Learning Biological Processes with Global Constraints

Aju Thalappillil Scaria; Jonathan Berant; Mengqiu Wang and Christopher D. Manning
Stanford University, Stanford

Justin Lewis and Brittany Harding
University of Washington, Seattle

Peter Clark
Allen Institute for Artificial Intelligence, Seattle

Abstract

Biological processes are complex phenomena involving a series of events that are related to one another through various relationships. Systems that can understand and reason over biological processes would dramatically improve the performance of semantic applications involving inference such as question answering (QA) -- specifically “How?” and “Why?” questions. In this paper, we present the task of process extraction, in which events within a process and the relations between the events are automatically extracted from text. We represent processes by graphs whose edges describe a set of temporal, causal and co-reference event-event relations, and characterize the structural properties of these graphs (e.g., the graphs are connected). Then, we present a method for extracting relations between the events, which exploits these structural properties by performing joint inference over the set of extracted relations. On a novel dataset containing 148 descriptions of biological processes (released with this paper), we show significant improvement comparing to baselines that disregard process structure.

1 Introduction

A process is defined as a series of inter-related events that involve multiple entities and lead to an end result. Product manufacturing, economical developments, and various phenomena in life and social sciences can all be viewed as types of processes. Processes are complicated objects; consider for example the biological process of ATP synthesis described in Figure 1. This process involves 12 entities and 8 events. Additionally, it describes relations between events and entities, and the relationship between events (e.g., the second occurrence of the event ‘enter’, causes the event ‘changing’).

∗ Both authors equally contributed to the paper
we characterize a set of global properties of process structure that can be utilized during process extraction. For example, all events in a process are somehow connected to one another. Also, processes usually exhibit a “chain-like” structure reflecting process progression over time. We show that incorporating such global properties into our model and performing joint inference over the extracted relations significantly improves the quality of process structures predicted. We conduct experiments on a novel dataset of process descriptions from the textbook “Biology” (Campbell and Reece, 2005) that were annotated by trained biologists. Our method does not require any domain-specific knowledge and can be easily adapted to non-biology domains.

The main contributions of this paper are:

1. We define process extraction and characterize processes’ structural properties.
2. We model global structural properties in processes and demonstrate significant improvement in extraction accuracy.
3. We publicly release a novel data set of 148 fully annotated biological process descriptions along with the source code for our system. The dataset and code can be downloaded from http://nlp.stanford.edu/software/bioprocess/.

2 Process Definition and Dataset

We define a process description as a paragraph or sequence of tokens $x = \{x_1, \ldots, x_{|x|}\}$ that describes a series of events related by temporal and/or causal relations. For example, in ATP synthesis (Figure 1), the event of rotor spinning causes the event where an internal rod spins.

We model the events within a process and their relations by a directed graph $P = (V, E)$, where the nodes $V = \{1, \ldots, |V|\}$ represent event mentions and labeled edges $E$ correspond to event-event relations. An event mention $v \in V$ is defined by a trigger $t_v$, which is a span of words $x_i, x_{i+1}, \ldots, x_j$; and by a set of argument mentions $A_v$, where each argument mention $a_v \in A_v$ is also a span of words labeled by a semantic role $l$ taken from a set $\mathcal{L}$. For example, in the last event mention of ATP synthesis, $t_v = \text{produce}$, and one of the argument mentions is $a_v = (\text{ATP}, \text{RESULT})$. A labeled edge $(u, v, r)$ in the graph describes a relation $r \in R$ between the event mentions $u$ and $v$. The task of process extraction is to extract the graph $P$ from the text $x$.

A natural way to break down process extraction into sub-parts is to first perform semantic role labeling (SRL), that is, identify triggers and predict argument mentions with their semantic role, and then extract event-event relations between pairs of event mentions. In this paper, we focus on the second step, where given a set of event triggers $T$, we find all event-event relations, where a trigger represents the entire event. For completeness, we now describe the semantic roles $\mathcal{L}$ used in our dataset, and then

---

1 Argument mentions are also related by coreference relations, but we neglect that since it is not central in this paper.
present the set of event-event relations $R$.

The set $L$ contains standard semantic roles such as
AGENT, THEME, ORIGIN, DESTINATION and LOCATION. Two additional semantic roles were employed that are relevant for biological text: RESULT corresponds to an entity that is the result of an event, and RAW-MATERIAL describes an entity that is used or consumed during an event. For example, the last event ‘produce’ in Figure 1, has ‘ATP’ as the RESULT, and ‘ADP’ as the RAW-MATERIAL.

