Multi-Input Multi-Output Sequence Labeling for Joint Extraction of Fact and Condition Tuples from Scientific Text

Tianwen Jiang[‡†,1], **Tong Zhao**[†], **Bing Qin**[‡], **Ting Liu**[‡], **Nitesh V. Chawla**[†], **Meng Jiang**[†]
[‡]Harbin Institute of Technology, Harbin, Heilongjiang, China
[†]University of Notre Dame, Notre Dame, Indiana, USA
[‡]{twjiang,bqin,tliu}@ir.hit.edu.cn
[†]{tzhao2,nchawla,mjiang2}@nd.edu

Abstract

Condition is essential in scientific statement. Without the conditions (e.g., equipment, environment) that were precisely specified, facts (e.g., observations) in the statements may no longer be valid. Existing ScienceIE methods, which aim at extracting factual tuples from scientific text, do not consider the conditions. In this work, we propose a new sequence labeling framework (as well as a new tag schema) to jointly extract the fact and condition tuples from statement sentences. The framework has (1) a multi-output module to generate one or multiple tuples and (2) a multi-input module to feed in multiple types of signals as sequences. It improves F1 score relatively by 4.2% on BioNLP2013 and by 6.2% on a new bio-text dataset for tuple extraction.

1 Introduction

Conditions such as environment and equipment provide validation supports for facts, while the facts focus on scientific observation and hypothesis in scientific literature (Miller, 1947). Existing ScienceIE methods, which extract (subject, relational phrase, object)-tuples from scientific text, do not distinguish the roles of fact and condition. Simply adding a tuple classification module has two weak points: (1) one tuple may have different roles in different sentences; (2) the tuples in one sentence have high dependencies with each other, for example, given a statement sentence in a biochemistry paper (Tomilin et al., 2016):

"*We observed that ... alkaline pH increases the activity of TRPV5/V6 channels in Jurkat T cells.*"

an existing system (Stanovsky et al., 2018) would return one tuple as below:

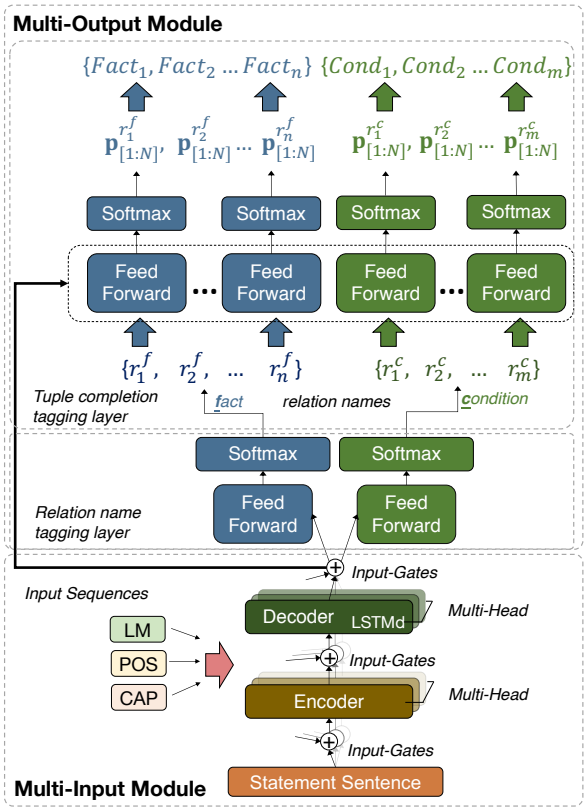


Figure 1: Our framework has two modules: (1) a multi-input module (bottom) based on a multi-head encoder-decoder model with multi-input gates; (2) a multi-output module (top) of a relation name tagging layer and a tuple completion tagging layer.

(alkaline_pH, increases, activity_of_TRPV5/V6 _channels_in_Jurkat_T_cells).

where (a) the object should just be the channel's activity and (b) the condition tuple (TRPV5/V6 _channels, in, Jurkat_T_cells) was not found. Note that the term "TRPV5/V6 _channels" is *not only* the concept in the fact tuple's object *but also* the condition tuple's subject.

In this work, we define the joint tuple extraction task as a *multi-output* sequence labeling problem. First, we create a new tag schema: Non-"O" tags

[1]This work was done when the first author was visiting the University of Notre Dame.

are formatted as "B/I-XYZ", where

- X ∈ {fact, condition};
- Y ∈ {1: subject; 2: relation; 3: object};
- Z ∈ {concept, attribute, relational phrase}.

Note that if Y="2" then Z="p". So, the number of non-"O" tags is 20. Now each fact/condition tuple can be represented as a tag sequence. Moreover, it is the first work in sequence labeling that concepts and attributes are separated. The fact tuple in the example will ideally be: (alkaline_pH, increases, {TRPV5/V6_channels : activity}).

Figure 1 shows our framework. Multiple tag sequences are generated after the LSTMd decoder, each of which represents a fact or condition tuple. This multi-output module has two layers: one is a relation name tagging layer that predicts the tags of relational phrases and determines the number of output sequences; the other is a tuple completion tagging layer that generates the tag sequences for completing the fact and condition tuples.

To address the challenge of modeling the complex tag schema, besides language model, we incorporate as much information as possible from upstream tools such as Part-of-Speech tagging (POS), Concept detection, Attribute name extraction, and Phrase mining (CAP). And we transform them into tag sequences as the model input. We observe strong dependencies between the token's POS/CAP tags and target tags. We appreciate the high accuracy of existing techniques making the multi-input sequences available for new datasets.

