# Multi-Round Parsing-based Multiword Rules for Scientific Knowledge Extraction

Joseph Kuebler, Lingbo Tong, Meng Jiang
*University of Notre Dame, Notre Dame, IN 46556, USA*
{jkuebler, ltong2, mjiang2}@nd.edu

***Abstract*—Information extraction (IE) in scientific literature has facilitated many down-stream knowledge-driven tasks. OpenIE, which does not require any relation schema but identifies a relational phrase to describe the relationship between a subject and an object, is being a trending topic of IE in sciences. The subjects, objects, and relations are often multiword expressions, which brings challenges for methods to identify the boundaries of the expressions given very limited or even no training data. In this work, we present a set of rules for extracting structured information based on dependency parsing that can be applied to any scientific dataset requiring no expert's annotation. Results on novel datasets show the effectiveness of the proposed method. We discuss negative results as well.**

***Index Terms*—Information extraction, Scientific knowledge, Dependency parsing.**

## I. Introduction

Mining structured information from scientific literature is important for many down-stream knowledge-driven tasks such as knowledge graph construction, population, and completion, as well as document retrieval, knowledge discovery, and hypothesis generation. In this work, we aim at this goal whose input and expected output can be illustrated by the example in Figure 1. The expectation has the following aspects, which leads the work to be *unsupervised OpenIE* supported by a set of multi-round parsing-based multiword rules. Let's look at the bottom box in the figure First, we argue that concept and relation recognition need to be *open schema* because (a) no existing concept database (e.g., those in BioNLP, UMLS, MeSH) has all the concepts in these examples and (b) no existing relation schema has the relations such as "detect", "bind", and "trigger". So the target form of tuples has subject, object, and relational phrase such as "strongly binding" and "only in". Second, the subjects and objects can be either concepts or *attributes of concepts* such as promoter and expression of miRNA (e.g., "KLF13::promoter" and 'KLF13::expression"), activation of specific pathways, and regulators of inhibitors. Third, the tuples can have a role of either fact or *condition of fact*. Conditions play an essential role in scientific statements: without the conditions that were precisely given by scientists, the facts might no longer be valid. For example, the expression of KLF13 must be "only in" sensitive PDXs, which should not be removed or ignored in the output.

As shown in the figure, the process of extracting the tuples would have to identify multiple types of structured information such as acronyms, noun phrases (NPs, as concepts and attributes) and verbal phrases (VPs, as relations) that are often *multiword expressions*. Acronym taggers and phrase chunkers are applied and studied: we find that (a) the extraction of one structure type can reduce the complexity of the sentence and (b) to accurately extract the multiwords, the dependency parsing tree can suggest ideas for the next round.

First, the dependency tags on the tree (e.g., "compound", "appos") help find acronyms even when the syntax is complex, like "ChIP" for "chromatin immunoprecipitation". Then the parenthesis are removed to simplify the sentence.

Second, the second round of parsing can help detect noun phrases. When the noun phrases are replaced by "NP$x$" ($x = 1, 2...$), the sentence can be significantly shortened, e.g., "NP1 detected NP2 strongly binding at the NP3..."

Third, the third parsing identifies the relationship between the noun phrases: one NP is an attribute of the other NP, e.g, "promoter of KLF13"; or a NP should be separated into two, one for concept and the other for attribute, e.g., "KLF13 promoter". Then we replace the concept or concept's attribute, which is a subject or object, by "CA$x$" ($x = 1, 2...$) so that all the remaining words in the sentence would not be related to the subjects or objects.

Fourth, another round of parsing can identify verbal phrases that might not be included in any existing relation schema. The phrases preserve adverbial modifiers that later can be used for learning and inference in down-stream tasks. The verbal phrases are tagged as "VP$x$" ($x = 1, 2...$).

Lastly, we perform parsing on the simplest variant of the sentence we can have. We design a set of rules to extract fact and condition tuples from the parsing tree. A lot of parsing and reorganizing the tree was done so that the final tree would be simpler, and patterns would be more common, when the initial trees could be quite large and the patterns were lengthy.

**Main contributions:** We build the multi-round parsing-based scientific OpenIE method and collected novel biomedical literature datasets, COVID-19 and miRNA for epithelial cancer. Each dataset has more than 50 sentences that were annotated by five experts in the corresponding domains. Our method performs much better than existing unsupervised OpenIE methods, however, we still identify quite a few false predictions. Finally, we study negative results and make error analysis.

## II. Related Work

In this section, we review recent papers on three topics related with our work: scientific information extraction, OpenIE, and dependency parsing.

978-1-6654-3858-2/21/$31.00 ©2021 IEEE
DOI 10.1109/ICKG52313.2021.00051



Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on June 22,2022 at 00:03:13 UTC from IEEE Xplore. Restrictions apply.

