformative or incoherent tuples.

Pattern-based information extraction can be traced back to Hearst patterns, such as “NP0 such as {NP1, NP2, ...}”, which are used for hyponymy relation extraction [18]. Mitchell et al. [27] introduce Never-Ending Language Learning (NELL) based on free-text predicate patterns. Patty [29] aims to extract a set of typed lexical patterns along the shortest dependency path. MetaPAD [20] generates quality meta-patterns (i.e., relational patterns between entity types) by context-aware segmentation, groups synonymous meta-patterns, and adjusts entity-type levels for appropriate granularity in the pattern groups. TruePIE [23] further adopts truth discovery ideas to extract reliable meta-patterns. Using these patterns to match the corpus, tuples with entities and relation phrases can be extracted.

Biomedical Open Information Extraction. BioIE aims to automatically extract structured semantic information from unstructured biomedical corpus. It includes tasks such as named entity recognition, relation extraction, and event extraction. The methods for BioIE include dictionary-based methods, rule-based methods, statistical methods, classification-based methods and hybrid methods [39, 43]. Several survey papers are published containing thorough and systematic summaries of previous work on BioIE [1, 5–7, 14, 16, 19, 22, 26, 34, 39, 43, 45]. We focus on introducing some related work about BioOpenIE, which is a novel paradigm of BioIE that is most related to our work.

OpenIE, as a novel information extraction paradigm, begins to attract great attention in BioIE domain. It does not require any pre-specified relation types but aims to extract all the relation tuples from a corpus with no or little human supervision. Liu et al. [25] introduce some applications of OpenIE to BioIE in their survey paper. For example, Attias et al. [28] present an OpenIE system for biomedical named entity recognition based on NELL [27], and propose a method for assessing seed qualities to prevent semantic drift. Nebot et al. [30, 31] propose a scalable method to extract surface-form biomedical relationships not specific to any relation type and
Amphetamine and cocaine decreased susceptibility to myoclonus in young mice and increased susceptibility in mature mice.

Clauses extraction
- CHEMICAL_Amphetamine "decreased" susceptibility to DISEASE_myoclonus in young SPECIES_mice
- CHEMICAL_cocaine "decreased" susceptibility in young SPECIES_mice
- CHEMICAL_Amphetamine "decreased" susceptibility
- CHEMICAL_cocaine "decreased" susceptibility
- CHEMICAL_Amphetamine "increased" susceptibility in mature SPECIES
- CHEMICAL_cocaine "increased" susceptibility

Output: relationships
CHEMICAL: Amphetamine decreased susceptibility to DISEASE: myoclonus in young SPECIES: mice
CHEMICAL: cocaine decreased susceptibility to DISEASE: myoclonus in young SPECIES: mice
CHEMICAL: decreased susceptibility to DISEASE in young SPECIES
CHEMICAL: increased susceptibility in mature SPECIES

Figure 1: The overall framework of CPIE: Clause+Pattern-guided Information Extraction.

4 CLAUSE EXTRACTION
As mentioned above, if the whole sentence has a complicated structure, the tokens between the entities can be lengthy, resulting in the sparsity of relation patterns. To alleviate this problem, we first extract clauses, which are simplified and express some coherent piece of information, from the sentence.

Definition 4.1 (Clause Extraction). A clause is a part of a sentence that consists of one subject (S), one verb (V), and optionally an indirect object (O), a direct object (O), a complement (C), and one or more adverbials (A). The goal of clause extraction is to segment the whole sentence into clauses and represent them in the format of (subject, relation, optional components).

We use ClausIE [10] for clause extraction. We first concatenate all the words in each entity mention with underlines to make them an integrated unit. Then we select those sentences with at least one typed entity as the input for ClausIE.

For each input sentence, ClausIE conducts the following steps:
Dependency Parsing. We adopt Stanford dependency parser [21] to discover the syntactical structure of a sentence. The output parsing tree has a set of directed syntactic relations between the words in the sentence. Figure 2 shows an example. The root of the tree is the verb “appear”. It is connected to “doses” via a subject relation (nsubj) and to “useful” via a complement relation (xcomp). For a complete list of relations, please refer to https://nlp.ml.io/jg/software/pac/standep.html.

