Learning Structures of Biological Processes with Joint Inference

Pei-Chun Chen, Hao Zhang

I. Introduction

Biological Processes involves a series of events and entities that are related to one another. Current state-of-the-art question answering systems can handle bio questions about atomic facts (such as In the fungi life cycle, what cells form a diploid zygote?), but are unable to answer more complex "how" and "why" questions that involve deep understanding of the process structure. This project seeks to address this problem by having the computer read and construct structures of biological processes from text. Specifically, the structure includes event-event relation and event-entity association, which can be thought of as graph with edges. The edge between two events describes temporal, causal or coreferent relation. The edge between an event and an entity exists if the entity is associated with (an argument of) the event. Thus, the event-event relation prediction is a multi-class classification while the event-entity relation is binary.

In this project, we aim to predict the events, entities associated with events, and event-event relations at the same time through joint inference in the hope of lowering cascading errors introduced from the pipeline framework [5], i.e. predict entities and inter-event relations based on the event prediction. The joint inference problem is formulated using integer linear programming (ILP) and solved by Gurobi [1]. The insights from extensive error analyses are incorporated to improve the system. In the end, the joint inference system achieves better performance (F1 score) than the pipeline system, which suggests further exploration of join inference in our future work.

II. Previous Work

Our project is an extension of the paper Learning Biological Processes with Global Constraints [5]. The work in [5] focuses on event-event relation prediction and has a strong assumption that gold event triggers are already known. Thus, it learns only from the gold triggers in the training set and predicts only on the gold triggers in the test set. This largely simplifies the original problem, i.e. predict all the events, entities and event-event relations from plain text. In this project, we address the original problem and predict the whole structure using joint inference.

One thing to note is that [5]. also uses joint inference and formulates global constraints (connectivity, chain structure and relation triads) on event-event relations with ILP. However, the joint inference in this project is very different. We continue using some global constraints (connectivity and part of the relation triads, specifically SAME contradiction and PREV contradiction) from [5] but add many new constraints connecting the event prediction, entity prediction and event-event relation prediction, which make predicting the whole structure in one shot possible. The details are provided in section IV.A.

III. Dataset

We continue using the same dataset as [5] does. A brief description is provided here for the completeness of the report. For more details, please refer to section II and section IV of [5].

148 process descriptions were extracted by going through chapters from the textbook Biology by Neil A. Campbell and Jane B. Reece. The definition of a process is a contiguous sequence of sentences that describes a process, i.e., a series of events that lead towards some objective. Each process description was then annotated by biologists after presented with annotation guidelines. Process descriptions were parsed with Stanford constituency and dependency parsers (Klein and Manning, 2003 [4]; de Marneffe et al., 2006 [3]) and 35 process descriptions were set aside as a test set (number of

The authors are with the Department of Electronic Engineering at Stanford University, CA, 94305, United States. Email: peichun2@stanford.edu, hzhang22@stanford.edu.

training set trigger pairs: 1932, number of test set trigger pairs: 906). All the result numbers shown in this project are averaged over 10-fold cross validation on the training set.

There are 11 possible event-event relations, R = {PREV, NEXT, SUPER, SUB, CAUSES, CAUSED, ENABLES, ENABLED, COTEMP, SAME, NONE} [5]. Among them, PREV-NEXT, SUPER-SUB, CAUSES-CAUSED, ENABLES-ENABIED are pairs of relations with reverse directed relation of each other. More specifically, the classifier is a function that maps a pair of events to a relation, ex.$f(t_i, t_j) = PREV \iff f(t_j, t_i) = NEXT$, where t_i is the i-th trigger in its description. Since f(t_i, t_j) completely determines f(t_j, t_i), we only consider pairs with $i < j$.

Entities were labeled with their semantic roles with respect to the events, ex. AGENT, ORIGIN, DESTINATION, etc. However, in this project, we only consider if an entity is an argument of an event and ignores the semantic role labeling (SRL) from the annotation.