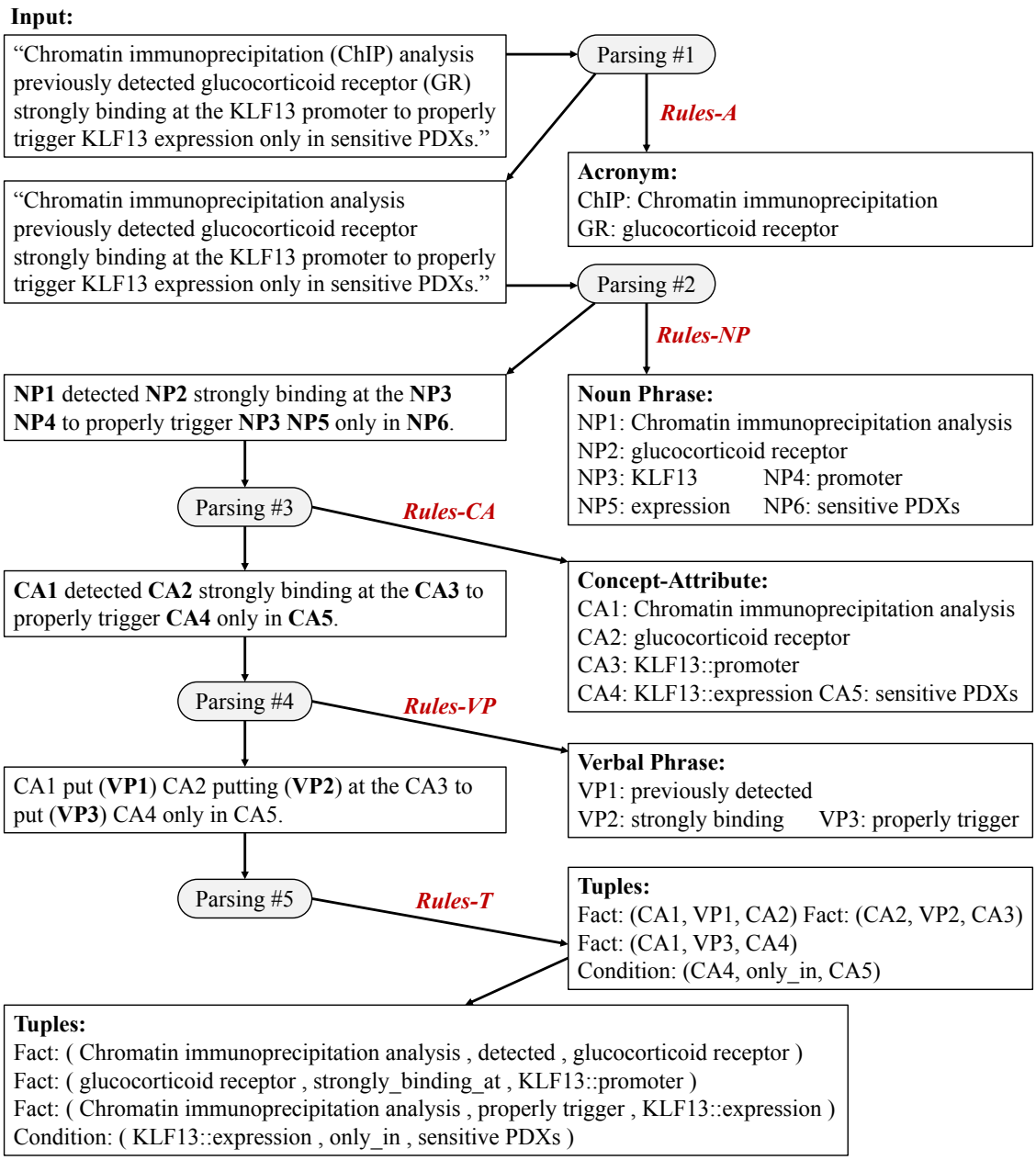


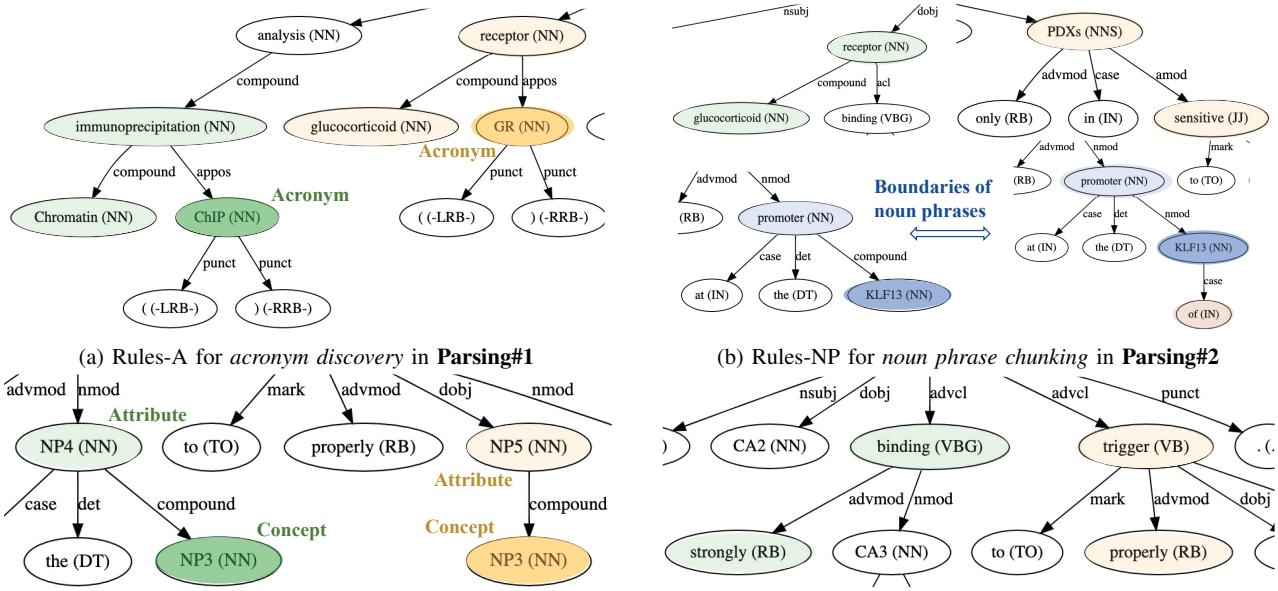

Fig. 1: To turn scientific statement sentences (unstructured) into fact/condition tuples (structured), we perform multi-round parsing and design a set of rules to extract the structures step by step.