Clause Identification. According to the dependency parsing tree, ClausIE first constructs a clause for each subject relation (e.g., nsubj) connecting the subject (S) and the governor verb (V). Besides, objects (O) and complements (C) linked with V through dobj, iobj,
Table 1: Patterns and clause types we use [10]. S: Subject, V: Verb, C: Complement, O: Direct object, O_i: Indirect object, A : Adverbial, V_r: Intransitive verb, V_c: Copular verb, V_e: Extended-copular verb, V_{mt}: Monotransitive verb, V_{dt}: Ditransitive verb, V_{ct}: Complex-transitive verb.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Type</th>
<th>Example</th>
<th>Derived Clauses</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV_t</td>
<td>SV</td>
<td>Colectomy works.</td>
<td>(Colectomy, works)</td>
</tr>
<tr>
<td>SV_r A</td>
<td>SVA</td>
<td>Pulmonary toxicity is in lungs.</td>
<td>(Pulmonary toxicity, is, in lungs)</td>
</tr>
<tr>
<td>SV_r C</td>
<td>SVC</td>
<td>Pulmonary toxicity is vital.</td>
<td>(Pulmonary toxicity, is, vital)</td>
</tr>
<tr>
<td>SV_{mt} O</td>
<td>SVO</td>
<td>Nitrofurantoin causes pulmonary toxicity.</td>
<td>(Nitrofurantoin, causes, pulmonary toxicity)</td>
</tr>
<tr>
<td>SV_{dt} O, O</td>
<td>SVOO</td>
<td>Colectomy gives the patient a chance.</td>
<td>(Colectomy, gives, the patient, a chance)</td>
</tr>
<tr>
<td>SV_{ct} O, A</td>
<td>SVOA</td>
<td>Colectomy removes the colon away.</td>
<td>(Colectomy, removes, the colon, away)</td>
</tr>
<tr>
<td>SV_{ct} O, C</td>
<td>SVOC</td>
<td>Pulmonary toxicity causes rats to die.</td>
<td>(Pulmonary toxicity, causes, rats, to die)</td>
</tr>
<tr>
<td>SV_r AA</td>
<td>SV</td>
<td>Pulmonary toxicity often appears in lungs in rats.</td>
<td>(Pulmonary toxicity, often appears)</td>
</tr>
<tr>
<td>SV_r AA</td>
<td>SVA</td>
<td>Pulmonary toxicity is often in lungs in rats.</td>
<td>(Pulmonary toxicity, is, often in lungs)</td>
</tr>
<tr>
<td>SV_r CA</td>
<td>SVC</td>
<td>Pulmonary toxicity is vital in the last century.</td>
<td>(Pulmonary toxicity, is, vital)</td>
</tr>
<tr>
<td>SV_{mt} O, A</td>
<td>SVO</td>
<td>Nitrofurantoin causes pulmonary toxicity in rats.</td>
<td>(Nitrofurantoin, causes, pulmonary toxicity)</td>
</tr>
<tr>
<td>ASV_{mt} O</td>
<td>SVO</td>
<td>In rats, Nitrofurantoin causes pulmonary toxicity.</td>
<td>(Nitrofurantoin, causes, pulmonary toxicity)</td>
</tr>
</tbody>
</table>

Figure 2: An example sentence with dependency parse.

xcomp or ccomp, as well as adverbials (A) related with V through advmod, advcl or prep_in will also be included.

Clause Type Analysis. Once clauses have been identified, ClausIE tries to analyze the type of each clause. A complete list of all clause types used in [10] is given in Table 1. Based on the lexical and syntactical structure, the clause type classification is essentially a decision tree. The complete decision flow can be found in [10].

Proposition Generation. Given the clause type, we are able to decide which tokens to place into the subject, the relation, and the optional components. ClausIE maps the subject of the clause to the subject of the proposition. For the optional components, an argument is created and includes the tokens following the verb and then the tokens preceding the verb, in the order in which they appear.

If there is no clause extracted for an input sentence, we will keep the original input sentence as the output clause. The output clauses are further selected to contain at least one typed entity for meta-pattern extraction.

5 META-PATTERN EXTRACTION

Definition 5.1 (Meta-pattern Extraction). Given a corpus C as a list of typed sentences (clauses) S, i.e., C = \{S_1, S_2, ..., S_n\}, each sentence is a sequence of word tokens t, i.e., S = t_1t_2...t_m, in which t_j ∈ T ∪ W (T is the set of entity types and W is the set of non-type words). The goal of meta-pattern extraction is to extract meta-patterns mp, which are sub-sequences of the typed sentences (clauses) S that contain at least one token in the set of entity types T. Quality meta-patterns are selected by some criteria and further grouped into synonymous groups MPG = \{mp_1, mp_2, ..., mp_k\}, in which mp_i and mp_j are synonymous meta-patterns.