The event-event relation set $R$ contains the following (assuming a labeled edge $(u, v, r)$):

1. PREV denotes that $u$ is an event immediately before $v$. Thus, the edges $(u, v, \text{PREV})$ and $(v, w, \text{PREV})$, preclude the edge $(u, w, \text{PREV})$. For example, in “When a photon strikes ... energy is passed ... until it reaches ...”, there is no edge (strikes, reaches, PREV) due to the intervening event ‘passed’.

2. COTEMP denotes that events $u$ and $v$ overlap in time (e.g., the first two event mentions flowing and enter in Figure 1).

3. SUPER denotes that event $u$ includes event $v$. For instance, in “During DNA replication, DNA polymerases proofread each nucleotide...” there is an edge (DNA replication, proofread, SUPER).

4. CAUSES denotes that event $u$ causes event $v$ (e.g., the relation between changing and spins in sentence 2 of Figure 1).

5. ENABLES denotes that event $u$ creates preconditions that allow event $v$ to take place. For example, the description “…cause cancer cells to lose attachments to neighboring cells... allowing them to spread into nearby tissues” has the edge (lose, spread, ENABLES). An intuitive way to think about the difference between Causes and Enables is the following: if $u$ causes $v$ this means that if $u$ happens, then $v$ happens. If $u$ enables $v$, then if $u$ does not happen, then $v$ does not happen.

6. SAME denotes that $u$ and $v$ both refer to the same event (spins and Spinning in Figure 1).

Early work on temporal logic (Allen, 1983) contained more temporal relations than are used in our relation set $R$. We chose a relation set $R$ that captures the essential aspects of temporal relations between events in a process, while keeping the annotation as simple as possible. For instance, we include the SUPER relation that appears in temporal annotations such as the Timebank corpus (Pustejovsky et al., 2003) and Allen’s work, but in practice was not considered by many temporal ordering systems (Chambers and Jurafsky, 2008; Yoshikawa et al., 2009; Do et al., 2012). Importantly, our relation set also includes the relations CAUSES and ENABLES, which are fundamental to modeling processes and go beyond simple temporal ordering.

We also added event coreference (SAME) to $R$. Do et al. (2012) used event coreference information in a temporal ordering task to modify probabilities provided by pairwise classifiers prior to joint inference. In this paper, we simply treat SAME as another event-event relation, which allows us to easily perform joint inference and employ structural constraints that combine both coreference and temporal relations simultaneously. For example, if $u$ and $v$ are the same event, then there can exist no $w$, such that $u$ is before $w$, but $v$ is after $w$ (see Section 3.3).

We annotated 148 process descriptions based on the aforementioned definitions. Further details on annotation and data set statistics are provided in Section 4 and Table 1.

### Structural properties of processes
Coherent processes exhibit many structural properties. For example, two argument mentions related to the same event cannot overlap – a constraint that has been used in the past in SRL (Toutanova et al., 2008). In this paper we focus on three main structural properties of the graph $P$. First, in a coherent process, all events mentioned are related to one another, and hence the graph $P$ must be connected. Second, processes tend to have a “chain-like” structure where one event follows another, and thus we expect

|               | Avg | Min | Max |
|---------------|-----|-----|-----|
| # of sentences| 3.80| 1   | 15  |
| # of tokens   | 89.98| 19  | 319 |
| # of events   | 6.20| 2   | 15  |
| # of non-NONE relations | 5.64| 1   | 24  |

Table 1: Process statistics over 148 process descriptions. NONE is used to indicate no relation.
| Deg. | Gold | Local | Global |
|------|------|-------|--------|
| 0    | 0    | 29    | 0      |
| 1    | 219  | 274   | 224    |
| 2    | 369  | 337   | 408    |
| 3    | 46   | 14    | 17     |
| ≥4   | 22   | 2     | 7      |

Table 2: Node degree distribution for event mentions on the training set. Predictions for the Local and Global models were obtained using 10-fold cross validation.

nodes’ degree to generally be \( \leq 2 \). Indeed, 90% of event mentions have degree \( \leq 2 \), as demonstrated by the Gold column of Table 2. Last, if we consider relations between all possible triples of events in a process, clearly some configurations are impossible, while others are common (illustrated in Figure 2). In Section 3.3, we show that modeling these properties using a joint inference framework improves the quality of process extraction significantly.

### 3 Joint Model for Process Extraction

Given a paragraph \( x \) and a trigger set \( T \), we wish to extract all event-event relations \( E \). Similar to Do et al. (2012), our model consists of a local pairwise classifier and global constraints. We first introduce a classifier that is based on features from previous work. Next, we describe novel features specific for process extraction. Last, we incorporate global constraints into our model using an ILP formulation.