The multi-input multi-output sequence labeling framework is named as MIMO. Experiments demonstrate that it improves F1 score relatively by 6.2% over state-of-the-art models for tuple extraction on a new bio-text dataset we will introduce in the later section. When transferred to the BioNLP2013 dataset without additional training, it improves F1 score relatively by 4.2%. We apply MIMO to a large set of 15.5M MEDLINE papers and construct a knowledge graph: An example can be found in Figure 4.

2 A New Dataset

We built a system with GUI (Figure 2) to collect a new dataset for the joint tuple extraction purpose, named Biomedical Conditional Fact Extraction (BioCFE). Three participants (experts in biomedical domain) manually annotated the fact and condition tuples from statement sentences from 31 paper abstracts in the MEDLINE database. The annotation procedure took over 30 minutes on average for each paper. Here is a brief guide to the system. First, the users merged the token(s) into a span. Second, they gave a proper number of fact and/or condition tuple(s), where the proper number is not fixed but depends on the concrete sentence. Each tuple has five slots (subject's concept, subject's attribute, relation phrase, object's concept, and object's attribute). Third, they dragged the spans filling into the slots. If the three annotations are inconsistent, we filtered out the case. Eventually we have 756 fact tuples and 654 condition tuples from 336 annotated sentences. It is common to see one sentence having multiple facts and/or conditions, and actually 61%/52% statement sentences have more than one fact/condition tuples.

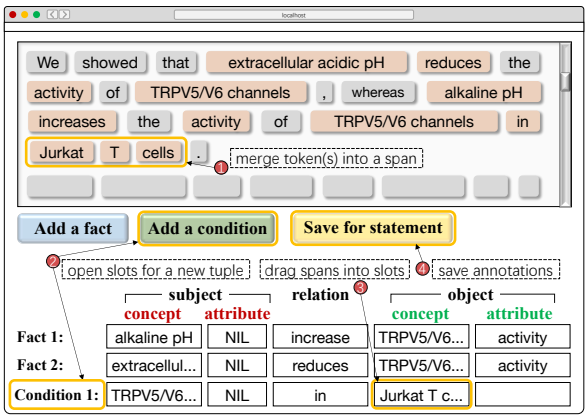


Figure 2: Annotation by four steps: (1) merge token(s) into a span; (2) make slots for a new tuple; (3) drag spans into the slots; (4) save annotations.

3 The Proposed Approach

Our approach has two modules: (1) a multi-input module that harnesses recent NLP development to process the text for input sequences from multiple tasks and feeds them into a multi-head encoder-decoder model with multi-input gates; (2) a multi-output module that generates multiple tuple tag sequences for fact and condition tuples, which consists of a relation name tagging layer and a tuple completion tagging layer, as shown in Figure 1.

3.1 The Multi-Input Module

Preprocessing for input sequences: Following fundamental NLP techniques have achieved high accuracy requiring no additional training with labeled data: Language Model (LM) (Howard and Ruder, 2018), POS (Labeau et al., 2015), CAP (Luan et al., 2018; Jiang et al., 2017; Shang

et al., 2018; Wang et al., 2018a). For any given input sentence, we tokenize it and represent each token by its word embedding (pre-trained GloVe vector in this paper). Then we get another three input sequences by the input sentence and the above three fundamental NLP techniques. (1) A pre-trained LSTM-based language model takes the sentence as input and returns semantic embedding sequence, where the dependencies between a token and its predecessors in distant contexts are preserved. (2) We employ NLTK tool to generate the POS tag sequence for the given sentence. The POS tag sequence indicates syntactic patterns of the words in a sentence, that is the dependencies between POS tags and output tags, like verbs (e.g., VBD) and predicates (e.g., B-f2p). (3) Multiple complementary IE techniques are used to detect concepts, attributes and phrases from the given sentences, being merged and resulting a CAP sequence. We make tags in the format of "B/I - c/a/p" for the tokens of concepts, attributes, and phrases.

Each sequence encodes a specific type of dependencies. A combination of multi-type dependencies learns the complicated dependencies on the 21 tuple tags better than any sole type. LM learns the dependencies between a token and its predecessors in distant contexts, which helps predict the position of subject, relation, and object. POS encodes the syntactic features of words. Dependencies between the POS tag and tuple tag (e.g., "VBD" and "B-f2p") can be modeled. We also spot high dependencies between the CAP tag and tuple tag. For example, the tokens of "B/I-c" (concept) and "B/I-a" (attribute) tags have high probability of being labeled as "B/I-XYc" and "B/I-XYa" in the output sequences, respectively.

Multi-head Encoder-Decoder: We investigate two neural models as encoder: one is bidirectional LSTM (BiLSTM), the other is the renown, bidirectional encoder representations from Transformers (BERT). We adopt a LSTM structure as the decoding layer (LSTMd) (Zheng et al., 2017). We observe that the input sequences may have different tag predictability on different sentences. For short sentences, POS and CAP are more useful (modeling local dependencies); for long sentences, LM is more effective (modeling distant dependencies). In order to secure the model's robustness on massive data, we apply a multi-head mechanism to the encoder-decoder model. Each head of the encoder-decoder is fed with one type of input sequence, and they are combined at the end of decoder layer. Thus, the tag prediction becomes more stable than using a simple encoder-decoder without the multi-head.

Multi-input gates: We adopt the multi-input gates in ResNet (He et al., 2016) to take the most use of the multi-input sequences. We add the gates to the input of BiLSTM or BERT encoder, the input of LSTMd decoder, and the multi-output module.