Candidate meta-pattern generation. We assume the meta-patterns for relationship extraction should contain at least two typed entities. Given the typed clauses from the previous step, we take the segment between the first and last typed entity in each input clause as a candidate meta-pattern. This step generates a large number of noisy meta-patterns. These candidate meta-patterns are then used for quality selection.

5.1 Quality meta-pattern selection

Given a large number of candidate meta-patterns, we use a set of contextual features to train a classifier that estimates the quality of each candidate meta-pattern. These contextual features have also been adopted by MetaPAD [20].

1. Frequency: A quality meta-pattern should occur frequently in the corpus. We use the normalized count of each meta-pattern c(mp)/N, where N is the total number of word tokens in the corpus, to measure the frequency of each meta-pattern. We ignore all the meta-patterns that appear less than 10 times in the corpus, i.e., c(mp) < 10.

2. Concordance: A quality meta-pattern should have a frequency significantly higher than that is expected due to chance, which
is a higher concordance. The null hypothesis is: word tokens in the corpus is generated from a series of independent Bernoulli trials. Suppose the total number of word tokens \( N \) in the corpus is very large. The expected frequency of a pair of sub-patterns \( (mp_1, mp_r) \) under the null hypothesis is:

\[
\mu_0(c((mp_1, mp_r))) = N \cdot p(mp_1) \cdot p(mp_r), \tag{1}
\]

where \( p(mp) = \frac{c(mp)}{N} \) is the empirical probability of the meta-pattern \( mp \). For each meta-pattern \( mp \), we examine all the divisions of dividing \( mp \) into \( (mp_1, mp_r) \) without overlapping. We use \( Z \) score to measure the concordance of each meta-pattern \( mp \) by finding the division \( (mp_1, mp_r) \) with the maximum \( Z \) score:

\[
Z(mp) = \max_{(mp_1, mp_r) = mp} \frac{c(mp) - \mu_0(c((mp_1, mp_r)))}{\sigma_{(mp_1, mp_r)}}, \tag{2}
\]

where \( \sigma_{(mp_1, mp_r)} \) is the standard deviation of the counts of \( mp_1 \) and \( mp_r \). For example, the meta-pattern “CHEMICAL against CHEMICAL induced DISEASE” may have a higher concordance than the meta-pattern “GENE, GENE and GENE”, thus is more likely to be a quality meta-pattern.

(3) Informativeness: A quality meta-pattern should have more informative context words. We use the average inverse document frequency (IDF) score of all the non-type context words \( cw(mp) \) in each meta-pattern \( mp \) as its informativeness score:

\[
Informativeness(mp) = \frac{\sum_{w \in cw(mp)} \log \frac{N}{M(w)}}{|cw(mp)|}, \tag{3}
\]

where \( M(w) \) is the total number of meta-patterns containing the context word \( w \). For example, the meta-pattern “CHEMICAL induced GENE” may be more informative than the meta-pattern “CHEMICAL and GENE”, thus is more likely to be a quality meta-pattern.

(4) Completeness: A quality meta-pattern should have a higher ratio between the frequencies of the meta-pattern and its sub-patterns. We use the average frequency ratio between each meta-pattern \( mp \) and all its sub-patterns \( sub(mp) \) as its completeness score:

\[
Completeness(mp) = \frac{\sum_{s \in sub(mp)} \frac{c(mp)}{c(s)}}{|sub(mp)|}. \tag{4}
\]

For example, the meta-pattern “CHEMICAL against CHEMICAL induced DISEASE” may be more complete than its sub-pattern “CHEMICAL against”, thus is more likely to be a quality meta-pattern.

(5) Coverage: A quality meta-pattern should extract more distinct instances from the corpus. We used the normalized count of the number of distinct instances \( ic(mp) / N \) extracted by each meta-pattern \( mp \) to measure the coverage of each meta-pattern. After extracting the above features for each meta-pattern, we build a random forest based classifier to learn the quality function \( Q(mp) \) that maps each meta-pattern to a quality score. We manually selected less than 20 positive and negative meta-patterns, respectively, as the training examples for the model training. After obtaining the quality score of each meta-pattern, we select the meta-pattern with a quality score above a threshold \( \delta \) as quality meta-patterns. Here we use \( \delta = 0.9 \). We also assign the same quality score to the extracted instances of each meta-pattern as the instance score for comparison with other OpenIE models.