### A. *Scientific Information Extraction*

Pre-trained scientific language models such as SciBERT [1] and BioBERT [2] have been widely used to perform specific information extraction tasks after being trained (called "fine-tuned") with a set of annotations on the tasks. [3] identified and classified scientific keyphrases with BERT models. [4] detected named entities from electronic medical record with BioBERT. [5] used BioBERT to detect health-related messages from Twitter data. [6] detected BioBERT-based models to identify conditional statements and build biomedical knowledge graphs. There has not been much published work on unsupervised methods for scientific information extraction [7]. Scientific knowledge graphs are constructed from the textual information [8] where a specific type of graphs is faceted concept hierarchy [9]. Comprehensive scientific knowledge graphs represent conditional statements in structured form [10], which has shown the advantage of knowledge completeness for scientific literature search [11]. Zeng et al. used SciBERT and BioBERT to automatically build scientific taxonomies by generating concept names word by word [12].



Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on June 22,2022 at 00:03:13 UTC from IEEE Xplore. Restrictions apply.




(a) Rules-A for *acronym discovery* in **Parsing#1**

(b) Rules-NP for *noun phrase chunking* in **Parsing#2**

(c) Rules-CA for *concept/attribute recognition* in **Parsing#3**

(d) Rules-VP for *relational phrase chunking* in **Parsing#4**

Fig. 2: Multiword rules for extracting preliminary structures (prior to tuples) in multi-round dependency parsing.

### B. Open Information Extraction

OpenIE techniques used to leverage linguistic structures such as dependency relations [13], [14], clauses [15], and numerical rules [16]. [17] used meta textual patterns to discover open-schema facts in biomedical literature. [18] developed a supervised learning framework based on sequence labeling to leverage deep learning for OpenIE. [19] used OpenIE to construct open-schema knowledge graphs. Conjunctive sentences [20] and span models [21] were created for OpenIE, respectively. Recently, iterative learning methods become popular in OpenIE [22], [23]. Most existing scientific or biomedical IE methods use fixed relational schema to detect pair-wise relational facts. We argue the limitations in the introduction section and address them with OpenIE.

### C. Dependency Parsing

A dependency parser analyzes the grammatical structure of a sentence, establishing relationships between "head" words and words which modify those heads [24]. It generates a dependency parse tree that is a directed graph which has the following features: Root node can only be head in head-dependent pair. Nodes except Root should have only one parent/head. A unique path should exist between Root and each node in the tree [25]. Recently, Allen Institute for AI developed SciSpaCy that contains fast and robust models for biomedical natural language processing [26]. It can support dependency parsing with three model options: small, medium, and large [27]. The accuracy on part-of-speech tagging is as high as 0.9891. However, the F1 score on named entity recognition is 0.67–0.70, and thus, the performance on information extraction could even worse. We develop a pipeline to bridge the gap with multi-round parsing and multiword rules.

## III. Proposed Approach

In this section, we present our pipeline of *five* rounds of dependency parsing and information extraction as well as the multiword rules for the tasks.

### A. Dependency parser

SciSpaCy is a Python package containing SpaCy models for processing biomedical text. The parsing model is named "en_core_sci_lg" with a description as "a full SpaCy pipeline for biomedical data with a larger vocabulary and 600k word vectors". It creates a dependency parse tree where the nodes (i.e., tokens) have part-of-speech tag and the links between head-dependent pairs have dependency tag (see Figure 2).

### B. Acronym discovery: Rules-A in Parsing#1

Acronyms can be identified based on the initials such as "glucocorticoid receptor" as "GR". However, sometimes the patterns could be complex like "ChIP" for "chromatin immunoprecipitation". The parse tree suggests the boundaries of the concept's name (see Figure 2a).

### C. Noun phrase chunking: Rules-NP in Parsing#2

Noun chunker is to group adjacent adjectives and nouns together to form noun phrases. They consisted of nouns and adjectives that together formed one phrase that could be referred to as a whole. This simplified the tree and reduced noise. For example, the adjective "sensitive (JJ)" can be put together with the noun "PDXs (NNS)" to form a phrase node "sensitive PDXs (NP)" to simplify the parse tree (see Figure 2b). The noun chunker leverages parsing patterns such as dependency "amod" between JJ and NNS, dependency "compound" between NN and NN, and numbers (labelled as CD) after NN (e.g., a nucleic



Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on June 22,2022 at 00:03:13 UTC from IEEE Xplore. Restrictions apply.