3.2 Multi-Output Module

We propose to generate multiple output sequences. As annotating multiple tuples from one sentence is common, a token may have different expected tags in the tuples. On BioCFE, we observe that 93.8% statement sentences make multiple tuples: 21.7% of the sentences have at least one token that appears in at least one fact tuple and at least one condition tuple, expecting tags "B/I-fYZ" and "B/I-cYZ"; 18.1% of the sentences have at least one token that appears in one condition tuple as a part of subject and in another condition tuple as a part of object, expecting tags "B/I-c1Z" and "B/I-c3Z". Therefore, we extend the typical one-output sequence labeling to a multi-output design.

Then what is the number of output sequences? We reveal the significant role of relation names in making tuples. If we tagged the relation names out, for each relation name, of tags beginning with "B-f2p" as a fact's and "B-c2p" as a condition's, the module would generate an output sequence, respectively. Then we extract all possible tuples, whose relation has been specified, from every output sequence. Two observations on the annotated data support this idea: We transform each of the 1,410 tuples into a tag sequence. For the same sentence, if the tuples' relation names are the same, we merge their tag sequences into one and then use the matching function in (Stanovsky et al., 2018) to recover the tuples. First, 0 token has conflicting tags among the 240 merged sequences. Second, the recovery has 0 missing or wrong tuple. So, generating one output sequence and completing the tuples per relation name is practical.

The multi-output module has two layers: one is a relation name tagging layer and the other is a tuple completion tagging layer.

Relation name tagging (RNT) layer: It consists of feed-forward neural networks (FFNs) and softmax layers. Decoded vectors are fed into the FFNs

and the softmax predict the probability distribution of tags on fact and condition, respectively:

$$\mathbf{p}_i^f = \text{softmax}(\text{FFN}_{RNT}^f(\mathbf{d}_i)), \quad (1)$$
$$\mathbf{p}_i^c = \text{softmax}(\text{FFN}_{RNT}^c(\mathbf{d}_i)). \quad (2)$$

where f is for fact and c for condition. $\mathbf{d}_i$ denotes the i-th token's vector given by the LSTMd.

Now we have two tag sequences, one for fact and the other for condition. As we have argued with one-output, extracting tuples from the "two-output" sequences cannot resolve the tag conflicts, either. Here we extract only the relation names: $\{r_1^f, r_2^f, \cdots, r_n^f\}$ denotes the n relation names (beginning with "B-f2p" tag) in fact tuples and $\{r_1^c, r_2^c, \cdots, r_m^c\}$ denotes the m relation names (beginning with "B-c2p" tag) in condition tuples.

Tuple completion tagging (TCT) Layer: This layer predicts n fact tag sequences and m condition tag sequences. Each sequence is generated by a FFN and a softmax layer. The FFN obtains the relation name from the RNT layer. The FFN's input also includes the token's vectors from the encoder-decoder model of the multi-input module.

Here we take condition sequences as an example to describe the details of the method. When predicting the j-th tag sequence, we define the position embedding of the i-th token as follows, representing the relative position to the j-th relation name's tag "B-c2p":

$$\mathbf{v}_{i,j}^c = g_{\text{emb}}(r_j^c, i). \quad (3)$$

Thus, the tag probability distributions of the i-th token in the condition tag sequences are:

$$\begin{cases} \mathbf{p}_i^{(r_1^c)} = \text{softmax}(\text{FFN}_{TCT}^c(\mathbf{v}_{i,1}^c + \mathbf{d}_i)), \\ \mathbf{p}_i^{(r_2^c)} = \text{softmax}(\text{FFN}_{TCT}^c(\mathbf{v}_{i,2}^c + \mathbf{d}_i)), \\ \cdots, \\ \mathbf{p}_i^{(r_m^c)} = \text{softmax}(\text{FFN}_{TCT}^c(\mathbf{v}_{i,m}^c + \mathbf{d}_i)). \end{cases} \quad (4)$$

Similarly, we have the following tag distributions for the i-th token in the fact tag sequences:

$$\begin{cases} \mathbf{p}_i^{(r_1^f)} = \text{softmax}(\text{FFN}_{TCT}^f(\mathbf{v}_{i,1}^f + \mathbf{d}_i)), \\ \mathbf{p}_i^{(r_2^f)} = \text{softmax}(\text{FFN}_{TCT}^f(\mathbf{v}_{i,2}^f + \mathbf{d}_i)), \\ \cdots, \\ \mathbf{p}_i^{(r_n^f)} = \text{softmax}(\text{FFN}_{TCT}^f(\mathbf{v}_{i,n}^f + \mathbf{d}_i)), \end{cases} \quad (5)$$

where $\mathbf{v}_{i,j}^f$ is the position embedding of the i-th token in the j-th fact sequence, representing the relative position to the relation name's tag "B-f2p".

Finally, we apply the matching function in (Stanovsky et al., 2018) to complete and extract the tuples (i.e., the concepts and/or attributes in the subjects and objects) for each output sequence.

3.3 Loss Function and Training

Given a sentence s, the loss function of the relation name tagging layer can be written as below:

$$\ell_{RNT}^s = -\sum_{i=1}^{N^s}(\log(p_{i,y_i^f}^f) + \log(p_{i,y_i^c}^c)), \quad (6)$$

where $\mathbf{p}_{i,y}^f$ and $\mathbf{p}_{i,y}^c$ are the probability of predicting y as the tag of the i-th token in the fact and condition tag sequences, respectively. y_i^f and y_i^c are the observed tag of the i-th token in the fact and condition tuple, respectively. N^s is the length of the sentence s.