IV. Experiments and Results

A. Joint Inference

In this section, we describe how we incorporate constrains into our model to generate coherent global process structures. Let $T_{i,e}$ be the score for an event trigger i to be labeled as e ($e \in \{T, F\}$, i.e. trigger i is an event or not) and $t_{i,e}$ be the corresponding indicator variable. Let $A_{i,j,e}$ be the score for a relation e between an event trigger i and an candidate argument j, where $e \in \{T, F\}$ (entity j is an argument of event i or not) and $a_{i,j,e}$ be the corresponding indicator variable. Let $Y_{i,j,r}$ be the score for a relation r (the possible relations are described in section III) between the trigger pair (t_i, t_j) and $y_{i,.j,r}$ be the corresponding indicator variable. The scores $T_{i,e}, A_{i,j,e}$ and $Y_{i,j,r}$ are from MaxEnt based local classifiers trained on the annotated samples using lexical, dependency tree based and parse tree based features. Feature details are discussed in section III of [5].

Our goal is to find an assignment for the indicators.

$$t = \{t_{i,e} | 1 \leq i \leq n, e \in E\} \quad (1)$$

$$a = \{a_{i,j,e} | 1 \leq i \leq n, 1 \leq j \leq m_i, e \in E\} \quad (2)$$

$$y = \{y_{i,j,r} | 1 \leq i \leq j \leq n, r \in R\} \quad (3)$$

where n is the number of possible event triggers, m_i is the number of possible arguments of trigger i, $E = \{T, F\}$ and R is defined in section III. With no global constraints, this can be formulates as the following ILP:

$$\begin{aligned} \underset{t,a,y}{\operatorname{argmax}} \sum_{i,e} T_{i,e} t_{i,e} &+ \sum_{i,j,e} A_{i,j,e} a_{i,j,e} \quad (4) \\ &+ \sum_{i,j,r} Y_{i,j,r} y_{i,j,r} \\ s.t.\ \forall_i \sum_e t_{i,e} &= 1 \quad (5) \\ \forall_{i,j} \sum_e a_{i,j,e} &= 1 \quad (6) \\ \forall_{i,j} \sum_r y_{i,j,r} &= 1 \quad (7) \end{aligned}$$

The constraint here ensures that each candidate event trigger is classified as either an event or not, each candidate argument for an event trigger is classified as either an argument or not, and there is exactly one relation between each event pair. In the rest of this section, we describe constraints that result in a coherent global process structure.

1) Arguments for events only: If a candidate event trigger is classified as non-event, then it should not have any arguments. In our definition, only events have associated arguments. Thus,

$$\forall_{i,j} t_{i,F} \rightarrow a_{i,j,F} \iff t_{i,F} \leq a_{i,j,F} \quad (8)$$

2) Arguments for an event cannot overlap: Two arguments related to the same event cannot overlap a constraint that has been used in the past in SRL (Toutanova et al., [6]). More specifically, the sub-tree of a tree node already marked as an entity for an event t should not be tagged as an entity for t as well. Thus, for each parent entity node, we identify its children (sub-trees). If the parent is classified as an argument for an event t, then any of the children should not be an argument.

$$\begin{aligned} \forall_{i,j} a_{i,j,T} \rightarrow a_{i,\text{children}(j),F} \iff \quad (9) \\ a_{i,j,T} \leq a_{i,\text{children}(j),F} \end{aligned}$$

3) Relations between two events only: An event-event relation, by its name, should only exist between two candidate event triggers that are both classified as events. Thus, if there is a not-NONE

relation between t_i and t_j, i.e. $f(t_i, t_j) = r\ R$ and r is not NONE, then both t_i and t_j should be events.

$$\forall_{i,j,r,r\neq\text{NONE}}\ y_{i,j,r} \rightarrow t_{i,T} \wedge t_{j,T} \iff \\ y_{i,j,r} \leq t_{i,T} \text{ and } y_{i,j,r} \leq t_{j,T} \quad (10)$$

TABLE I

	P	R	F1
Event	0.708	0.608	0.651
Entity	0.728	0.132	0.223
E-E Realtion	0.602	0.031	0.08

10-fold cross validation results on the training set using joint inference

The results of joint inference are shown in Table I. We can see that except for the event trigger identification, the performance, especially for the event-event relation, is very bad. We carefully examined each prediction and found the event-event relation local classifier predicted almost all the event pairs to have a relation NONE. The reason is that we are now considering all the possible event-event pairs so the relation labels are hugely skewed, i.e. most of them are NONE. The relation distribution in Table II makes everything clear. Since the number of NONE relations dominates, the local classifier learns to predict NONE most of the time, which causes the recall to be extremely low and hence a low F1.