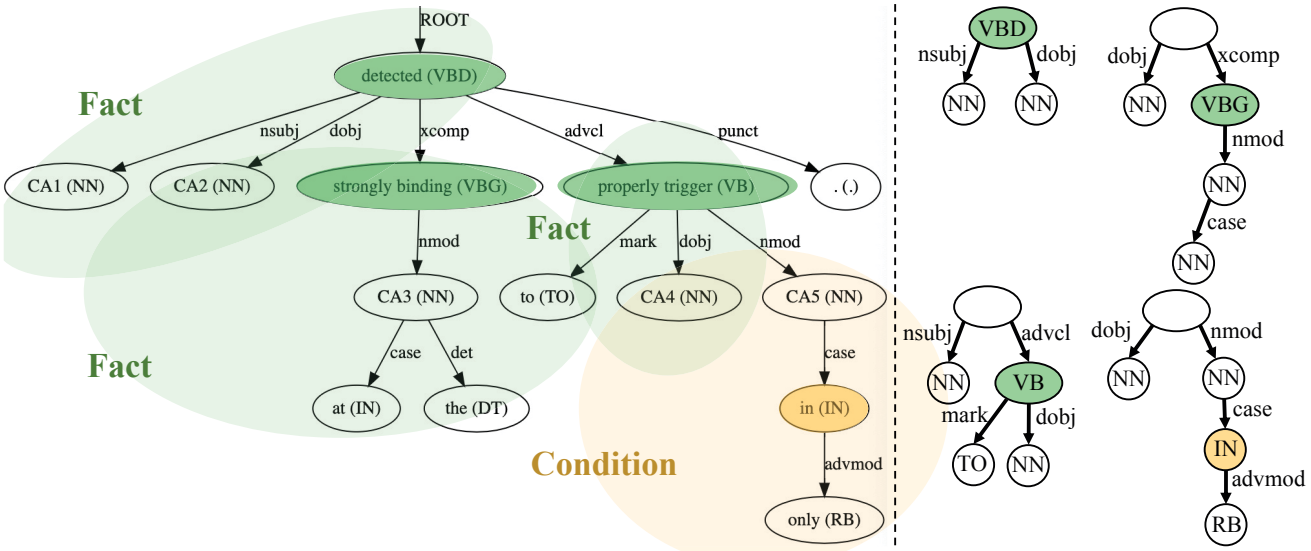



Fig. 3: Rules-T for *fact and condition tuple extraction* in **Parsing#5**

acid stain "Hoechst 33342"). One special case is that when the adjective is "all" or "no", it will be parsed as attribute instead of a modifier on a noun phrase. Another case is that a pair of noun phrases could be expressed in multiple kinds of patterns such as "A B" (e.g., "KLF13 promoter") and "B of A" (e.g., "promoter of KLF13"). Given two nouns "A B", they could be connected by tag "compound"; only by leveraging "B of A" from sentences elsewhere, we could identify the separating boundaries of the noun phrases in "A B".

### *D. Concept and attribute recognition: Rules-CA in Parsing#3*

After parsing, we remove DT tokens (e.g., "a", "an", and "the") from noun phrases (NPs) in order to further simply the tree. Then we identify NPs connected by the preposition "of (IN)". This link typically indicates that one NP is an attribute of the other NP (see Figure 2c). The concept and its attribute are merged into one single node. The merged node and individual concept node are labelled as "CA" (for concept or attribute).

### *E. Verbal (relational) phrase chunking: Rules-VP in Parsing#4*

A verb chunker is created to do similar simplification by grouping adverbs, helping verbs, and main verbs together to form a verbal phrase (VP) (see Figure 2d). Then an algorithm searches for a preposition connected to the verb under the "prep" dependency. If one is found, a noun phrase is searched for that is connected to the preposition under the "pobj" dependency. If one is found the preposition is attached at the end of VP. And if there is an extra adjective linked between the verb and the preposition, it will be put between the VP and preposition.

### *F. Fact and condition tuple extraction: Rules-T in Parsing#5*

An algorithm searches the parse tree for patterns or rules that could be used to form a tuple (see Figure 3). The tuples can be classified into fact tuples and condition tuples. For fact tuples, the first rule or pattern looks for verbs that have a CA as a subject ("nsubj" dependency) node as well as a CA as an object ("dobj" or "pobj" dependency), forming (subject, verb, object)-tuples. Each time a tuple is formed, the function also searches for additional VPs or NPs that were connected to the tuple with the "conj" dependency. If a connection was found, a new tuple would form, replacing the original CA with the new one that was connected with the "conj" dependency. The next rule iterates over all CAs in a sentence and searches for ones that have a VP node connected with the "relcl" dependency tag. It then checks if that verb node has a noun node connected with the object dependency tag.

For condition tuples, a rule looks for CAs that are connected through a preposition (IN). Each CA is iterated over and searching for preposition that linked to it. If a preposition was found, it is then iterated over to find another CA, forming (subject, preposition, object)-tuple. It should be noted that the preposition "of" is excluded because these has already parsed into noun phrases to form CAs. Another pattern forms general condition tuples if there is no object found. This happens when a verb phrase is found, and then a CA is found connected with the subject dependency "nsubj". The next step is to find an object connected to the verb, but if none is found a general tuple would form with a "NIL" object, as (subject, verb, NIL).

## IV. Experiments

In this section, we conduct experiments to demonstrate the effectiveness of our multi-round parsing-based method, compared with existing OpenIE methods. We will first present the dataset collection, baseline methods, and evaluation metrics, and then experimental results and our analysis.