The loss function of the tuple completion tagging layer is consisted of two parts, loss on fact tuples and loss on condition tuples:

$$\begin{cases} \ell_{TCT}^s = \ell_{fact}^s + \ell_{cond.}^s, \\ \ell_{fact}^s = -\sum_{i=1}^{N^s}\sum_{j=1}^{n}\log(p_{i,y_{i,j}^f}^{(r_j^f)}), \\ \ell_{cond.}^s = -\sum_{i=1}^{N^s}\sum_{j=1}^{m}\log(p_{i,y_{i,j}^c}^{(r_j^c)}), \end{cases} \quad (7)$$

where n and m are the number of fact and condition tag sequences for the sentence s, respectively. $p_{i,y_{i,j}^f}^{(r_j^f)}$ and $p_{i,y_{i,j}^c}^{(r_j^c)}$ are the probability of predicting $y_{i,j}^f$ and $y_{i,j}^c$ as the tag of the i-th token in the j-th fact and condition tag sequence, respectively.

The overall loss function for optimization is:

$$\ell = \ell_{RNT} + \ell_{TCT} = \sum_{s\in\mathcal{S}}(\ell_{RNT}^s + \ell_{TCT}^s), \quad (8)$$

where $\mathcal{S}$ is the set of statement sentences.

Training details: On one hand, Equations (6) and (7) show that the error signal can be propagated from the RNT/TCT layers to the encoder-decoder model. On the other hand, the RNT layer specifies the relation names, or say, the tokens that have tags "B/I-f2p" and "B/I-c2p" for each tag sequence in the TCT layer. So we cannot have smooth gradients for back propagation from the TCT layer to the RNT layer. So, in order to have good learning effectiveness, the quality of predicting relation names has been secured beforehand. We pre-train the RNT layer with the multi-input module till the relation name's tag prediction achieves a higher-than-0.8 F1 score. Then we plug the TCT layer onto the RNT layer and train the entire framework to generate the multi-output tag sequences.

4 Experiments

We evaluate the performance of condition/fact tag prediction and tuple extraction by the proposed MIMO model, its variants, and state-of-the-art models on the newly annotated BioCFE dataset and transferred to the BioNLP2013 dataset.

4.1 Experimental Setup

Datasets: Statistics of BioCFE has been given in the Section 2. Additionally, the attribute-related tags take 11.7% and 9.4% of non-"O" tags in fact and condition tuples, respectively. So, it is important to distinguish concept and attribute. To the best of our knowledge, it is the first time that conditional information was carefully annotated on biomedical literature.

We use the system in Figure 2 to annotate a subset of BioNLP2013 Cancer Genetics (CG) task dataset (Nédellec et al., 2013). We have 197 fact tuples and 173 condition tuples. We use this BioNLP dataset as an extra test set for task of fact and condition tuples extraction, but the model will not be trained on this dataset.

Validation: The ratio of training:validation:test is 60:8:32. For BioCFE, the evaluation set has 242 fact tuples and 209 condition tuples (on average from 108 sentences). We repeat five times, evaluate the performance, and report average results.

Evaluation metrics: For tag prediction, We use standard metrics, precision, recall, and F1 scores. We have similar observations on Micro F1 scores as Macro F1 scores, so we report Macro F1 only. For evaluating tuple extraction, we use pair-wise comparison to match the extracted and ground-truth tuples. We evaluate the correctness on the tuple's five slots using the same metrics.

Baselines: We compare with statistical sequence labeling methods: Structured Support Vector Machine (SVM) (Tsochantaridis et al., 2005) and Conditional random field (CRF) (Lafferty et al., 2001). We compare with a neural sequence labeling method, BiLSTM-LSTMd (Zheng et al., 2017). We replace its encoder with BERT (Devlin et al., 2018) to make it a more competitive baseline. We also compare against two renown OpenIE systems, Stanford OpenIE (Angeli et al., 2015) and AllenNLP OpenIE (Stanovsky et al., 2018) followed by a condition/fact classification.

We enhance statistical sequence labeling models with multi-input signals for fairness, and train them for fact tuple and condition tuple extraction separately. In the neural baselines (BiLSTM-LSTMd and BERT-LSTMd), fact extraction and condition extraction share the encoder-decoder model and use different, proper parameters in the linear-softmax layer.

Hyperparameters: The multi-input module has a BiLSTM/BERT encoder and a LSTM decoder. The word embeddings were obtained from GloVe (Pennington et al., 2014) with the dimension size d_{WE} = 50. The language model dimension size ns d_{LM} = 200. The size of POS tag embedding is d_{POS} = 6. The size of CAP tag embedding is d_{CAP} = 3. The number of LSTM units in the encoding layer is 300. The number of transformer units in the BERT encoding layer is 768.

4.2 Results on BioCFE

In this section, we present overall performance, ablation study, error analysis, and efficiency.

4.2.1 Overall Performance

Table 1 shows that the proposed multi-input multi-output sequence labeling model with a BERT encoder consistently performs the best over all the baselines on tag prediction and tuple extraction. Compared to BiLSTM-LSTMd, BiLSTM-based MIMO improves F1 score relatively by 7.1% on tag prediction and by 8.8% on tuple extraction; compared to BERT-LSTMd, BERT-based MIMO improve F1 by **4.7%** and **6.2%** on the two tasks, respectively. Apparently the BERT encoder significantly improves the performance (by 16.9–17.2% on tag prediction and 7.7–10.3% on tuple extraction). And the MIMO design can further improve it. Neural sequence labeling models perform better than OpenIE systems and statistical methods. Neural sequence labeling models are more adaptive to learning structures with the new tag schema. Open IE method plus a condition/fact classification is not effective.