TABLE II

Training Set		Test Set	
Relation	Count	Relation	Count
Next	9	Next	2
Enabled	1	Enabled	1
Enables	15	Enables	0
Enabled	1	Enabled	1
Cotemporal	66	Cotemporal	11
Caused	18	Caused	0
Causes	117	Causes	26
Super	34	Super	2
Sub	9	Sub	0
Same	74	Same	5
Previous	187	Previous	21
NONE	114716	Super	23776

Event-event relation distribution in the training set and the test set

Thus, for the joint inference to work, we have to first make the data less skewed so that the local classifier predicts better. Two approaches, down sampling and event filtering, are discussed in the following 2 sections, respectively.

B. Joint Inference and Down Sampling

TABLE III

	P	R	F1
Event	0.384	0.832	0.522
Entity	0.726	0.135	0.227
E-E Realtion	0.050	0.311	0.085

10-fold cross validation results on the training set using joint inference with down sampling on NONE event-event relation

The prediction performance after down sampling the NONE-relation pairs (specifically, we randomly choose x NONE-relation pairs where x is the number of not-NONE relation pairs) during training is shown in Table III. The F1 scores for entity identification and event-event relation classification only increase a bit, and the score for event identification is largely reduced from 0.651 to 0.522. The possible reason for this bad performance is data distribution mismatch. Specifically, down sampling is only performed on the training data and we predict on all possible candidates. Thus, the data distribution is very different, resulting in high false positives (lowering precision from 0.602 to 0.05) for event-event relation prediction. We thus try another approach to combat the skewed data in IV.C.

C. Joint Inference and Filtering

Instead of enumerating all possible event-event and event-entity pairs, we set a threshold θ to filter events. Specifically, for a possible event trigger t, if $f(t, T) \geq \theta$, i.e. the score of classifying t as an event is bigger than θ, then we consider possible entities and relations with other events for it. If $f(t, T) < \theta$, we totally ignore t. Note that the original joint inference in IV.A is equal to setting θ as 0, i.e. considering all possibilities. The pipeline framework equals to setting θ as 0.5, i.e. only if an event trigger t is identified as an event by the local classifier do we consider its entities and relations with other identified events. Thus, the idea is to explore θ between 0 and 0.5 in the hope of finding a sweet spot where we do not filter out too many

true events but are able to hugely reduce the NONE-relation pairs and make the classifier works.

One main difference between section IV.B and section IV.C is that in section IV.B, we only down sample event-event pairs with NONE relation, which does not help the entity prediction performance at all. In this section, by filtering out events, we remove less possible event-entity pairs and event-event pairs at the same time. The other main difference is that event filtering is performed both during training and testing, so the data distribution is similar, i.e. the training data that the classifier learns from and the testing data that the classifier predicts on have similar distribution.