#### 3.1 Local pairwise classifier

The local pairwise classifier predicts relations between all event mention pairs. In order to model the direction of relations, we expand the set \( \mathcal{R} \) to include the reverse of four directed relations: PREV-NEXT, SUPER-SUB, CAUSES-CAUSED, ENABLES-ENABLED. After adding NONE to indicate no relation, and including the undirected relations COTEMP and SAME, \( \mathcal{R} \) contains 11 relations. The classifier is hence a function \( f : T \times T \to \mathcal{R} \). As an example, \( f(t_i, t_j) = \text{PREV} \) if \( f(t_j, t_i) = \text{NEXT} \). Let \( n \) be the number of triggers in a process, and \( t_i \) be the \( i \)-th trigger in its description. Since \( f(t_i, t_j) \) completely determines \( f(t_j, t_i) \), it suffices to consider only pairs with \( i < j \). Note that the process graph \( \mathcal{P} \) is undirected under the new definition of \( \mathcal{R} \).

Table 3 describes features from previous work (Chambers and Jurafsky, 2008; Do et al., 2012) extracted for a trigger pair \((t_i, t_j)\). Asterisks (*) indicate features that are duplicated, once for each trigger.

#### 3.2 Classifier extensions

A central source of information to extract event-event relations from text are connectives such as after, during, etc. However, there is variability in the occurrence of these connectives as demonstrated by the following two sentences (connectives in boldface, triggers in italics):

1. **Because** alleles are exchanged during gene flow, genetic differences are **reduced**.

2. **During** gene flow, alleles are exchanged, and genetic differences are **hence reduced**.

Even though both sentences express the same relation (exchanged, reduced, CAUSES), the connectives used and their linear position with respect to the triggers are different. Also, in sentence 1, gene flow intervenes between exchanged and reduced. Since our dataset is small, we wish to identify the triggers related to each connective, and share features between such sentences. We do this using the syntactic structure and by clustering the connectives.
Sentence 1 presents a typical case where by walking up the dependency tree from the marker because, we can find the triggers related by this marker: because mark exchanged advect reduced. Whenever a trigger is the head of an adverbial clause and marked by a mark dependency label, we walk on the dependency tree and look for a trigger in the main clause that is closest to the root (or the root itself in this example). By utilizing the syntactic structure, we can correctly spot that the trigger gene flow is not related to the trigger exchanged through the connective because, even though they are linearly closer. In order to reduce sparseness of connectives, we created a hand-made clustering of 30 connectives that maps words into clusters\(^2\) (e.g., because, since and hence to a “causality” cluster). After locating the relevant pair of triggers, we use these clusters to fire the same feature for connectives belonging to the same cluster. We perform a similar procedure whenever a trigger is part of a prepositional phrase (imagine sentence 1 starting with “due to allele exchange during gene flow . . . ”) by walking up the constituency tree, but details are omitted for brevity. In sentence 2, the connective hence is an adverbial modifier of the trigger reduced. We look up the cluster for the connective hence and fire the same feature for the adjacent triggers exchanged and reduced.

We further extend our features to handle the rich relation set necessary for process extraction. The first event of a process is often expressed as a nominalization and includes subsequent events (SUPER relation), e.g., “The Calvin cycle begins by incorporating . . . “. To capture this, we add a feature that fires when the first event of the process description is a noun. We also add two features targeted at the

\(^2\)The full set of connectives and their clustering are provided as part of our publicly released package.

SAME relation: one indicating if the lemmas of \(t_i\) and \(t_j\) are the same, and another specifying the determiner of \(t_j\), if it exists. Certain determiners indicate that an event trigger has already been mentioned, e.g., the determiner this hints a SAME relation in “The next steps decompose citrate back to oxaloacetate. This regeneration makes . . . “. Last, we add as a feature the dependency path between \(t_i\) and \(t_j\), if it exists, e.g., in “meiosis produces cells that divide . . . “, the feature \(\frac{dobj}{rcmod}\) is fired for the trigger pair produces and divide. In Section 4.1 we empirically show that our extensions to the local classifier substantially improve performance.

For our pairwise classifier, we train a maximum entropy classifier that computes a probability \(p_{ijr}\) for every trigger pair \((t_i, t_j)\) and relation \(r\). Hence, 
\[
f(t_i, t_j) = \arg\max_r p_{ijr}.
\]

### 3.3 Global Constraints

Naturally, pairwise classifiers are local models that can violate global properties in the process structure. Figure 3 (left) presents an example for predictions made by the pairwise classifier, which result in two triggers (deleted and duplicated) that are isolated from the rest of the triggers. In this section, we discuss how we incorporate constraints into our model to generate coherent global process structures.