Compared to BERT-LSTMd, the BERT-based MIMO improves precision and recall relatively by 8.3% and 1.3% on tag prediction; and relatively by 3.1% and 9.3% on tuple extraction, respectively. When the tags were more precisely predicted, the tuple's five slots would be more accurately filled, and we would have more complete tuples.

We also observe that the improvements on condition's tags/tuples are consistently bigger than the improvements on fact's tag/tuples. It demonstrates that the MIMO design recognizes the role of conditions in the statement sentences better.

Methods	Tag Prediction (%)			Tuple Extraction(%)		
	Prec.	***Rec.***	***F1 / F1***$_{Fact}$***, F1***$_{Cond.}$	***Prec.***	***Rec.***	***F1 / F1***$_{Fact}$***, F1***$_{Cond.}$
Allennlp OpenIE (Stanovsky et al., 2018)	-	-	-	42.60	38.22	40.29 / -, -
Stanford OpenIE (Angeli et al., 2015)	-	-	-	47.11	41.62	44.19 / -, -
Structured SVM (Tsochantaridis et al., 2005)	32.68	25.80	28.83 / 32.76, 24.71	47.62	46.15	46.87 / 45.01, 48.72
CRF (Lafferty et al., 2001)	60.07	41.92	49.37 / 56.23, 41.87	65.19	62.44	63.78 / 64.07, 63.44
BiLSTM-LSTMd (Zheng et al., 2017)	61.00	56.26	58.53 / 65.16, 51.78	71.57	66.55	68.97 / 69.51, 68.41
BERT-LSTMd	70.07	70.19	70.13 / 74.30, 65.88	78.64	73.67	76.08 / 76.14, 75.99
MIMO (BiLSTM based)	67.80	58.24	62.66 / 66.67, 58.58	75.35	74.67	75.01 / 74.91, 75.10
MIMO (BERT based)	**75.91**	**71.08**	**73.41 / 76.01**, **70.75**	**81.06**	**80.53**	**80.79 / 79.94**, **81.64**

Table 1: The proposed MIMO outperforms existing methods on tag prediction and tuple extraction in the BioCFE dataset. The MIMO with BERT-based encoder performs the best. Higher score performs better.

4.2.2 Ablation Study

Table 2 compares variants of the proposed model to evaluate the effectiveness of the following components: (1) *multi-input sequences*, such as none, or one (in LM, POS, and CAP), double combination, or triple combination; (2) *multi-input encoder model*, BiLSTM or BERT; (3) *multi-output module*, with the RNT layer only (generating one fact tag sequence and one condition tag sequence) or a combination of RNT and TCT layers (generating multiple sequences for each tuple type).

Multi-input sequences: When the choices of the encoder model and multi-output layers are specified, we observe that triple combination of input sequences performs better than double combinations and the double combinations win over the sole input. An additional sequence makes a relative F1 improvement by 1.0–2.4%. The triple combination improves F1 relatively by 3.2–4.1%. This demonstrates that the three types of input sequences encode *complementary information* for learning dependencies in the proposed tag schema.

First, the language model learns the dependencies between a token and its predecessors in distant contexts. Having the LM sequence recognizes subjects and objects relative to the relation names and reduces the false positives of "B/I-X1Z" and "B/I-X3Z". Second, the POS tag encodes the token's syntactic feature. Having the POS sequence improves the precision of tag prediction. For example, verbs and prepositions (e.g., "in", "during") often act as the relation name of facts and conditions, respectively; conjunction words (e.g., "that", "which") indicate subordinate clauses, so the noun phrase before the conjunction word is likely to be the subject of the tuple given by the clause. Third, the formerly-detected concepts, attribute names, and phrases are absolutely useful for tagging the slots of subjects and objects. In other words, the tags "B/I-c" and "B/I-a" in the CAP sequence are strongly associated with the target tags "B/I-XYc" and "B/I-XYa", respectively.

Encoder in the multi-input module: Comparing the middle three columns (BiLSTM-based encoder) and the right-hand three columns (BERT-based encoder), one can easily tell the significant improvement brought by the BERT model.

Layers in the multi-output module: If the multi-output models have both RNT and TCT layers, the F1 score is relatively 1.4–5.0% higher than the models that have the RNT layer only. Moreover, the recall is improved relatively by 1.5–9.0%. So the TCT layer, which generates multiple tag sequences for each type of tuple (i.e., fact and condition), plays a very important role in recognizing the multiple tuples from one statement sentence.

4.2.3 Error Analysis

Table 3 presents the confusion matrices made by the BERT-based MIMO on predicting non-"O" tags for facts and conditions, respectively. The columns are predicted tags and the rows are actual ones. Perfect results would be diagonal matrices.

We observe that the numbers at the diagonal are consistently bigger than the numbers on the corresponding row and column. The accuracy scores are 0.905 for predicting fact tags and 0.908 for predicting condition tags. Of the 182 actual "B-f2p", the model predicted that 175 were "B-f2p"; of the 186 actual "B-c2p", it predicted that one was "I-c1c" and one was "I-c3c". It demonstrates the high accuracy (0.961 and 0.989) of extracting relation names for multi-output generation.