TABLE IV

θ	Structure	P	R	F1
0.05	Event	0.707	0.614	0.654
	Entity	0.704	0.159	0.258
	E-E Realtion	0.567	0.062	0.109
0.10	Event	0.699	0.628	0.659
	Entity	0.654	0.201	0.306
	E-E Realtion	0.514	0.097	0.160
0.15	Event	0.678	0.659	0.666
	Entity	0.631	0.233	0.339
	E-E Realtion	0.455	0.137	0.206
0.20	Event	0.659	0.702	0.677
	Entity	0.523	0.361	0.426
	E-E Realtion	0.348	0.192	0.243
0.25	Event	0.650	0.722	**0.681**
	Entity	0.500	0.406	0.446
	E-E Realtion	0.335	0.232	**0.271**
0.30	Event	0.651	0.720	0.680
	Entity	0.502	0.416	**0.453**
	E-E Realtion	0.294	0.244	0.263
0.35	Event	0.661	0.691	0.673
	Entity	0.504	0.401	0.445
	E-E Realtion	0.300	0.242	0.266
0.40	Event	0.680	0.680	0.678
	Entity	0.520	0.402	0.451
	E-E Realtion	0.307	0.240	0.267
0.45	Event	0.690	0.646	0.665
	Entity	0.521	0.394	0.448
	E-E Realtion	0.315	0.231	0.265
0.50	Event	0.716	0.602	0.651
	Entity	0.540	0.373	0.439
	E-E Realtion	0.333	0.215	0.259

10-fold cross validation results on the training set using joint inference with filtering controlled by θ

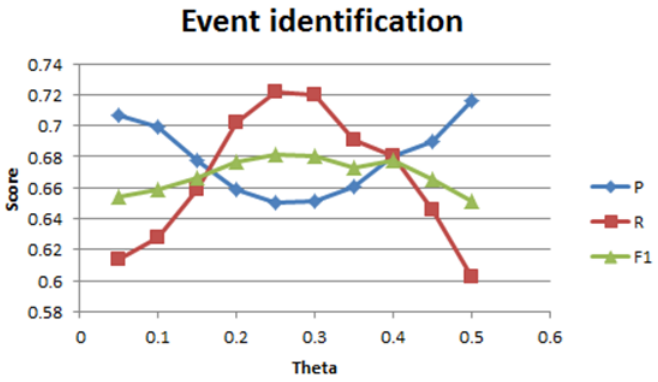

Fig. 1: Event Identification

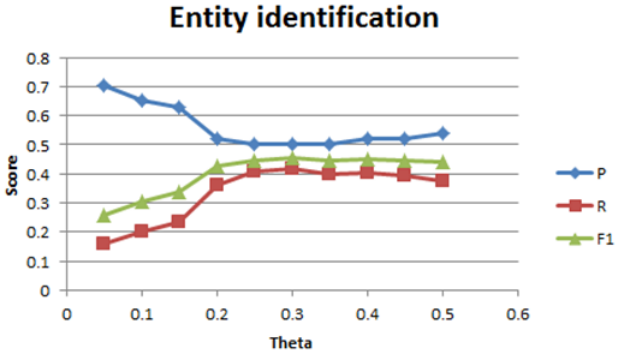

Fig. 2: Entity Identification

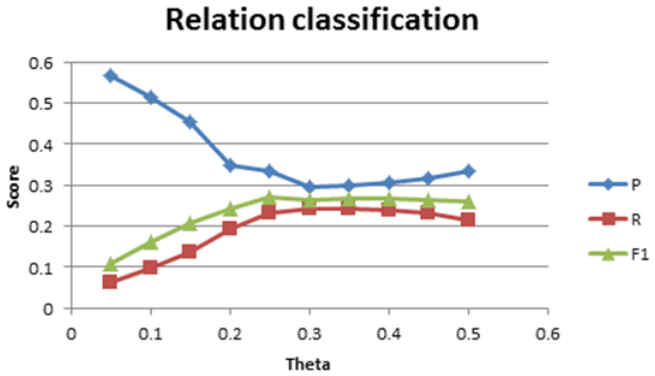

Fig. 3: Relation Classification

The results with different values of θ are shown in Table IV. From the table, we can see that $\theta = 0.25$ yields the best F1 for event trigger and event-event relation prediction, 0.681 and 0.271, respectively. $\theta = 0.3$ yields the best F1 for entity prediction, 0.453. In addition, as θ increases from 0.05 to approximately 0.25, the precision of event, entity and relation prediction all decreases, while the recall increases. As θ further increases from 0.25 to 0.5, both trends go the opposite directions, i.e. precision