5.2 Synonymous meta-pattern grouping

Synonymous meta-patterns expressing the same kind of relationship should be grouped together to reduce the redundancy of extracted relationship types and enrich the relationship instances under each relationship type. We group synonymous meta-patterns under the following two assumptions:

(1) Synonymous meta-patterns should have the same entity types. For example, “CHEMICAL induces DISEASE” and “CHEMICAL induces GENE” cannot be synonymous meta-patterns.

(2) Synonymous meta-patterns should share similar extracted relationship instances. For example, in Figure 3, the four meta-patterns share some extracted instances and all expressing the meaning of “CHEMICAL induces DISEASE”, thus should be grouped as a single relationship type.

![Figure 3: Grouping synonymous patterns by their shared instance sets.](image)

We use a different method for pattern grouping compared with MetaPAD. For each meta-pattern, we generate the term frequency-inverse document frequency (TF-IDF) vector of its instance sets as its feature vector. The instances also contain the type information. Then we perform k-means clustering to group synonymous meta-patterns in groups. Since the number of groups cannot be pre-decided, we set k to be the total number of quality meta-patterns divided by 5. We assume that on average, each group will contain 5 synonymous meta-patterns. Examples of synonymous meta-pattern groups are discussed in Section 6.3.

6 EXPERIMENTS

6.1 Dataset

CPIE only requires biomedical text corpus as input. To test the effectiveness of CPIE, we collect a subset of PubMed paper abstracts. The PubMed id (PMID) of each selected paper can be found in the Comparative Toxicogenomics Database (CTD) [8], in which all the selected PMIDs are shown to contain some biomedical relationships. CTD is a human-curated database containing biological entities and their relationships. The entity and relationship statistics of our collected papers in CTD are shown in Table 2.

We focus on three entity types in CTD: gene, chemical, and disease. Among these three entity types, there are three relation types in CTD: chemical-gene, chemical-disease, and disease-gene relationships. We first randomly select 248,064 relationships from
Table 2: Statistics of the CTD dataset subset used in our experiments.

<table>
<thead>
<tr>
<th></th>
<th># of Entity</th>
<th># of Relation</th>
<th># of PMID</th>
</tr>
</thead>
</table>
| Chemical-Gene    | Chemical: 7,187  
Gen: 667 | 163,126 | 26,786 |
| Chemical-Disease | Chemical: 7,187  
Disease: 598 | 84,104 | 1,619 |
| Disease-Gene     | Gene: 669  
Disease: 599 | 834 | 1,622 |
| Total            | Chemical: 7,187  
Gene: 669  
Disease: 599 | 248,064 | 28,007 |

Table 3: Performance comparison with state-of-the-art OpenIE systems, using Precision (P@k), Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG@k), k = 50, 100.

<table>
<thead>
<tr>
<th></th>
<th>P@50</th>
<th>P@100</th>
<th>MAP</th>
<th>NDCG@50</th>
<th>NDCG@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClausIE [10]</td>
<td>0.800</td>
<td>0.860</td>
<td>0.800</td>
<td>0.710</td>
<td>0.875</td>
</tr>
<tr>
<td>Stanford [2]</td>
<td>0.800</td>
<td>0.800</td>
<td>0.765</td>
<td>0.714</td>
<td>0.870</td>
</tr>
<tr>
<td>Ollie [37]</td>
<td>0.920</td>
<td>0.890</td>
<td>0.917</td>
<td>0.935</td>
<td>0.982</td>
</tr>
<tr>
<td>MiniIE [13]</td>
<td>0.840</td>
<td>0.830</td>
<td>0.820</td>
<td>0.800</td>
<td>0.918</td>
</tr>
<tr>
<td>CPIE</td>
<td>0.940</td>
<td>0.900</td>
<td>0.956</td>
<td>0.956</td>
<td>0.991</td>
</tr>
</tbody>
</table>

6.2 Performance comparison

Baselines. To show the effectiveness of CPIE, we compare it with the following state-of-the-art OpenIE approaches:

- **ClausIE** [10] adopts clause patterns to handle long-distance relationships.
- **Stanford OpenIE** [2] learns a clause splitter via distant training data.
- **Ollie** [37] utilizes open pattern learning and extracts patterns over dependency path and part-of-speech tags.
- **MiniIE** [13] refines tuples extracted by ClausIE by identifying and removing parts that are considered overly specific.