Settings				BiLSTM-based Encoder (%)			BERT-based Encoder (%)		
LM	**POS**	**CAP**	**MO**	***Prec.***	***Rec.***	***F1 / F1$_{Fact}$, F1$_{Cond.}$***	***Prec.***	***Rec.***	***F1 / F1$_{Fact}$, F1$_{Cond.}$***
			✗	71.57	66.55	68.97 / 69.51, 68.41	78.64	73.67	76.08 / 76.14, 75.99
✔			✗	72.84	67.36	69.99 / 69.22, 70.75	79.57	74.77	77.10 / 77.47, 76.71
	✔		✗	72.68	68.11	70.32 / 71.85, 68.78	79.66	74.59	77.04 / 76.80, 77.27
		✔	✗	72.84	67.69	70.17 / 69.42, 70.91	79.01	74.02	76.43 / 77.43, 75.43
✔	✔		✗	73.38	68.81	71.02 / 71.86, 70.15	80.66	75.52	78.01 / 77.91, 78.09
✔		✔	✗	73.71	68.14	70.82 / 70.34, 71.27	80.90	76.20	78.48 / 78.35, 78.60
	✔	✔	✗	74.17	69.12	71.56 / 70.89, 72.21	81.13	76.20	78.59 / 78.42, 78.73
✔	✔	✔	✗	74.63	69.21	71.82 / 71.68, 71.94	**81.74**	76.29	78.92 / 78.67, 79.16
			✔	71.80	72.34	72.07 / 72.39, 71.73	77.38	79.19	78.27 / 76.64, 79.89
✔			✔	72.41	73.35	72.88 / 71.99, 73.77	79.04	79.87	79.45 / 79.09, 79.81
	✔		✔	73.85	73.74	73.80 / 72.64, 74.96	79.40	79.50	79.45 / 78.66, 80.24
		✔	✔	72.69	74.27	73.47 / 72.19, 74.75	79.05	79.72	79.39 / 78.41, 80.36
✔	✔		✔	74.43	73.73	74.08 / 73.19, 74.96	79.67	**80.65**	80.16 / 79.16, 81.14
✔		✔	✔	74.31	74.33	74.32 / 74.45, 74.19	79.97	79.56	79.76 / 79.06, 80.47
	✔	✔	✔	75.15	74.12	74.63 / 74.69, 74.57	79.41	79.98	79.70 / 79.49, 79.90
✔	✔	✔	✔	**75.35**	**74.67**	**75.01 / 74.91, 75.10**	81.06	80.53	**80.79 / 79.94, 81.64**

Table 2: The proposed MIMO that employs (a) multi-input Language Models, POS tags, and Concept-Attribute-Phrase sequences, (b) multi-output tag sequences, (c) BERT-based encoder performs the best on tuple extraction.

The ovals in each confusion matrix present the most significant type of error. Of a small set of actual subjects, the model predicted them as objects, and vise versa, though the fact/condition role and concept role were correctly predicted.

The dashed circles show the second frequent type of error. Of the actual "I-f2p" tokens, the model predicted that 7 were "B-f2p"; for the actual "I-c2p", it predicted that 6 were "B-c2p". Basically, it was because of missing the beginning word of the relational phrases. Of the actual "B-f3a" tokens, the model predicted 6 were "I-f2p". Future work will aim at improving the prediction of the boundaries of long relational phrases.

4.2.4 Efficiency

All the experiments were conducted on 16 Graphics Cards (GeForce GTX 1080 Ti), where one individual model only used 1 GPU. Each model was trained for 1,000 epochs. For the BiLSTM-LSTMd MIMOs, the pre-training took 2.4 hours and the re-training (TCT layer) took 0.4 hour. For the BERT-LSTMd MIMOs of the best performance, the pre-training took 3.5 hours and the re-training took 0.9 hour. It took 5.7 hours to extract fact and condition tuples from 141 million sentences in the MEDLINE text data. It is comparable with existing approaches in terms of scalability.

4.3 Results on BioNLP2013

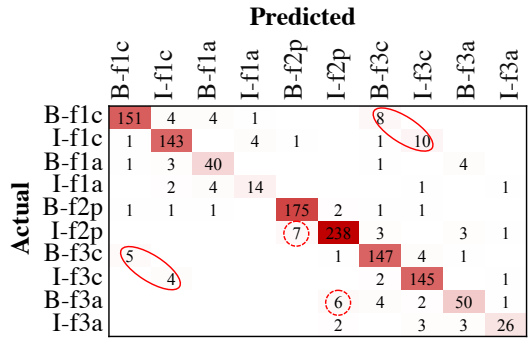
As shown in Table 3, the BERT-LSTMd MIMO model achieves an F1 score of 0.790 on tuple extraction from BioNLP2013. Note that the model was trained on BioCFE that has no overlapping sentence with BioNLP2013. This score is comparable with the testing F1 score on the BioCFE (0.808), which demonstrates the effectiveness and reliability of the proposed model.