Let \(\theta_{ijr}\) be the score for a relation \(r\) between the trigger pair \((t_i, t_j)\) (e.g., \(\theta_{ijr} = \log p_{ijr}\)), and \(y_{ijr}\) be the corresponding indicator variable. Our goal is to find an assignment for the indicators \(y = \{y_{ijr} \mid 1 \leq i < j \leq n, r \in \mathcal{R} \}\). With no global constraints this can be formulated as the following ILP:

\[
\begin{align*}
\text{max} & \quad \sum_{i < j} \sum_{r \in \mathcal{R}} \theta_{ijr} y_{ijr} \\
\text{subject to} & \quad \sum_{r \in \mathcal{R}} y_{ijr} \leq 1, \quad \forall i < j,
\end{align*}
\]

Figure 2: Relation triangles (a)-(c) are common in the gold standard while (d)-(e) are impossible.
\[
\arg\max_y \sum_{ijr} \theta_{ijr} y_{ijr} \\
\text{s.t.} \forall i,j \sum_r y_{ijr} = 1
\]

where the constraint ensures exactly one relation between each event pair. We now describe constraints that result in a coherent global process structure.

**Connectivity** Our ILP formulation for enforcing connectivity is a minor variation of the one suggested by Martins et al. (2009) for dependency parsing. In our setup, we want \( \mathcal{P} \) to be a connected undirected graph, and not a directed tree. However, an undirected graph \( \mathcal{P} \) is connected iff there exists a directed tree that is a subgraph of \( \mathcal{P} \) when edge directions are ignored. Thus the resulting formulation is almost identical and is based on flow constraints which ensure that there is a path from a designated root in the graph to all other nodes.

Let \( \mathcal{R} \) be the set \( \mathcal{R} \setminus \text{NONE} \). An edge \((t_i, t_j)\) is in \( E \) iff there is some non-NONE relation between \( t_i \) and \( t_j \), i.e. iff \( y_{ij} := \sum_{r \in \mathcal{R}} y_{ijr} \) is equal to 1. For each variable \( y_{ij} \) we define two auxiliary binary variables \( z_{ij} \) and \( z_{ji} \) that correspond to edges of the directed tree that is a subgraph of \( \mathcal{P} \) when edge directions are ignored. Thus the resulting formulation is almost identical and is based on flow constraints which ensure that there is a path from a designated root in the graph to all other nodes.

Next, we add constraints that ensure that the graph structure induced by the auxiliary variables is a tree rooted in an arbitrary node 1 (The choice of root does not affect connectivity). We add for every \( i \neq j \) a flow variable \( \phi_{ij} \) which specifies the amount of flow on the directed edge \( z_{ij} \).

\[
\sum_i z_{i1} = 0, \forall_{j \neq 1} \sum_i z_{ij} = 1 \tag{3}
\]
\[
\sum_i \phi_{1i} = n - 1 \tag{4}
\]
\[
\forall_{j \neq 1} \sum_i \phi_{ij} - \sum_k \phi_{jk} = 1 \tag{5}
\]
\[
\forall_{i \neq j} \phi_{ij} \leq n \cdot z_{ij} \tag{6}
\]

Equation 3 says that all nodes in the graph have exactly one parent, except for the root that has no parents. Equation 4 ensures that the outgoing flow from the root is \( n - 1 \), and Equation 5 states that each of the other \( n - 1 \) nodes consume exactly one unit of flow. Last, Equation 6 ties the auxiliary variables to the flow variables, making sure that flow occurs only on edges. The combination of these constraints guarantees that the graph induced by the variables \( z_{ij} \) is a directed tree and consequently the graph induced by the objective variables \( y \) is connected.

**Chain structure** A chain is a connected graph where the degree of all nodes is \( \leq 2 \). Table 2 presents nodes’ degree and demonstrates that indeed process graphs are close to being chains. The following constraint bounds nodes’ degree by 2:

\[
\forall_j (\sum_{i<j} y_{ij} + \sum_{j<k} y_{jk} \leq 2) \tag{7}
\]

Since graph structures are not always chains, we add this as a soft constraint, that is, we penalize the objective for each node with degree \( > 2 \). The chain structure is one of the several soft constraints we enforce. Thus, our modified objective function is \( \sum_{ijr} \theta_{ijr} y_{ijr} + \sum_{k \in \mathcal{K}} \alpha_k C_k \), where \( \mathcal{K} \) is the set of soft constraints, \( \alpha_k \) is the penalty (or reward for desirable structures), and \( C_k \) indicates whether a constraint is violated (or satisfied). Note that under this formulation our model is simply a constrained conditional model (wei Chang et al., 2012). The parameters \( \alpha_k \) are tuned on a development set (see Section 4).