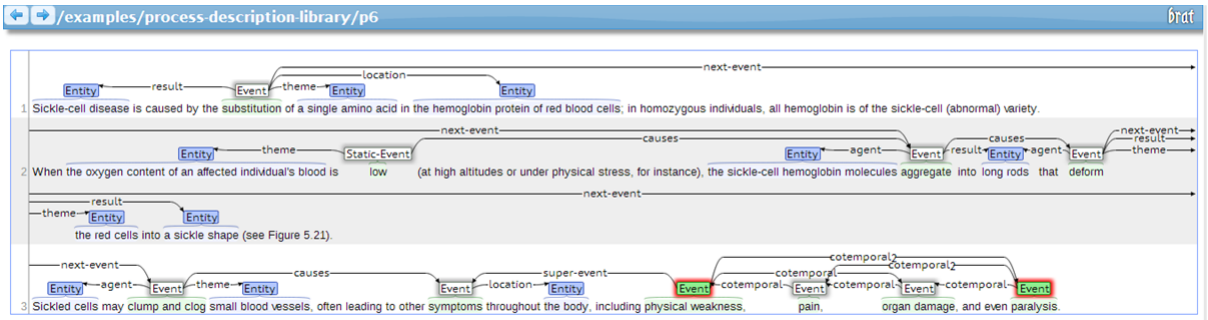

Fig. 4: Example1

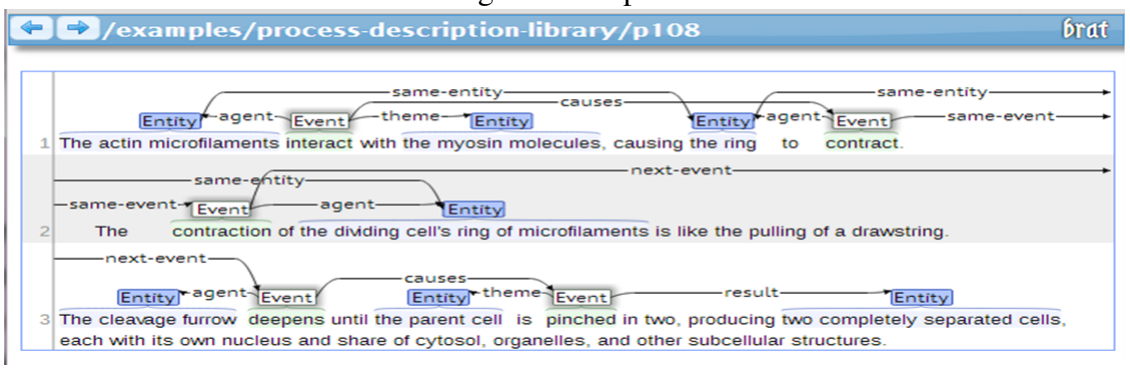

Fig. 5: Example2

increases while recall decreases. The recall value is generally smaller than the precision value and dominates the F1 trend. The trends of precision, recall and F1 vs θ. are depicted in Figure 1, Figure2 and Figure3.

V. Analysis

A. Error Analysis

In addition to observing the resulting precision, recall and F1, we look at the bio processes and compare the annotation with our prediction in order to gain insight on what mistakes we frequently make and come up with ideas to fix them. We discuss about two examples below.

1) Example 1: For this process, our system correctly predicts 'deform' and 'aggregate' as events. However, it falsely predicts 'long rods that deform the red cells into a sickle shape', which overlaps event 'deform', as an entity for event 'aggregate'.

2) Example 2: For this process, our system accurately predicts all the events. However, for event 'contract' (which has an entity 'the ring'), no argument for it is predicted at all.