Our model improves the F1 score relatively by 4.2% over the best baseline BERT-LSTMd. The improvement on recall is more substantial: It improves recall relatively by 5.8%. It was because of the design of the multi-output module: the TCT layer generates multiple tag sequences based on the relation names predicted by the RNT layer. A token in a statement sentence may have different roles in different tuples of the same type (fact or condition). For example, given the following statement sentence:

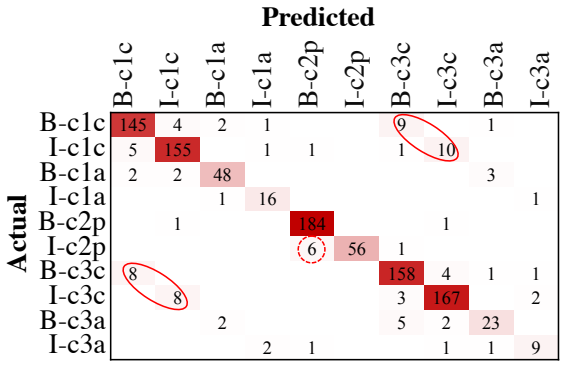
"*Immunohistochemical staining of the tumors demonstrated a decreased number of blood vessels in the treatment group versus the controls.*"

The proposed model is able to find one fact tuple and two condition tuples precisely:

-*Fact 1:* ({tumors:immunohistochemical_staining}, demonstrated, {blood_vessels:decreased_number})



(a) Fact tags



(b) Condition tags

Figure 3: Confusion matrices on predicting fact tags (Top) and condition tags (Bottom) in BioCFE data.

-*Condition 1:* (blood_vessels,in,**treatment_group**)
-*Condition 2:* (**treatment_group**,versus,controls)

Note that the concept "treatment_group" acts as the *object* of Condition Tuple 1 (having tags "B/I-c3c") and the *subject* of Condition Tuple 2 (having tags "B/I-c1c"). The multi-output design tackled this issue while other models could not.

Compared with BioCFE: On BioCFE, the F1 score on condition tuple extraction is a bit higher than that on fact tuple extraction (81.64 vs 79.94). On BioNLP2013, we have the opposite observation (78.58 vs 79.42). They are still comparable but if we look at the error cases, we find that most of the false predictions of condition tuple come from long sentences (having more than 30 words). And 35% of the sentences in BioNLP are long sentences, while only 5% in Bio CFE are long. Long dependency modeling is always challenging for IE, especially condition extraction. We will study it in the future work.

4.4 A Visualized Case Study

Scientific knowledge graph enables effective search and exploration. It is certainly important to represent the conditions of the corresponding fact being valid in the graph. As we have applied our model to the large MEDLINE dataset, Figure 4 visualizes the fact and condition tuples extracted from four statement sentences about "cell proliferation". On the left side, we find (1) "VPA treatment" and the "incubation" of "HDLs" increased cell proliferation, while (2)"Chlorin e6-PDT" and the "inhibition" of "MiR-199a-5p" decreased cell proliferation. On the right, we are aware of the conditions of the factual claims. They describe the methodology of the observation (e.g., "using", "in combination with") or the context (e.g., "in" a specific disease or "from" specific animals). In some other cases, we find the temperature and pH values are detected as the conditions of observations.

Methods	*Prec. (%)*	*Rec. (%)*	*F1 (%)*
Allennlp OpenIE	41.84	37.87	39.75
Stanford OpenIE	45.04	42.99	43.99
Structured SVM	49.69	47.54	48.59
CRF	59.55	56.29	57.88
BiLSTM-LSTMd	65.33	63.09	64.19
BERT-LSTMd	76.99	74.74	75.85
MIMO (BERT based)	**78.93**	**79.07**	**79.00**

Table 3: The BERT-LSTMd MIMO model performs the best on tuple extraction in BioNLP2013.

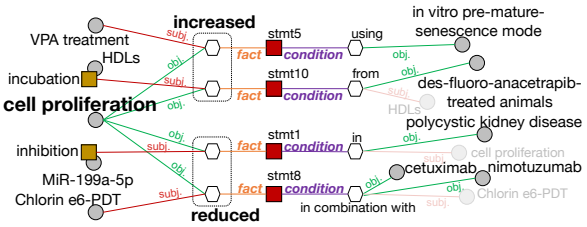


Figure 4: Structuring tuples detected from four statement sentences that mention "cell proliferation" into a snapshot of scientific knowledge graph with fact tuples on the left and condition tuples on the right.

5 Related Work

5.1 Scientific Information Extraction

Information extraction in scientific literature, e.g., computer science, biology and chemistry, has been receiving much attention in recent years. ScienceIE in computer science focus on concept recognition and factual relation extraction (Luan et al., 2017; Gábor et al., 2018; Luan et al., 2018). ScienceIE in biological literature aims at identifying the relationships between biological concepts

(i.e., proteins, diseases, drugs and genes) (Kang et al., 2012; Liu et al., 2018; Xu et al., 2018). Rule-based approaches were used in early studies (Rindflesch and Fiszman, 2003; Kang et al., 2012). Recently, a wide line of neural network models have been proposed and outperformed traditional methods (Wang et al., 2018b; Liu et al., 2018; Xu et al., 2018; Jiang et al., 2019). Wang *et al.* (2018b) investigated different kinds of word embeddings on different NLP tasks in the biological domain. Liu *et al.* (2018) employed attention-based neural networks to extract chemical-protein relations. Xu *et al.* (2018) used the BiLSTM model to recognize the drug interaction. In our work, we extract biological relational facts as well as their conditions. The condition tuples are essential to interpreting the factual claims.

5.2 Open-Domain IE

Open IE refers to the extraction of (subject, relation, object)-triples from plain text (Angeli et al., 2015; Stanovsky et al., 2018; Saha et al., 2018; Wang et al., 2018a). The schema for the relations does not need to be specified in advance. Distant supervision has been widely used because the size of the benchmark data is often limited (Banko et al., 2007; Wu and Weld, 2010). Stanovsky *et al.* (2018) proposed supervised neural methods for OpenIE. The idea was to transform annotated tuples into tags and learn via sequence tagging. We create a new tag schema and propose a novel sequence labeling framework.