**Relation triads** A relation triad (or a relation triangle) for any three triggers \( t_i, t_j \) and \( t_k \) in a process is a 3-tuple of relations \((f(t_i, t_j), f(t_j, t_k), f(t_i, t_k))\). Clearly, some triads are impossible while others are quite common. To find triads that could improve process extraction, the frequency of all possible triads in both the training set and the output of the pairwise classifier were found, and we focused on those for which the classifier and the gold standard disagree. We are interested in triads that never occur in training data but are predicted by the classifier, and vice versa. Figure 2 illustrates some of the triads found and Equa-
tions 8-12 provide the corresponding ILP formulations. Equations 8-10 were formulated as soft constraints (expanding the set $K$) and were incorporated by defining a reward $\alpha_k$ for each triad type. On the other hand, Equations 11-12 were formulated as hard constraints to prevent certain structures.

1. **SAME** transitivity (Figure 2a, Eqn. 8): Coreference transitivity has been used in past work (Finkel and Manning, 2008) and we incorporate it by a constraint that encourages triads that respect transitivity.

2. **CAUSE-COTEMP** (Figure 2b, Eqn. 9): If $t_i$ causes both $t_j$ and $t_k$, then often $t_j$ and $t_k$ are co-temporal. E.g., in “genetic drift has led to a loss of genetic variation and an increase in the frequency of . . .”, a single event causes two subsequent events that occur simultaneously.

3. **COTEMP** transitivity (Figure 2c, Eqn. 10): If $t_i$ is co-temporal with $t_j$ and $t_j$ is co-temporal with $t_k$, then usually $t_i$ and $t_k$ are either co-temporal or denote the same event.

4. **SAME** contradiction (Figure 2d, Eqn. 11): If $t_i$ is the same event as $t_k$, then their temporal ordering with respect to a third trigger $t_j$ may result in a contradiction, e.g., if $t_j$ is after $t_i$, but before $t_k$. We define 5 temporal categories that generate $\binom{5}{2}$ possible contradictions, but for brevity present just one representative hard constraint. This constraint depends on prediction of temporal and co-reference relations jointly.

5. **PREV** contradiction (Figure 2e, Eqn. 12): As mentioned (Section 3.3), if $t_i$ is immediately before $t_j$, and $t_j$ is immediately before $t_k$, then $t_i$ cannot be immediately before $t_k$.

\[ y_{ij} \text{SAME} + y_{jk} \text{SAME} + y_{ik} \text{SAME} \geq 3 \quad (8) \]
\[ y_{ij} \text{CAUSE} + y_{ik} \text{CAUSE} + y_{jk} \text{COTEMP} \geq 3 \quad (9) \]
\[ y_{ij} \text{COTEMP} + y_{jk} \text{COTEMP} + y_{ik} \text{COTEMP} + y_{ik} \text{SAME} \geq 3 \quad (10) \]
\[ y_{ij} \text{PREV} + y_{jk} \text{PREV} + y_{ik} \text{SAME} \leq 2 \quad (11) \]
\[ y_{ij} \text{PREV} + y_{jk} \text{PREV} - y_{ik} \text{NONE} \leq 1 \quad (12) \]

\[ y_{ij} \text{SAME} + y_{jk} \text{SAME} + y_{ik} \text{SAME} \geq 3 \quad (8) \]
\[ y_{ij} \text{CAUSE} + y_{ik} \text{CAUSE} + y_{jk} \text{COTEMP} \geq 3 \quad (9) \]
\[ y_{ij} \text{COTEMP} + y_{jk} \text{COTEMP} + y_{ik} \text{COTEMP} + y_{ik} \text{SAME} \geq 3 \quad (10) \]
\[ y_{ij} \text{PREV} + y_{jk} \text{PREV} + y_{ik} \text{SAME} \leq 2 \quad (11) \]
\[ y_{ij} \text{PREV} + y_{jk} \text{PREV} - y_{ik} \text{NONE} \leq 1 \quad (12) \]

We experimented with a reward for certain triads or a penalty for others and empirically found that using rewards results in better performance on the development set.

We used the Gurobi optimization package\(^4\) to find an exact solution for our ILP, which contains $O(n^2|R|)$ variables and $O(n^3)$ constraints. We also developed an equivalent formulation amenable to dual decomposition (Sontag et al., 2011), which is a faster approximation method. But practically, solving the ILP exactly with Gurobi was quite fast (average/median time per process: 0.294 sec/0.152 sec on a standard laptop).

4 Experimental Evaluation

We extracted 148 process descriptions by going through chapters from the textbook “Biology” and marking any contiguous sequence of sentences that describes a process, i.e., a series of events that lead towards some objective. Then, each process description was annotated by a biologist. The annotator was first presented with annotation guidelines and annotated 20 descriptions. The annotations were then discussed with the authors, after which all process descriptions were annotated. After training a second biologist, we measured inter-annotator agreement $\kappa = 0.69$, on 30 random process descriptions.