### *A. Datasets*

We recruited experts in the fields of target-directed miRNA degradation (TDMD) and myeloid-derived suppressor cell (MDSC) to annotate *three* datasets. Data statistics are given



Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on June 22,2022 at 00:03:13 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Three datasets annotated for Sci-OpenIE.

| Dataset | #Doc | #Sent. | #Facts | #Cond. |
|---|---|---|---|---|
| TDMD-50 | 16 | 52 | 139 | 140 |
| COVID-200 | 21 | 216 | 349 | 355 |
| MDSC-600 | 25 | 612 | 945 | 1,164 |

TABLE II: Effectiveness of the proposed method: Compared with the OpenIE baseline methods, the proposed method achieves high Micro F1 and Macro F1 scores.

| | TDMD-50 | | COVID-200 | | MDSC-600 | |
|---|---|---|---|---|---|---|
| | Micro F1 | Macro F1 | Micro F1 | Macro F1 | Micro F1 | Macro F1 |
| IMoJIE | 0.034 | 0.066 | 0.033 | 0.067 | 0.026 | 0.051 |
| OpenIE6 | 0.040 | 0.076 | 0.045 | 0.085 | 0.029 | 0.055 |
| SOIE | 0.089 | 0.168 | 0.058 | 0.111 | 0.054 | 0.099 |
| **Ours** | **0.199** | **0.313** | **0.127** | **0.221** | **0.102** | **0.179** |

TABLE III: Ablation study: Compared with the variants of the proposed method, the multiple rounds of parsing and corresponding rules are effective.

| | TDMD-50 | | COVID-200 | | MDSC-600 | |
|---|---|---|---|---|---|---|
| | Micro F1 | Macro F1 | Micro F1 | Macro F1 | Micro F1 | Macro F1 |
| Parsing#1–#5 | 0.199 | 0.313 | 0.127 | 0.221 | 0.102 | 0.179 |
| w/o Parsing#2 (NP) | 0.186 | 0.292 | 0.112 | 0.183 | 0.099 | 0.172 |
| w/o Parsing#4 (VP) | 0.161 | 0.255 | 0.108 | 0.182 | 0.090 | 0.156 |
| w/o Parsing#5 (Tuple) | 0.162 | 0.262 | 0.110 | 0.187 | 0.090 | 0.161 |

in Table I. About 16–25 documents were annotated for each dataset. The datasets are named according to the research domain of the literature and the number of annotated sentences. The smallest dataset TDMD-50 has only *statement sentences* that were recognized by the experts. The medium-size COVID-200 was from the well-cited COVID-19 literature and only *abstracts* were annotated. And the largest MDSC-600 annotated every sentence in the *full-text*. The annotators follow a codebook about Scientific OpenIE to create fact and condition tuples. We observe that a sentence in abstract or full-text creates 1.5–1.6 facts and 1.6–1.9 conditions (see COVID-200 and MDSC-600); a statement sentence creates 2.7 facts and 2.7 conditions (see TDMD-50).

### B. Baseline Methods

We compare with the following OpenIE methods:

- SOIE [14]: It was developed by a Stanford team and was one of the most popular OpenIE tools. It leveraged linguistic structures in dependency parse trees.
- OpenIE6 [22]: It performed iterative grid labeling and coordination analysis to improve the performance of OpenIE.
- IMoJIE [23]: It adopted an iterative memory-based framework that could produce the next extraction conditioned on all previously extracted tuples.

To compete with these methods in a fair environment, Parsing#1 was performed to remove parentheses about acronyms in their input; and Parsing#3 was performed to post-process their results for concept-attribute structures.

As we developed multiword rules for Parsing#2 (noun phrase chunking), Parsing#4 (verbal phrase chunking), and Parsing#5 (tuple extraction), we will investigate the effect of each component in ablation studies by *replacing* one parsing round and the corresponding rules by *phrase chunkers and tuple extraction algorithms* in SciSpaCy and SOIE to create method variants.

### C. Evaluation Metrics

For each method, we report final F1 scores using precision and recall [28], [29], [30]. Precision represents the percentage of tuples that were correct and recall represents what percentage of possible tuples were found. This F1 was applied with a macro and micro method. Macro F1 treats each sentence as a separate instance and then averages all sentences scores together. Micro F1 treats all of the sentences as one big set and takes the F1 of everything as a whole. Both are useful. Micro accounts for certain sentences being more difficult than others and doesn't favor sentences with a small number of tuples or easier tuples. Macro gives a better idea of how effective the algorithm is on a random single sentence.

