Learning Biological Processes with Global Constraints

Abstract

Biological processes are complex phenomena involving a series of events that are related to one another through multiple dependencies. Systems that can understand and reason over text describing biological processes could dramatically improve the performance of semantic applications 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 large 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'*).

Automatically extracting the structure of processes from text is crucial for applications that require reasoning, such as non-factoid QA. For instance, answering a question on ATP synthesis, such as *"How do H+ ions contribute to the production of ATP?"* requires a structure that links *H+ ions* (Figure 1, sentence 1) to *ATP* (Figure 1, sentence 4) through a sequence of intermediate events. Such *"How?"* questions are common on FAQ websites (Surdeanu et al., 2011), which further supports the importance of process extraction.

Process extraction is related to two recent lines of work in Information Extraction – event extraction and timeline construction. Traditional event extraction focuses on identifying a closed set of events within a single sentence. For example, the BioNLP 2009 and 2011 shared tasks (Kim et al., 2009; Kim et al., 2011) consider nine events types related to proteins. In practice, events are currently almost always extracted from a single sentence. Process extraction, on the other hand, is centered around discovering *relations* between events that span *multiple* sentences. The set of possible event types in process extraction is also much larger.

Timeline construction involves identifying temporal relations between events (Do et al., 2012; McClosky and Manning, 2012; D'Souza and Ng, 2013), and is thus related to process extraction as both focus on event-event relations spanning multiple sentences. However, events in processes are tightly coupled in ways that go beyond simple temporal ordering, and these dependencies are central for the process extraction task. Hence, capturing process structure requires modeling a larger set of relations that includes temporal, causal and co-reference relations.

In this paper, we formally define the task of process extraction and present automatic extraction methods. Our approach handles an open set of event types and works over multiple sentences, extracting a rich set of event-event relations. Furthermore, we characterize a set of global properties of process structure that can be utilized during process extrac-

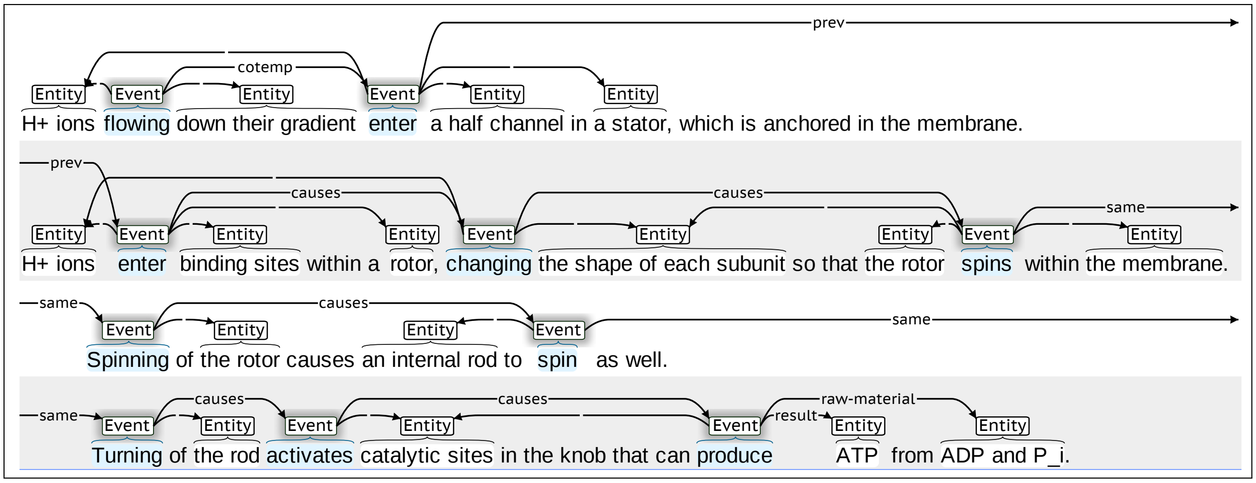

prev
cotemp
Entity
Event
Entity
Event
Entity
Entity
H+ ions flowing down their gradient enter a half channel in a stator, which is anchored in the membrane.
prev
causes
causes
same
Entity
Event
Entity
Entity
Event
Entity
Entity
Event
Entity
H+ ions enter binding sites within a rotor, changing the shape of each subunit so that the rotor spins within the membrane.
same
causes
same
Event
Entity
Entity
Event
Spinning of the rotor causes an internal rod to spin as well.
same
causes
causes
raw-material
result
Event
Entity
Event
Entity
Event
Entity
Entity
Turning of the rod activates catalytic sites in the knob that can produce ATP from ADP and P_i.