Process descriptions were parsed with Stanford constituency and dependency parsers (Klein and Manning, 2003; de Marneffe et al., 2006), and 35 process descriptions were set aside as a test set (number of training set trigger pairs: 1932, number of test set trigger pairs: 906). We performed 10-fold cross validation over the training set for feature selection and tuning of constraint parameters. For each constraint type (connectivity, chain-structure, and five triad constraints) we introduced a parameter and tuned the seven parameters by coordinate-wise ascent, where for hard constraints a binary parameter controls whether the constraint is used, and for soft constraints we attempted 10 different reward/penalty values. For our global model we defined $\theta_{ijr} = \log p_{ijr}$, where $p_{ijr}$ is the probability at edge $(t_i, t_j)$ for label $r$, given by the pairwise classifier.

We test the following systems: (a) **All-PREV**: Since the most common process structure was chain-like, we simply predict PREV for every two adjacent triggers in text. (b) **LocalBase**: A pairwise classifier with features from previous work (Section 3.1) (c) **Local**: We experimented with a reward for certain triads or a penalty for others and empirically found that using rewards results in better performance on the development set.

\(^4\)www.gurobi.com
Table 4: Test set results on all experiments. Best number in each column is bolded. † and ‡ denote statistical significance ($p < 0.01$) against Localbase and Local baselines, respectively.

|                | Temporal P | Temporal R | Temporal F₁ | Full P | Full R | Full F₁ |
|----------------|------------|------------|-------------|--------|--------|---------|
| All-Prev       | 58.4       | 54.8       | 56.6        | 34.1   | 32.0   | 33.0    |
| Localbase      | 61.5       | 51.8       | 56.2        | 52.1   | 43.9   | 47.6    |
| Local          | 63.2       | 55.7†      | 59.2        | 54.7   | 48.3†  | 51.3    |
| Chain          | 64.5†‡     | 60.5†      | 62.4†       | 56.1   | 52.6†  | 54.3†   |
| Global         | 63.9       | 61.4†      | 62.6†       | 56.2   | 54.0†  | 55.0†   |

A pairwise classifier with all features (Section 3.2) (d) Chain: For every two adjacent triggers, choose the non-NONE relation with highest probability according to Local. This baseline heuristically combines our structural assumptions with the pairwise classifier. We deterministically choose a connected chain structure, and then use the classifier to label the edges. (e) Global: Our full model that uses ILP inference.

To evaluate system performance we compare the set of predictions on all trigger pairs to the gold standard annotations and compute micro-averaged precision, recall and F₁. We perform two types of evaluations: (a) Full: evaluation on our full set of 11 relations (b) Temporal: Evaluation on temporal relations only, by collapsing PREV, CAUSES, and ENABLES to a single category and similarly for NEXT, CAUSED, and ENABLED (inter-annotator agreement $\kappa = 0.75$). We computed statistical significance of our results with the paired bootstrap resampling method of 2000 iterations (Efron and Tibshirani, 1993), where the units resampled are trigger-trigger-relation triples.

### 4.1 Results

Table 4 presents performance of all systems. We see that using global constraints improves performance almost invariably on all measures in both full and temporal evaluations. Particularly, in the full evaluation Global improves recall by 12% and overall F₁ improves significantly by 3.7 points against Local ($p < 0.01$). Recall improvement suggests that modeling connectivity allowed Global to add correct relations in cases where some events were not connected to one another.

The Local classifier substantially outperforms Localbase. This indicates that our novel features (Section 3.2) are important for discriminating between process relations. Specifically, in the full evaluation Local improves precision more than in the temporal evaluation, suggesting that designing syntactic and semantic features for connectives is useful for distinguishing PREV, CAUSES, and ENABLES when the amount of training data is small.

The Chain baseline performs only slightly worse than our global model. This demonstrates the strong tendency of processes to proceed linearly from one event to the other, which is a known property of discourse structure (Schegloff and Sacks, 1973). However, since the structure is deterministically fixed, Chain is highly inflexible and does not allow any extensions or incorporation of other structural constraints or domain knowledge. Thus, it can be used as a simple and efficient approximation but is not a good candidate for a real system. Further support for the linear nature of process structure is provided by the All-Prev baseline, which performs poorly in the full evaluation, but in temporal evaluation works reasonably well.

Table 2 presents the degree distribution of Local and Global on the development set comparing to the gold standard. The degree distribution of Global is more similar to the gold standard than Local. In particular, the connectivity constraint ensures that there are no isolated nodes and shifts mass from nodes with degree 0 and 1 to nodes with degree 2.

Table 5 presents the order in which constraints were introduced into the global model using coordinate ascent on the development set. Connectivity is the first constraint to be introduced, and improves performance considerably. The chain constraint, on the other hand, is included third and the improvement in F₁ score is relatively smaller. This can be explained by the distribution of degrees in Table 2 which shows that the predictions of Local does not have many nodes with degree $> 2$. As for triad constraints, we see that four constraints are important and are included in the model, but one is discarded.