Evaluation Metrics. We randomly sample 100 sentences from the 28,007 input PubMed abstracts for performance comparison. All of the compared benchmarks, as well as our method, will assign a confidence score to each extracted tuple. We rank all the tuples according to their confidence scores. Based on the ranking list, the following measures can be adopted: (1) $P@k$ is the ratio of correct tuples in the top $k$ extractions. (2) $MAP$ is the mean average precision of the whole ranking list. (3) $NDCG@k$ is the normalized discounted cumulative gain at rank $k$. The detailed definition of these common ranking measures can be found in [38]. Note that we do not use recall in OpenIE since it is infeasible to know all the “correct” tuples.

For each method, we select the top 100 tuples from its ranking list and manually label them. The annotator is asked to evaluate without knowing which model produced the results, eliminating potential bias in evaluation. Similar to the settings in previous studies [10], one tuple will be judged as correct if it reads smoothly and meets the fact described in the sentence. For example, both (“DISEASE”, “is”, “DISEASE”) and (“DISEASE”, “induced by”, “CHEMICAL in SPECIES”) are correct. However, (“SPECIES”, “is”, “a CHEMICAL”) and (“CHEMICAL”, “inhibited”, “GENE and”) will not be counted since they have logical or syntactical mistakes. Besides, each tuple should describe exactly one proposition. Tuples with zero or more than two propositions (e.g., (“CHEMICAL”, “and”, “CHEMICAL”) and (“DISEASE”, “is”, “induced by CHEMICAL and has no relationships with GENE”)) will be labeled as incorrect.

Results. Table 3 and Figure 4 demonstrate the performances of different OpenIE approaches. Our method CPIE can be seen as clause extraction + meta-pattern discovery. We also tried MetaPAD alone without clause extraction but it doesn’t work due to pattern sparsity in biomedical text. In Table 3, our method is consistently the best according to the rank-based measures, especially compared with ClausIE without meta-pattern discovery. Ollie ranks the second and achieves a similar $P@100$ with our method. However, for $MAP$ and $NDCG@50$, CPIE outperforms Ollie by a large margin, indicating we have a rather high precision for top-ranked tuples. In Figure 4, our curve is higher than other baselines for any $k$. Besides, we can observe that for CPIE and Ollie, $P@k$ decreases with $k$, which means the tuple score is a good indicator of the correctness. From this perspective, the features we defined in Section 5.1 are reliable in evaluating the quality of tuples or meta-patterns.

Distinctiveness and Simplicity. For OpenIE, there are two other important criteria.

- **Distinctiveness**: For the same sentence, we expect that the extracted tuples should have different semantics with each other. It is not satisfying if they are just paraphrasing each other.
- **Simplicity**: Each tuple should clearly explain only one proposition. The ideal cases could be using phrases with two or three words to represent subjects, predicates, and objects.

A direct way to examine distinctiveness of our extractions is to calculate the average Jaccard similarity between extractions from
We first look at the quality meta-patterns and the synonymous where

\[ \text{consider} \begin{cases} 2\text{-ary relations since multi-ary ones may be extensions of them.} \end{cases} \]

patterns have the potential to be further resolved and consolidated
date meta-patterns in total. Most of the candidate meta-patterns
pattern generation, we get 185,403 two-entity and three-entity can-
some extracted relation tuples in detail.

6.3 Case study

6.3.1 Quality meta-pattern distribution. After candidate meta-
the same sentence. Suppose we get \( n \) tuples \( \{(h_1, r_1, t_1), (h_2, r_2, t_2), \ldots, (h_n, r_n, t_n)\} \) from sentence \( s \), the average Jaccard similarity is defined as

\[
\text{Avg Jacc Sim}(s) = \frac{1}{\binom{n}{2}} \sum_{1 \leq i < j \leq n} \frac{|\{|h_i, r_i, t_i\} \cap \{|h_j, r_j, t_j\}|}{|\{|h_i, r_i, t_i\}| \cup \{|h_j, r_j, t_j\}|}.
\]

where \( \{h_i, r_i, t_i\} \) denotes the set of tokens appearing in any component of tuple \( i \).

We present the average Jaccard similarity distribution of all sentences in Figure 5, from which we can clearly see that Ollie and CPIE extracts the most distinctive facts as they both consider not to be overly specific. In contrast, ClausIE and Stanford OpenIE suffer for the duplication problem.