Figure 1: Partial annotation of the ATP synthesis process. Most of the semantic roles have been removed for simplicity.

tion. For example, all events in a process are somehow connected to one another. Also, processes usually exhibit a "chain-like" structure reflecting process progresses 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 significantly improve extraction accuracy.
3. We publicly release a novel data set of 148 fully annotated biological process descriptions.

2 Process Definition and Dataset

We define a process description as a paragraph or sequence of tokens $\mathbf{x} = \{x_1, \dots x_{|\mathbf{x}|}\}$ 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 $\mathcal{P} = (V, E)$, where the nodes $V = \{1, \dots, |V|\}$ 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 $x_i, x_{i+1}, \dots, x_j$; 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, $t_v =$ *produce*, and one of the argument mentions is $a_v =$ *(ATP,* RESULT*)*. A labeled edge (u, v, r) in the graph describes a relation $r \in \mathcal{R}$ between the event mentions u and v. The task of process extraction is to extract the graph $\mathcal{P}$ from the text $\mathbf{x}$.[1]

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 $\mathcal{T}$, 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 present the set of event-event relations $\mathcal{R}$.

The set $\mathcal{L}$ 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,

[1] Argument mentions are also related by coreference relations, but we neglect that since it is not central in this paper.

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 relation set $\mathcal{R}$ 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, PREV) and (v, w, PREV), preclude the edge (u, w, 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).

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

Our relation set contains the relations CAUSES and ENABLES, which are important for modeling processes and go beyond just temporal ordering. The SUPER relation appears in temporal annotations such as the Timebank corpus (Pustejovsky et al., 2003) and in work on temporal logic (Allen, 1983), but in practice it is not considered by many temporal ordering systems (Chambers and Jurafsky, 2008; Yoshikawa et al., 2009; Do et al., 2012).

We also added event coreference (SAME) to $\mathcal{R}$. 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)

	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

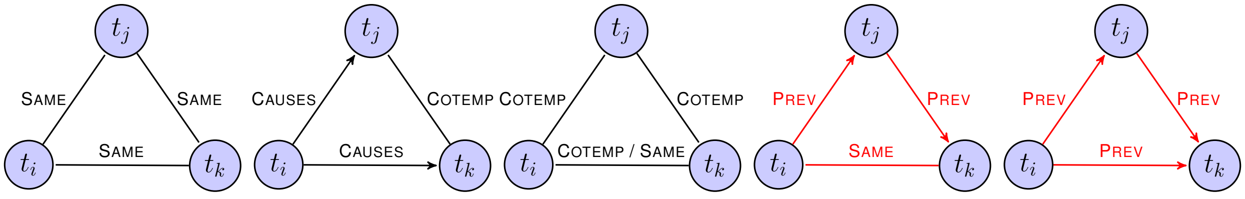
Table 1: Process statistics over 148 process descriptions. NONE is used to indicate no relation.

We annotated 148 process descriptions based on the aforementioned definitions, which can be downloaded from `http://to.be.filled.in`. 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 $\mathcal{P}$. First, in a coherent process, all events mentioned are related to one another, and hence the graph $\mathcal{P}$ must be connected. Second, processes tend to have a "chain-like" structure where one event follows another, and thus we expect nodes' degree to generally be ≤ 2. Indeed, 90% of event mentions have degree ≤ 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 $\mathbf{x}$ and a trigger set $\mathcal{T}$, 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



(a) SAME transitivity (b) CAUSE-COTEMP (c) COTEMP transitivity (d) SAME contradiction (e) PREV contradiction

Figure 2: Relation triangles (a)-(c) are common in the gold standard while (d)-(e) are impossible.

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.

Feature	Description
POS	Pair of POS tags
Lemma	Pair of lemmas
Prep*	Preposition lexeme, if in a prepositional phrase
Sent. count	Quantized number of sentences between triggers
Word count	Quantized number of words between triggers
LCA	Least common ancestor on constituency tree, if exists
Dominates*	Whether one trigger dominates other
Share	Whether triggers share a child on dependency tree
Adjacency	Whether two triggers are adjacent
Words btw.	For adjacent triggers, content words between triggers
Temp. btw.	For adjacent triggers, temporal connectives (from a small list) between triggers

Table 3: Features extracted for a trigger pair (t_i, t_j). Asteriks (*) indicate features that are duplicated, once for each trigger.