### D. Experimental Results

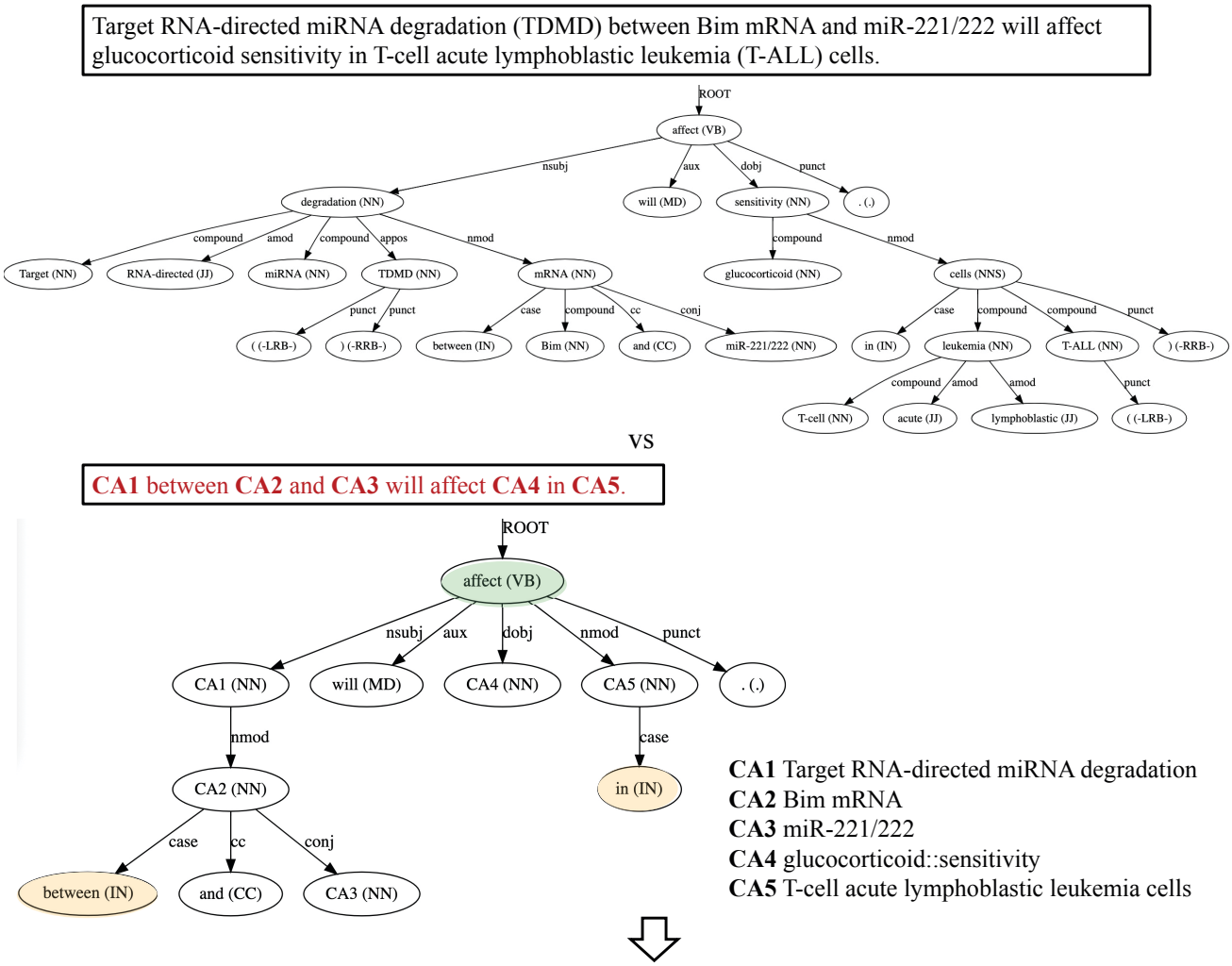
We present and analyze the results in the experiments to answer the following questions: (1) Is the proposed method more effective than the OpenIE baseline methods? (2) Are the technical components, i.e., multi-round parsing and multiword rules, useful for the task of Scientific OpenIE?

*1) Effectiveness: Compared with OpenIE Baseline Methods:* Table II presents the results on three datasets that compare our method with three baseline methods.

We observe that SOIE performs the best among the baseline methods though the other two methods are deep learning-based. The main reason is that the rule-based SOIE can better be generalized to text from any scientific domains, while OpenIE6 and IMoJIE are not capable to extract the tuples accurately from the new scientific text – they were trained



Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on June 22,2022 at 00:03:13 UTC from IEEE Xplore. Restrictions apply.




| | Subject (CA) | Relation (VP) | Object (CA) |
|---|---|---|---|
| **Fact** | Target RNA-directed miRNA degradation | affect | glucocorticoid::sensitivity |
| **Condition** | Target RNA-directed miRNA degradation | between | Bim mRNA |
| **Condition** | Target RNA-directed miRNA degradation | between | miR-221/222 |
| **Condition** | glucocorticoid::sensitivity | in | T-cell acute lymphoblastic leukemia cells |

Fig. 4: Case study: The multi-round parsing process can significantly simplify the structure of "sentence".

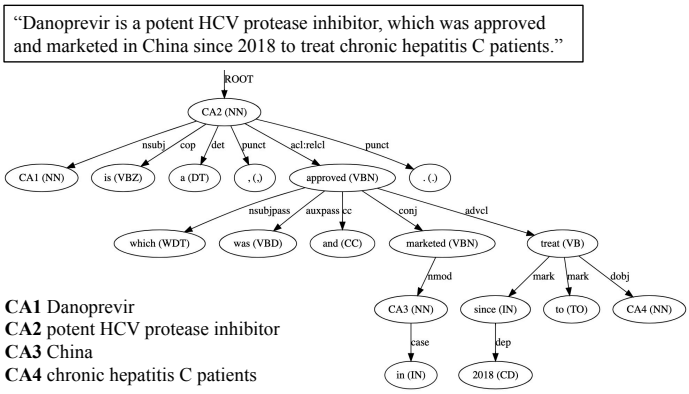
on the annotated datasets in their original papers. The second reason that accounts for this observation is that SOIE can be easily adapted to SciSpaCy dependency parse tree by modifying the patterns used in Parsing#1 and Parsing#3 (concept-attribute structures).