Last, we examined the results of Global when macro-averaging over processes, i.e., assigning each process the same weight by computing recall, precision and F₁ for each process and averaging those scores. We found that results are quite similar (with a slight improvement): in the full evalua-
Table 5: Order by which constraint parameters were set using coordinate ascent on the development set. For each parameter, the value chosen and $F_1$ score after including the constraint are provided. Negative values correspond to penalties, positive values to rewards, and a value of $\infty$ indicates a hard constraint.

| Order | Parameter name          | Value (α) | $F_1$ score |
|-------|-------------------------|-----------|-------------|
| 1     | Connectivity constraint | $\infty$  | 51.2        |
| 2     | SAME transitivity       | 0.5       | 52.9        |
| 3     | Chain constraint        | -0.5      | 53.3        |
| 4     | CAUSE-COTEMP            | 1.0       | 53.7        |
| 5     | PREV contradiction      | $\infty$  | 53.8        |
| 6     | SAME contradiction      | $\infty$  | 53.9        |

The global model depends on the predictions of the local classifier, and so enforcing global constraints does not guarantee improvement in performance. For instance, if \textit{Local} produces a graph that is disconnected (e.g., \textit{deleted} in Figure 3, left), then \textit{Global} will add an edge. However, the label of the edge is determined by scores computed based on the local classifier, and if this prediction is wrong, we will now be penalized for both the false negative of the correct class (just as before), and also for the false positive of the predicted class. Despite that we see that \textit{Global} improves overall performance by 3.7 $F_1$ points on the test set.

5 Related Work

A related line of work is biomedical event extraction in recent BioNLP shared tasks (Kim et al., 2009; Kim et al., 2011). Earlier work employed a pipeline architecture where first events are found, and then their arguments are identified (Miwa et al., 2010; Björne et al., 2011). Subsequent methods predicted events and arguments jointly using Markov logic (Poon and Vanderwende, 2010) and dependency parsing algorithms (McClosky et al., 2011). Riedel and McCallum (2011) further improved performance by capturing correlations between events and enforcing consistency across arguments.

Temporal event-event relations have been extensively studied (Chambers and Jurafsky, 2008; Yoshikawa et al., 2009; Denis and Muller, 2011; Do et al., 2012; McClosky and Manning, 2012; D’Souza and Ng, 2013), and we leverage such techniques in our work (Section 3.1). However, we extend beyond temporal relations alone, and strongly rely on dependencies between process events. Chambers and Jurafsky (2011) learned event templates (or frames), where events that are related to one another and their semantic roles are extracted. Recently, Cheung et al. (2013) proposed an unsupervised generative model for inducing such templates. A major difference in our work is that we do not learn typical event relations from a large and redundant corpus, but are given a paragraph and have a “one-shot” chance to extract the process structure.

We showed in this paper that global structural properties lead to significant improvements in extraction accuracy, and ILP is an effective framework
for modeling global constraints. Similar observations and techniques have been proposed in other information extraction tasks. Reichart and Barzilay (2012) tied information from multiple sequence models that describe the same event by using global higher-order potentials. Berant et al. (2011) proposed a global inference algorithm to identify entailment relations. There is an abundance of examples of enforcing global constraints in other NLP tasks, such as in coreference resolution (Finkel and Manning, 2008), parsing (Rush et al., 2012) and named entity recognition (Wang et al., 2013).

6 Conclusion

Developing systems that understand process descriptions is an important step towards building applications that require deeper reasoning, such as biological process models from text, intelligent tutoring systems, and non-factoid QA systems. In this paper we have presented the task of process extraction, and developed methods for extracting relations between process events. Processes contain events that are tightly coupled through strong dependencies. We have shown that exploiting these structural dependencies and performing joint inference over all event mentions can significantly improve accuracy over several baselines. We have also released a new dataset containing 148 fully annotated descriptions of biological processes. Though the models we built were trained on biological processes, they do not encode domain specific information, and hence should be extensible to other domains.

In this paper we assumed that event triggers are given as input. In future work, we want to perform trigger identification jointly with extraction of event-event relations. As explained in Section 4.2, the performance of our system is confined by the performance of the local classifier, which is trained on relatively small amounts of data. Since data annotation is expensive, it is important to improve the local classifier without increasing the annotation burden. For example, one can use unsupervised methods that learn narrative chains (Chambers and Jurafsky, 2011) to provide some prior on the typical order of events. Alternatively, we can search on the web for redundant descriptions of the same process and use this redundancy to improve classification. Last, we would like to integrate our method into QA systems and allow non-factoid questions that require deeper reasoning to be answered by matching the questions against the learned process structures.