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 $f : \mathcal{T} \times \mathcal{T} \rightarrow \mathcal{R}$. As an example, $f(t_i, t_j) =$ PREV iff $f(t_j, t_i) =$ 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). Some features were omitted since they did not yield improvement in performance on a development set, or they require gold annotations provided in TimeBank, which we do not have. To reduce sparseness, we convert nominalizations into their verbal forms when computing word lemmas, using WordNet's (Fellbaum, 1998) derivation links.

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 bold, 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* $\xleftarrow{mark}$ *exchanged* $\xleftarrow{advcl}$ *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 (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 is a noun. We also add two features targeted at the 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 $\xrightarrow{dobj}\xrightarrow{rcmod}$ 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 process structure. Figure 3 (left) presents an example for predictions made by the pairwise classifier, which result in two triggers (*deleted* and *depleted*) 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 θ_{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 $\mathbf{y} = \{y_{ijr} \mid 1 \leq i < j \leq n, r \in \mathcal{R}\}$. With no global constraints this can be formulated as the following ILP:

$$\begin{gathered}\arg\max_{\mathbf{y}} \sum_{ijr} \theta_{ijr} y_{ijr} \\ \text{s.t.} \forall_{i,j} \sum_{r} y_{ijr} = 1\end{gathered} \tag{1}$$

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 $\bar{\mathcal{R}}$ be the set $\mathcal{R} \setminus$ 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 $y_{ij} := \sum_{r \in \bar{\mathcal{R}}} y_{ijr}$ 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 $\mathcal{P}$. We ensure that the edges in the tree exist also in $\mathcal{P}$ by tying each auxiliary variable to its corresponding ILP variable:

$$\forall_{i<j}\ z_{ij} \leq y_{ij}, z_{ji} \leq y_{ij} \tag{2}$$

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 ϕ_{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 $\mathbf{y}$ is connected.

Chain structure A chain is a connected graph where the degree of all nodes is ≤ 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, α_k is the penalty (or reward for desirable structures), and C_k indicates whether a constraint is violated (or satisfied). The parameters α_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 Equations 8-12 provide the corresponding ILP formulations. Equations 8-10 were formulated as soft constraints (expanding the set $\mathcal{K}$) and were incorporated by defining a reward α_k for each triad type. [2] 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

[2] 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.

that generate $\binom{5}{2}$ possible contradictions, but for brevity present just one representative hard constraint. This constraint depends on prediction 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.

.

$$y_{ij}\textsc{Same} + y_{jk}\textsc{Same} + y_{ik}\textsc{Same} \geq 3 \quad (8)$$

$$y_{ij}\textsc{Causes} + y_{ik}\textsc{Causes} + y_{jk}\textsc{Cotemp} \geq 3 \quad (9)$$

$$y_{ij}\textsc{Cotemp} + y_{jk}\textsc{Cotemp} + y_{ik}\textsc{Cotemp} + y_{ik}\textsc{Same} \geq 3 \quad (10)$$

$$y_{ij}\textsc{Prev} + y_{jk}\textsc{Prev} + y_{ik}\textsc{Same} \leq 2 \quad (11)$$

$$y_{ij}\textsc{Prev} + y_{jk}\textsc{Prev} - y_{ik}\textsc{None} \leq 1 \quad (12)$$

We used the Gurobi optimization package to find an exact solution for our ILP, which contains $O(n^2|\mathcal{R}|)$ 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 is quite fast (average/median time per process: 0.294 sec/0.152 sec).

4 Experimental Evaluation

Our data set consists of 148 process descriptions annotated by a biologist. The annotator was 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 (# of training set trigger pairs: 1932, # 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 marginal probability at edge (t_i, t_j) for label r, given by pairwise classifier.

	Temporal			**Full**		
	P	R	F$_1$	P	R	F$_1$
All-Prev	58.4	54.8	56.6	34.1	32.0	33.0
Local$_{base}$	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
Global	**63.9**	**61.4**†‡	**62.6**†‡	**56.2**	**54.0**†‡	**55.0**†‡

Table 4: Test set results on all experiments. Best number in each column is bolded. † and ‡ denote statistical significance ($p < 0.01$) against *Local$_{base}$* and *Local* baselines, respectively.