By comparing our method's results with the baselines', we observe that ours perform significantly better with higher Micro F1 and Macro F1 scores than SOIE. This is because our method captures more correct tuples by iterating through every subtree with a noun or verb as the root node in the dependency tree and matching them with designed patterns, while being able to exclude disguised wrong tuples with effective noun and verb phrase chunking strategies. In addition, our built-in steps for concept and attribute recognition account for its better performance for tuple extraction compared with traditional triple extraction methods.

We observe that the Micro F1 score is much lower than Macro F1 score. This is because the Micro F1 is affected heavily by the sentences that have many annotated tuples but very few extracted tuples, while the Macro F1 takes each



Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on June 22,2022 at 00:03:13 UTC from IEEE Xplore. Restrictions apply.




**Extraction:**

| Subject (CA) | Relation (VP) | Object (CA) |
|---|---|---|
| Danoprevir | is | potent_HCV_protease_inhibitor |

**Expected results:**

| Subject (CA) | Relation (VP) | Object (CA) |
|---|---|---|
| Danoprevir | is | HCV::potent_protease_inhibitor |
| Danoprevir | was_approved_in | China |
| Danoprevir | was_marketed_in | China |
| Danoprevir | was_approved_since | 2018 |
| Danoprevir | was_marketed_since | 2018 |
| Danoprevir | treat | patients:chronic_hepatitis_C |

Fig. 5: An negative example: The highlighted components in the expected results could not be extracted by the current design of extraction rules.

sentence with equal weight and ignores the imbalance of tuple numbers. And we observe that the F1 score is lower when the dataset is bigger. This is because the larger dataset contains more various sentences, such as long sentences with multiple clauses and captions of figures and tables. The existence of various scientific expressions and special characters makes the tuple extraction more difficult.

*2) Ablation Study: Compared with Method Variants:* Table III presents the results to compare the proposed method with the methods that replace particular components with conventional tools.

We observe that (a) the Micro F1 scores would be relatively 6.5%, 11.8%, and 2.9% higher on TDMD-50, COVID-200, and MDSC-600, respectively; and (b) the Macro F1 scores would be relatively 6.7%, 17.2%, and 3.9% higher, if we enabled Parsing#2 and Rules-NP for noun phrase chunking. This is because the noun-chunking feature of the SpaCy library usually collected all the proper nouns, but did not always put them in the correct order. A common issue with the COVID data was that numbers following a word such as, "coronavirus 2", would not always be grouped together. These functions fine-tuned the noun-chunking by reordering and editing the noun-phrases. Other examples included articles such as "these" and "no." "These" was removed in instances where it began a noun-phrase, and no was changed from an adjective to an attribute (no_symptoms $\rightarrow$ symptoms::no).

With Parsing#4 and Rules-VP for verbal phrase chunking, we observe that (a) the Micro F1 scores would be relatively 19.1%, 15.0%, and 11.8% higher and (b) the Macro F1 scores would be relatively 18.5%, 17.6%, and 12.8% higher, on the three datasets respectively. This is because the verb-parsing function did not always pick up every word that should be included in the tuple. These functions and rules checked for the most common instances of the verb-phrase being incorrect and adjusted them. Most of the time, they were only missing one word that needed to be added to the phrase to make it correct. In the example below:

"Cases quickly grew and spread throughout the hospital."

the adverb "quickly" should be added to both verbs (quickly_spread_throughout and quickly_increased). Also, the preposition "throughout" should be added to the VP for the first tuple. With the addition of these rules, both of these things now occur.

With Parsing#5 and Rules-T for tuple extraction, we observe that (a) the Micro F1 scores would be relatively 18.6%, 13.4%, and 11.8% higher and (b) the Macro F1 scores would be relatively 16.3%, 15.4%, and 10.1% higher, on the three datasets respectively. This is because the tuple generation was adjusted to stricter rules. The NIL tuples were generated whenever a verb and and subject were found without an object. The tuple generation also better accounted for compound NPs and VPs. The rules were written to maximize precision. This is because rather than outputting all possible relations for NPs and VPs in a sentence, the phrases in a tuple had to all be directly related in a strict format.

### *E. Case Study*

Figure 4 presents a case where the multi-round parsing process significantly simplified the structure of "sentence". Fact and condition tuples could be easily extracted from the simplest (final) parse tree. The tuples can later be used for knowledge graph construction, knowledge discovery, exploration, inference, and hypothesis generation.

### *F. Discussion on Negative Results*

Figure 5 presents an instance of negative results. The first prominent issue was the depth at which each sentence was searched. As shown in the figure, the correct tuples formed a very complex pattern. In order to find such a pattern, the search depth must increase greatly. This was a very common occurrence, especially with complex sentences. Moving forward this could be expanded, but it would also require further optimization, because adding more depth to the search will increase the run time.

The second issue is that the pre-processing did not correctly identify each CA and VP. This is an issue with the SpaCy



Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on June 22,2022 at 00:03:13 UTC from IEEE Xplore. Restrictions apply.

noun-chunking feature as well as the verb-chunking function. This was a pretty common issue in sentences that had multiple parentheses and numbers within it, such as this one. Moving forward, the processing and finding of CAs and VPs could be improved so that more would be recognized before converting the tree to tuples.

## V. Conclusions

In this work, we presented a novel pipeline that has multiple rounds of dependency parsing with a set of rules for extracting different types of structured information. This pipeline was able to extract factual and conditional information from scientific literature in an OpenIE manner. It did not require any relation schema or annotated data for model training. The subjects, objects, and relations are often multiword expressions that need multiword rules for accurate extraction. Results on novel datasets showed the effectiveness of this method. We discuss negative results as well.