5.3 Sequence Labeling for IE

Statistical models have been studied for long, including Hidden Markov Models (HMM), Support Vector Machine (SVM), and Conditional Random Fields (CRF) (Lafferty et al., 2001; Tsochantaridis et al., 2005; Passos et al., 2014; Luo et al., 2015; Li et al., 2018). However, these methods rely heavily on hand-crafted features. Then neural network models become popular and obtain more promising performance than traditional statistical methods (Yang and Mitchell, 2017; Zheng et al., 2017; Wang et al., 2019; Yu et al., 2019). So, we use them as strong baselines.

6 Conclusions

We present a new problem to find conditional information in scientific statements. We created a new tag schema for jointly extracting condition and fact tuples from scientific text. We proposed a multi-input multi-output sequence labeling model to utilize results from well-established related tasks and extract an uncertain number of fact(s)/condition(s). Our model yields improvement over all the baselines on a newly annotated dataset BioCFE and a public dataset BioNLP2013. We argue that structured representations of knowledge, such as fact/condition tuple, for scientific statements will enable more intelligent downstream applications. In the future work, we will explore the use of the structured tuples to bridge the gap between text content and knowledge-based applications, such as knowledge-based scientific literature search.

Acknowledgments

he authors would like to thank the anonymous reviewers for their valuable comments and insightful suggestions. This work was supported in part by National Key Research and Development Program of China No. 2018YFB1005103, National Natural Science Foundation of China (NSFC) No. 61632011, and NSF Grant CCF-1901059.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. 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)*, volume 1, pages 344–354.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haifa Zargayouna, and Thierry Charnois. 2018. Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *SemEval*, pages 679–688.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*, pages 770–778.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 328–339.

Meng Jiang, Jingbo Shang, Taylor Cassidy, Xiang Ren, Lance M Kaplan, Timothy P Hanratty, and Jiawei Han. 2017. Metapad: Meta pattern discovery from massive text corpora. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 877–886. ACM.

Tianwen Jiang, Tong Zhao, Bing Qin, Ting Liu, Nitesh V Chawla, and Meng Jiang. 2019. 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*, pages 1634–1642. ACM.

Ning Kang, Bharat Singh, Zubair Afzal, Erik M van Mulligen, and Jan A Kors. 2012. Using rule-based natural language processing to improve disease normalization in biomedical text. *Journal of the American Medical Informatics Association*, 20(5):876–881.

Matthieu Labeau, Kevin Löser, and Alexandre Allauzen. 2015. Non-lexical neural architecture for fine-grained pos tagging. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 232–237.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Qi Li, Meng Jiang, Xikun Zhang, Meng Qu, Timothy P Hanratty, Jing Gao, and Jiawei Han. 2018. Truepie: Discovering reliable patterns in pattern-based information extraction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1675–1684. ACM.

Sijia Liu, Feichen Shen, Ravikumar Komandur Elayavilli, Yanshan Wang, Majid Rastegar-Mojarad, Vipin Chaudhary, and Hongfang Liu. 2018. Extracting chemical–protein relations using attention-based neural networks. *Database*, 2018:bay102.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreferencefor scientific knowledge graph construction. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*.

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2641–2651, Copenhagen, Denmark. Association for Computational Linguistics.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888.

David L Miller. 1947. The nature of scientific statements. *Philosophy of Science*, 14(3):219–223.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7. Association for Computational Linguistics.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Thomas C Rindflesch and Marcelo Fiszman. 2003. The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *Journal of biomedical informatics*, 36(6):462–477.

Swarnadeep Saha et al. 2018. Open information extraction from conjunctive sentences. In *COLING*, pages 2288–2299.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. 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)*, volume 1, pages 885–895.

Victor N Tomilin, Alena L Cherezova, Yuri A Negulyaev, and Svetlana B Semenova. 2016. Trpv5/v6 channels mediate ca2+ influx in jurkat t cells under the control of extracellular ph. *Journal of cellular biochemistry*, 117(1):197–206.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(Sep):1453–1484.

Xuan Wang, Yu Zhang, Qi Li, Yinyin Chen, and Jiawei Han. 2018a. 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*, pages 291–300. ACM.

Xueying Wang, Haiqiao Zhang, Qi Li, Yiyu Shi, and Meng Jiang. 2019. A novel unsupervised approach for precise temporal slot filling from incomplete and noisy temporal contexts. In *The World Wide Web Conference*, pages 3328–3334. ACM.

Yanshan Wang, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul Kingsbury, and Hongfang Liu. 2018b. A comparison of word embeddings for the biomedical natural language processing. *Journal of biomedical informatics*, 87:12–20.

Fei Wu and Daniel S Weld. 2010. Open information extraction using wikipedia. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 118–127. Association for Computational Linguistics.

Bo Xu, Xiufeng Shi, Zhehuan Zhao, and Wei Zheng. 2018. Leveraging biomedical resources in bi-lstm for drug-drug interaction extraction. *IEEE Access*, 6:33432–33439.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1436–1446.

Wenhao Yu, Zongze Li, Qingkai Zeng, and Meng Jiang. 2019. Tablepedia: Automating pdf table reading in an experimental evidence exploration and analytic system. In *The World Wide Web Conference*, pages 3615–3619. ACM.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1227–1236.