B. Improvement from Error Analysis Insights

From our observations, we collect a few ideas for potential improvement. The examples shown in Error Analysis part lead to two additional constraints implemented using ILP:

1) Argument and event cannot overlap: Although not always true in any dataset, in this specific dataset, none of the arguments and the events overlaps with each other. Thus, we build a map storing <key, value> pairs as <event trigger t, all the candidate arguments overlapping t >. If t is classified as an event, then all the candidate arguments overlapping it should not be classified as arguments for any event.

$$\forall_{i,j,k} t_{i,T} \rightarrow a_{k,j,F} \Longleftrightarrow t_{i,T} \leq a_{k,j,F} \tag{11}$$

where $a_{k,j}$ is a candidate argument overlapping t_i

2) Event has at least one entity: In this dataset, we notice almost all the events have at least one entity. Thus, we make it a hard constraint.

$$\forall_{i,j} t_{i,T} \rightarrow \sum_{i,j} a_{i,j,T} \geq 1 \Longleftrightarrow \tag{12}$$

$$-t_{i,T} + \sum_{i,j} a_{i,j,T} \geq 0$$

After incorporating the additional two constraints into the joint inference, the overall result can be seen in Table V. To maintain the brevity of the report, we only show the results using the optimal θ (0.25) from section IV.C. Comparing with the results using

TABLE V

	P	R	F1
Event	0.678	0.686	0.679
Entity	0.489	0.430	0.456
E-E Realtion	0.360	0.221	0.272

10-fold cross validation results on the training set using joint inference with additional constraints learnt from error analysis

$\theta = 0.25$ in Table IV, the precision of event and event-event relation goes up from 0.65 to 0.678 and 0.335 to 0.36, respectively, while the recall goes down from 0.722 to 0.686 and 0.232 to 0.221. In contrast, the precision of entity goes down from 0.5 to 0.489 while the recall goes up from 0.406 to 0.430. Comparing the overall F1 score, the event F1 decreases a bit from 0.681 to 0.679, while the entity F1 increases from 0.446 to 0.456 and the relation F1 from 0.271 to 0.272.

The score change validates learning from error, but also shows the complexity of structure prediction using joint inference. A small change can increase the prediction performance in one place (ex. entity) but harm the other (ex. event). Thus, it requires much trial and error to improve the system.

VI. Conclusion

In this project, we perform trigger identification, entity identification and event-event relation extraction using joint inference. The best F1 scores achieved for event, entity and event-event relation on the training set (averaged over 10-fold cross validation) are 0.681, 0.456 and 0.272, respectively. The prediction performance for event-event relation is lower than the others since it is a multi-class classification problem, which is much harder than the identification (binary) problem.

In order to compare with the pipeline framework, we train on the whole training set using $\theta = 0.25$ and predict on the test set. We limit our focus on event-event relation prediction while comparing. Using joint inference, we achieve (precision, recall, F1) of (0.3497, 0.25, 0.2916), which is better than the pipeline approach result: (P, R, F1) = (0.3514, 0.2281, 0.2766). However, the improvement is not as much as expected. We will try to exploit the power of joint inference better by examining deeper into the structure of bio-processes and come up with more constraints. Other possible future directions are discussed in section VII.

VII. Future Work

Joint inference gives better performance than the original pipeline approach. However, the improvement is limited. Thus, we decide to incorporate joint inference into the context of joint learning, which is already half implemented. Instead of using joint inference only on the test set with the weights learnt from the training set, we use joint inference during training as well and a structure perceptron to update weights according to the prediction from joint inference. Joint learning has been proved useful in predicting structures [2], and we hope this approach can improve the performance effectively.

The ultimate goal for this project is a question answering (QA) system that can handle biological questions. Given the annotated bio processes, we still need questions with answers from these processes in order to build the QA system. Currently, two bio-major Stanford undergraduate students are generating questions for us. Once our structure prediction system achieves an acceptable performance, we will start building the QA system.

VIII. Acknowledgement

We would like to thank Jonathan Berant and Vivek Srikumar for advice and fruitful discussion. In addition, we would like to thank our external collaborators including Vulcan employees who helped annotate the bioprocess data.

References

[1] Gurobi website www.gurobi.com.

[2] Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics, 2002.

[3] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[4] Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

[5] Aju Thalappillil Scaria, Jonathan Berant, Mengqiu Wang, Christopher D Manning, Justin Lewis, Brittany Harding, and Peter Clark. Learning biological processes with global constraints.

[6] Kristina Toutanova, Aria Haghighi, and Christopher D Manning. A global joint model for semantic role labeling. *Computational Linguistics*, 34(2):161–191, 2008.