Acknowledgments

The authors would like to thank Roi Reichart for fruitful discussion and the anonymous reviewers for their constructive feedback. This work was partially funded by Vulc...
Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2011. Extracting contextualized complex biological events with rich graph-based feature sets. *Computational Intelligence*, 27(4):541–557.

Neil Campbell and Jane Reece. 2005. *Biology*. Benjamin Cummings.

Nathanael Chambers and Daniel Jurafsky. 2008. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of EMNLP*.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *ACL*, pages 976–986.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Proceedings of NAACL-HLT*.

Marie-Catherine de Marneffe, Bill MacCartney, and Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2010. Joint in-
terpretation of event and temporal expressions in text. In *New Directions in Question Answering*.

Roi Reichart and Regina Barzilay. 2012. Multi-event extraction guided by global constraints. In *Proceedings of HLT-NAACL*.

Sebastian Riedel and Andrew McCallum. 2011. Fast and robust joint models for biomedical event extraction. In *Proceedings of EMNLP*.

Alexanderson M. Rush, Roi Reichert, Michael Collins, and Amir Globerson. 2012. Improved parsing and POS tagging using inter-sentence consistency constraints. In *Proceedings of EMNLP*.

Emanuel A Schegloff and Harvey Sacks. 1973. Opening up closings. *Semiotica*, 8(4):289–327.

David McClosky, Mihai Surdeanu, and Christopher D. Manning. 2011. Event extraction as dependency parsing. In *Proceedings of ACL*, pages 1626–1635.

Makoto Miwa, Rune Sætre, Jin-Dong Kim, and Jun’ichi Tsujii. 2010. Event extraction with complex event classification using rich features. *J. Bioinformatics and Computational Biology*, 8(1).

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *Proceedings of HLT-NAACL*.

James Pustejovsky, José M. Castañó, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 2003. TimeML: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering*.

Jenny Rose Finkel and Christopher D. Manning. 2008. A global joint model for semantic role labeling. *Computational Linguistics*, 34(2):161–191.

Mengjia Wang, Wanxiang Che, and Christopher D. Manning. 2013. Effective bilingual constraints for semi-supervised learning of named entity recognizers. In *Proceedings of AAAI*.

David McClosky, Mihai Surdeanu, and Christopher D. Manning. 2011. Event extraction as dependency parsing. In *Proceedings of ACL*, pages 1626–1635.

Makoto Miwa, Rune Sætre, Jin-Dong Kim, and Jun’ichi Tsujii. 2010. Event extraction with complex event classification using rich features. *J. Bioinformatics and Computational Biology*, 8(1).

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *Proceedings of HLT-NAACL*.

James Pustejovsky, José M. Castañó, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 2003. TimeML: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering*.

Roi Reichart and Regina Barzilay. 2012. Multi-event extraction guided by global constraints. In *Proceedings of HLT-NAACL*.

Sebastian Riedel and Andrew McCallum. 2011. Fast and robust joint models for biomedical event extraction. In *Proceedings of EMNLP*.

Alexander M. Rush, Roi Reichert, Michael Collins, and Amir Globerson. 2012. Improved parsing and POS tagging using inter-sentence consistency constraints. In *Proceedings of EMNLP*.

Emanuel A Schegloff and Harvey Sacks. 1973. Opening up closings. *Semiotica*, 8(4):289–327.

David McClosky, Mihai Surdeanu, and Christopher D. Manning. 2011. Event extraction as dependency parsing. In *Proceedings of ACL*, pages 1626–1635.

Makoto Miwa, Rune Sætre, Jin-Dong Kim, and Jun’ichi Tsujii. 2010. Event extraction with complex event classification using rich features. *J. Bioinformatics and Computational Biology*, 8(1).

Hoifung Poon and Lucy Vanderwende. 2010. Joint inference for knowledge extraction from biomedical literature. In *Proceedings of HLT-NAACL*.

James Pustejovsky, José M. Castañó, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 2003. TimeML: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering*.

Roi Reichart and Regina Barzilay. 2012. Multi-event extraction guided by global constraints. In *Proceedings of HLT-NAACL*.

Sebastian Riedel and Andrew McCallum. 2011. Fast and robust joint models for biomedical event extraction. In *Proceedings of EMNLP*.

Alexander M. Rush, Roi Reichert, Michael Collins, and Amir Globerson. 2012. Improved parsing and POS tagging using inter-sentence consistency constraints. In *Proceedings of EMNLP*.

Emanuel A Schegloff and Harvey Sacks. 1973. Opening up closings. *Semiotica*, 8(4):289–327.

David McClosky, Mihai Surdeanu, and Christopher D. Manning. 2011. Event extraction as dependency parsing. In *Proceedings of ACL*, pages 1626–1635.