To evaluate simplicity, we calculate the average tuple length for each sentence:

\[
\text{Avg Tup Len}(s) = \frac{1}{n} \sum_{i=1}^{n} |\{|h_i, r_i, t_i\}|.
\]

For different approaches, the distributions of the average tuple length are shown in Figure 6. CPIE again performs the best since meta-patterns are considered to be short and clear. For the tuples extracted by ClausIE, the object component is always very complicated since ClausIE directly segment the whole sentence into several parts. Sometimes the tuples are too long to illustrate one proposition clearly. According to our evaluation protocol, these tuples will be judged as wrong. In fact, even if the long tuples are logically and syntactically correct, it may not help downstream applications.

6.3 Case study

We first look at the quality meta-patterns and the synonymous meta-pattern groups as extracted relation types. Then we examine some extracted relation tuples in detail.

6.3.1 Quality meta-pattern distribution. After candidate meta-pattern generation, we get 185,403 two-entity and three-entity candidate meta-patterns in total. Most of the candidate meta-patterns appear less than 10 times in the corpus. Those low-frequency meta-patterns have the potential to be further resolved and consolidated with the high-frequency meta-patterns. In this study, we ignore those low-frequency meta-patterns, which results in around 1,000 candidate meta-patterns for quality validation. After ranking these candidate meta-patterns with their quality score (a score between 0 and 1), we selected those meta-patterns with a quality score above 0.9 as quality meta-patterns, which results in around 200 quality meta-patterns.

The count distribution of quality meta-patterns is shown in Figure 7. Different types of the meta-patterns are in different colors in the pie chart. For example, “CHEMICAL” means all the entities in the patterns are chemicals, and “GENE, CHEMICAL” means the pattern consists of both chemicals and genes regardless of their order. From the pie chart, we can see that most of the meta-patterns are relationships between gene and chemical, followed by meta-patterns between disease and chemical, and meta-patterns between chemicals.

In Table 4, we list the top 15 quality meta-patterns and their counts. These top meta-patterns describe concrete biomedical relationships between different entity types, which indicates that our quality pattern selection method performs well. The typical raw counts of these quality patterns are around 100 to 300, which is in the medium range. Apparently, the frequency is not the most indicating feature for quality meta-patterns. Informativeness and coverage play a more important role in identifying quality meta-patterns according to our feature importance analysis of the random forest classifier.

6.3.2 Synonymous meta-pattern groups. We group quality meta-patterns into around 40 synonymous groups. In Table 5, we list some examples of the synonymous pattern groups. For example, the first group refers to the relation type “CHEMICAL induce DISEASE”, which includes meta-patterns such as “CHEMICAL causes DISEASE”, “CHEMICAL leads to DISEASE”, “CHEMICAL induces DISEASE” and “CHEMICAL exposure causes DISEASE”. The meta-patterns within each group has very close semantic meaning and each group can be regarded as a specific relation type. These synonymous meta-pattern groups show the effectiveness of CPIE in automatically extracting relation types from massive corpus without supervision. By grouping synonymous meta-patterns together, we reduce the redundancy of extracted relation types and enrich...
Forskolin (10 microM), an activator of adenylyl cyclase, and terbutaline could be a more accurate subject in the "increased" relation.

One may note that, in this example, "renin mRNA and prorenin extract them under the relation of "CHEMICAL increased GENE".

such a long and complicated sentence, CPIE is able to accurately though the two entities, "forskolin" and "renin", are far apart in

cAMP levels, also increased renin mRNA and prorenin release. " Al-
tatine (100 microM), a beta2-adrenergic agonist known to increase

The Pubmed abstract with PMID "9256163". The original sentence is:

equal pattern extraction methods to directly extract meta-patterns
tual patterns for long-distance pattern discovery. How to extend the
tence structures. However, the ClausIE output is often noisy and

from input sentences and resolve the long and complicated sen-
ment for prepossessing may further improve the performance of our system.

Moreover, there are more than one relation type and more than
one relation instance in the above sentence. CPIE will also be able to
eat the relation tuple ("terbutaline", "renin") under the relation
type "CHEMICAL increased GENE", and the relation tuple ("terbut-
taline", "cAMP") under the relation type "CHEMICAL increased

7 DISCUSSION

7.1 Biomedical named entity recognition with distant supervision

Current meta-patterns only include the most common biomedical entity types - gene, chemical, and disease. Recognizing more entity types will further enrich the extracted meta-patterns and relation tuples. For example, this is a sentence from PubMed papers with PMID "236533": "Colectomy is more effective than high-dose steroid therapy in reversing the growth retardation caused by ulcerative colitis and is of greatest value if not delayed too long." In this sentence, "colectomy" and "steroid therapy" are treatment technologies and "ulcerative colitis" is a disease. If we can recognize the entity type "TREATMENT", we will be able to extract a relation type "TREATMENT treat DISEASE" from the above sentence.