## Acknowledgment

This research was supported by National Science Foundation (NSF) Grants IIS-1849816 and CCF-1901059.

## References

[1] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3606–3611.

[2] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[3] S. Park and C. Caragea, "Scientific keyphrase identification and classification by pre-trained language models intermediate task transfer learning," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 5409–5419.

[4] X. Yu, W. Hu, S. Lu, X. Sun, and Z. Yuan, "Biobert based named entity recognition in electronic medical record," in *2019 10th International Conference on Information Technology in Medicine and Education (ITME)*. IEEE, 2019, pp. 49–52.

[5] Y. Bai and X. Zhou, "Automatic detecting for health-related twitter data with biobert," in *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, 2020, pp. 63–69.

[6] T. Jiang, Q. Zeng, T. Zhao, B. Qin, T. Liu, N. Chawla, and M. Jiang, "Biomedical knowledge graphs construction from conditional statements," *IEEE/ACM transactions on computational biology and bioinformatics*, 2020.

[7] S. A. Salloum, M. Al-Emran, A. A. Monem, and K. Shaalan, "Using text mining techniques for extracting information from research articles," in *Intelligent natural language processing: Trends and Applications*. Springer, 2018, pp. 373–397.

[8] M. Jiang and J. Shang, "Scientific text mining and knowledge graphs," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3537–3538.

[9] Q. Zeng, M. Yu, W. Yu, J. Xiong, Y. Shi, and M. Jiang, "Faceted hierarchy: A new graph type to organize scientific concepts and a construction method," in *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, 2019.

[10] T. Jiang, T. Zhao, B. Qin, T. Liu, N. V. Chawla, and M. Jiang, "The role of" condition" a novel scientific knowledge graph representation and construction model," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1634–1642.

[11] T. Jiang, Z. Zhang, T. Zhao, B. Qin, T. Liu, N. V. Chawla, and M. Jiang, "Ctga: graph-based biomedical literature search," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2019, pp. 395–400.

[12] Q. Zeng, J. Lin, W. Yu, J. Cleland-Huang, and M. Jiang, "Enhancing taxonomy completion with concept generation via fusing relational representations," *arXiv preprint arXiv:2106.02974*, 2021.

[13] F. Wu and D. S. Weld, "Open information extraction using wikipedia," in *Proceedings of the 48th annual meeting of the association for computational linguistics*, 2010, pp. 118–127.

[14] G. Angeli, M. J. J. Premkumar, and C. D. Manning, "Leveraging linguistic structure for open domain information extraction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 344–354.

[15] L. Del Corro and R. Gemulla, "Clausie: clause-based open information extraction," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 355–366.

[16] S. Saha, H. Pal *et al.*, "Bootstrapping for numerical open ie," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 317–323.

[17] X. Wang, Y. Zhang, Q. Li, Y. Chen, and J. Han, "Open information extraction with meta-pattern discovery in biomedical literature," in *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2018, pp. 291–300.

[18] G. Stanovsky, J. Michael, L. Zettlemoyer, and I. Dagan, "Supervised open information extraction," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 885–895.

[19] I. Muhammad, A. Kearney, C. Gamble, F. Coenen, and P. Williamson, "Open information extraction for knowledge graph construction," in *International Conference on Database and Expert Systems Applications*. Springer, 2020, pp. 103–113.

[20] S. Saha *et al.*, "Open information extraction from conjunctive sentences," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2288–2299.

[21] J. Zhan and H. Zhao, "Span model for open information extraction on accurate corpus," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 9523–9530.

[22] K. Kolluru, V. Adlakha, S. Aggarwal, S. Chakrabarti *et al.*, "Openie6: Iterative grid labeling and coordination analysis for open information extraction," *arXiv preprint arXiv:2010.03147*, 2020.

[23] K. Kolluru, S. Aggarwal, V. Rathore, S. Chakrabarti *et al.*, "Imojie: Iterative memory-based joint open information extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5871–5886.

[24] M. S. Rasooli and J. Tetreault, "Yara parser: A fast and accurate dependency parser," *arXiv preprint arXiv:1503.06733*, 2015.

[25] D. Gusfield, "Algorithms on stings, trees, and sequences: Computer science and computational biology," *Acm Sigact News*, vol. 28, no. 4, pp. 41–60, 1997.

[26] M. Neumann, D. King, I. Beltagy, and W. Ammar, "Scispacy: Fast and robust models for biomedical natural language processing," *arXiv preprint arXiv:1902.07669*, 2019.

[27] J. Kanerva, F. Ginter, and S. Pyysalo, "Dependency parsing of biomedical text with bert," *BMC bioinformatics*, vol. 21, no. 23, pp. 1–12, 2020.

[28] J. L. Martinez-Rodriguez, I. López-Arévalo, and A. B. Rios-Alvarado, "Openie-based approach for knowledge graph construction from text," *Expert Systems with Applications*, vol. 113, pp. 339–355, 2018.

[29] W. Léchelle and P. Langlais, "Revisiting the task of scoring open ie relations," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[30] S. Bhardwaj, S. Aggarwal, and M. Mausam, "Carb: A crowdsourced benchmark for open ie," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 6263–6268.



Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on June 22,2022 at 00:03:13 UTC from IEEE Xplore. Restrictions apply.