Most biomedical named entity recognition systems use supervised machine learning models, which require human labeled training dataset for model development. However, it may not be possible to acquire the human labeled training data for all the entity types that we are interested in. One way is to leverage distant supervision, which automatically labels the corpus by some distant examples in the knowledge base and then use this partially labeled corpus for model training. For example, Medical Subject Headings (MeSH) is a knowledge base for biomedical entities. We can find "colectomy" and "steroid therapy" are treatment technologies and "ulcerative colitis" is a disease. If we can recognize the entity type "TREATMENT", we will be able to extract a relation type "TREATMENT treat DISEASE" from the above sentence.

7.2 Extend meta-pattern extraction

The current framework utilizes ClausIE to extract short clauses from input sentences and resolve the long and complicated sentence structures. However, the ClausIE output is often noisy and redundant, and errors from ClausIE can be propagated down to the next step of meta-pattern extraction. One way is to directly extend the texture meta-pattern extraction method to long and complicated sentence structures. For example, patterns on the dependency-parsing tree of the sentence can be incorporated with sequential textual patterns for long-distance pattern discovery. How to extend the meta-pattern extraction methods to directly extract meta-patterns on such complicated sentences is an interesting problem.
Table 5: Examples of synonymous groups of the quality meta-patterns.

<table>
<thead>
<tr>
<th>Synonymous group</th>
<th>Meta-patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHEMICAL induce DISEASE</td>
<td>CHEMICAL causes DISEASE</td>
</tr>
<tr>
<td></td>
<td>CHEMICAL leads to DISEASE</td>
</tr>
<tr>
<td></td>
<td>CHEMICAL induces DISEASE</td>
</tr>
<tr>
<td></td>
<td>CHEMICAL exposure causes DISEASE</td>
</tr>
<tr>
<td>CHEMICAL inhibit GENE</td>
<td>CHEMICAL decreased GENE</td>
</tr>
<tr>
<td></td>
<td>CHEMICAL decreases GENE</td>
</tr>
<tr>
<td></td>
<td>CHEMICAL inhibition of GENE</td>
</tr>
<tr>
<td></td>
<td>CHEMICAL suppressed GENE</td>
</tr>
<tr>
<td>CHEMICAL no effect on GENE</td>
<td>CHEMICAL had no effect on GENE</td>
</tr>
<tr>
<td></td>
<td>CHEMICAL did not affect GENE</td>
</tr>
<tr>
<td>DISEASE induced by CHEMICAL in SPECIES</td>
<td>DISEASE induced by CHEMICAL in SPECIES</td>
</tr>
<tr>
<td></td>
<td>DISEASE be induced by CHEMICAL in SPECIES</td>
</tr>
<tr>
<td>SPECIES treated with CHEMICAL</td>
<td>SPECIES were pretreated with CHEMICAL</td>
</tr>
<tr>
<td></td>
<td>SPECIES were administered with CHEMICAL</td>
</tr>
<tr>
<td></td>
<td>CHEMICAL treated SPECIES</td>
</tr>
<tr>
<td></td>
<td>SPECIES be induced by CHEMICAL</td>
</tr>
<tr>
<td></td>
<td>SPECIES were exposed to CHEMICAL</td>
</tr>
</tbody>
</table>

Table 6: Examples of relationship instances of the quality meta-patterns.

<table>
<thead>
<tr>
<th>Meta-pattern</th>
<th>Entity 1</th>
<th>Entity 2</th>
<th>Entity 3</th>
<th>PMID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHEMICAL increased GENE</td>
<td>forskolin</td>
<td>renin</td>
<td></td>
<td>9256163</td>
</tr>
<tr>
<td></td>
<td>nicotine</td>
<td>GM-CSF</td>
<td></td>
<td>9606035</td>
</tr>
<tr>
<td></td>
<td>TCDD</td>
<td>TGF-alpha</td>
<td></td>
<td>11309286</td>
</tr>
<tr>
<td></td>
<td>CP461</td>
<td>PKG</td>
<td></td>
<td>11602670</td>
</tr>
<tr>
<td></td>
<td>beta-naphthoflavone</td>
<td>CYP1B1</td>
<td></td>
<td>12843640</td>
</tr>
<tr>
<td>CHEMICAL inhibited DISEASE</td>
<td>rapamycin-FKBP</td>
<td>retinoblastoma</td>
<td></td>
<td>7532117</td>
</tr>
<tr>
<td></td>
<td>bile salts</td>
<td>cholestasis</td>
<td></td>
<td>12644037</td>
</tr>
<tr>
<td></td>
<td>TAS-108</td>
<td>tumor</td>
<td></td>
<td>15671561</td>
</tr>
<tr>
<td></td>
<td>zinc</td>
<td>cytotoxicity</td>
<td></td>
<td>15922008</td>
</tr>
<tr>
<td></td>
<td>DY-9760e</td>
<td>brain edema</td>
<td></td>
<td>16987238</td>
</tr>
<tr>
<td>DISEASE be induced by CHEMICAL in SPECIES</td>
<td>pulmonary toxicity</td>
<td>nitrofurantoin</td>
<td>rats</td>
<td>1313237</td>
</tr>
<tr>
<td></td>
<td>colon cancers</td>
<td>Pino-Pi</td>
<td>rats</td>
<td>14507667</td>
</tr>
<tr>
<td></td>
<td>metabolic disorders</td>
<td>PFOA</td>
<td>human</td>
<td>23978341</td>
</tr>
<tr>
<td></td>
<td>carcinogenesis</td>
<td>arsenic</td>
<td>humans</td>
<td>19524636</td>
</tr>
<tr>
<td></td>
<td>liver injury</td>
<td>carbon tetrachloride</td>
<td>rats</td>
<td>17173083</td>
</tr>
<tr>
<td>CHEMICAL induces apoptosis in SPECIES DISEASE</td>
<td>arsenic trioxide</td>
<td>human</td>
<td>gastric cancer</td>
<td>11146441</td>
</tr>
<tr>
<td></td>
<td>flavopiridol</td>
<td>human</td>
<td>leukemia</td>
<td>11464216</td>
</tr>
<tr>
<td></td>
<td>isoflurane</td>
<td>rat</td>
<td>pheochromocytoma</td>
<td>18227305</td>
</tr>
<tr>
<td></td>
<td>sodium butyrate</td>
<td>human</td>
<td>hepatoma</td>
<td>15177505</td>
</tr>
<tr>
<td></td>
<td>butyrate</td>
<td>human</td>
<td>hepatoma</td>
<td>15177505</td>
</tr>
<tr>
<td>CHEMICAL against CHEMICAL induced DISEASE</td>
<td>dimercaptosuccinic acid</td>
<td>arsenic</td>
<td>toxicity</td>
<td>15998567</td>
</tr>
<tr>
<td></td>
<td>zinc</td>
<td>cadmium</td>
<td>disorders in bone metabolism</td>
<td>23726800</td>
</tr>
<tr>
<td></td>
<td>selenium</td>
<td>cadmium</td>
<td>hematological disturbances</td>
<td>24954678</td>
</tr>
<tr>
<td></td>
<td>resveratrol</td>
<td>cisplatin</td>
<td>testicular damage</td>
<td>28606469</td>
</tr>
<tr>
<td></td>
<td>erdogestine</td>
<td>acetaminophen</td>
<td>renal toxicity</td>
<td>16532256</td>
</tr>
</tbody>
</table>

8 CONCLUSIONS

We propose a novel framework CPIE: Clause+Pattern-guided Information Extraction that automatically extract both relation type and relation tuples with little supervision. CPIE first resolves the long and complicated sentence structures by clause extraction and then uses texture meta-patterns to extract n-ary tuples with entity type information. Quality meta-patterns are selected and synonymous meta-patterns are grouped to produce more structured output for downstream application. Our method achieves the highest precision in comparison with state-of-the-art OpenIE baselines and keeps the distinctiveness and simplicity of extracted relation tuples. Case studies also show the power of CPIE in effectively dealing with real-world biomedical literature with complicated sentence structures and rich information. Future work to improve our framework includes: (1) include more entity types with distant supervision, (2) replace ClausIE with an extended meta-pattern extraction method to reduce error propagation.
ACKNOWLEDGMENTS

The research was sponsored in part by grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov), U.S. Army Research Lab under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA under Agreement No. W911NF-17-C-0099, National Science Foundation IIS-16-18481, IIS-17-04532, and IIS-17-41317, and DTRA HDTRA11810026. Any opinions, findings, and conclusions or recommendations expressed in this document are those of the author(s) and should not be interpreted as the views of any U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[22] Qi Li, Meng Jiang, Xikun Zhang, Meng Qu, and Jiwaei Han. 2018. TruePIE: Discovering reliable patterns in pattern-based information extraction. In KDD’18. ACM.