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) *Local$_{base}$*: A pairwise classifier with features from previous work (Section 3.1) (c) *Local*: A pairwise classifier with all features (Section 3.2) (d) *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 F$_1$. 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).

4.1 Results

Table 4 presents performance of all systems. We see that using global constraints improves performance on all measures in both full and temporal evaluations. Particularly, in the full evaluation recall improves by 12% and overall F$_1$ 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

Order	Parameter name	Value (α)	F_1 score
-	*Local model*	-	49.9
1	Connectivity constraint	∞	51.2
2	SAME transitivity	0.5	52.9
3	Chain constraint	-0.5	53.3
4	CAUSE-COTEMP	1.0	53.7
6	PREV contradiction	∞	53.8
7	SAME contradiction	∞	53.9

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 ∞ indicates a hard constraint.

$Local_{base}$. 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 *All-Prev* baseline performs badly in the full evaluation, but in temporal evaluation it works reasonably well. This demonstrates that process descriptions tend to proceed linearly from one event to the other, which is a known property of discourse structure (Schegloff and Sacks, 1973).

Table 2 presents the degree distribution of *Local* and *Global* on the development set comparing to the gold standard. Clearly, degree distribution of *Global* is much 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 F_1 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.

4.2 Qualitative Analysis

Figure 3 shows two examples where global constraints corrected the predictions of *Local*. In Figure 3, left, *Local* failed to predict the causal relations *skipped-deleted* and *used-duplicated*, possibly because they are not in the same sentence and are not adjacent to one another. By enforcing the connectivity constraint, *Global* correctly connects *deleted* and *duplicated* to the other triggers in the process.

In Figure 3, right, *Local* predicts a structure that results in a "SAME contradiction" structure. The triggers *bind* and *binds* cannot denote the same event if a third trigger *secrete* is temporally between them. However, *Local* predicts they are the same event, as they share a lemma. *Global* prohibits this structure and correctly predicts the relation as NONE.

To better understand the performance of *Local*, we analyzed the confusion matrix generated based on its predictions. Although this is a challenging 11-class classification task, most of the mass is concentrated on the matrix diagonal, as desired. Error analysis reveals that 17.5% of all errors are confusions between NONE and PREV, 11.1% between PREV and CAUSES, and 8.6% between PREV and COTEMP. This demonstrates that distinguishing the classes PREV, CAUSES and COTEMP is challenging for *Local*. Our current global constraints do not address this type of error, and thus an important direction for future work is to improve the local model.

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 *Local* produces a graph that is disconnected (e.g, *deleted* in Figure 3, left), then *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 *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

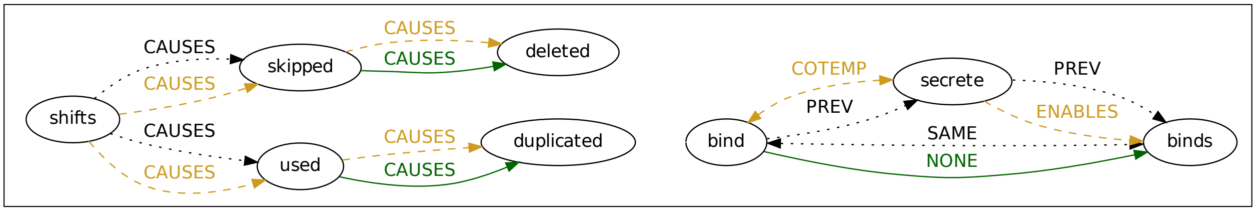


Figure 3: Process graph fragments. Black edges (dotted) are predictions of *Local*, green (solid) are predictions of *Global*, and gold (dashed) are gold standard edges. Original text, Left: "... the template *shifts* ..., and a part of the template strand is either *skipped* by the replication machinery or *used* twice as a template. As a result, a segment of DNA is *deleted* or *duplicated*." Right: "Cells of mating type A secrete a signaling molecule, which can *bind* to specific receptor proteins on nearby cells. At the same time, cells *secrete* factor, which *binds* to receptors on a cells."

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 applications that require deeper reasoning, such as building 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. Since data annotation is expensive, an important direction is to reduce the annotation burden by using data from similar domains or unannotated corpora. Lastly, we would like to integrate our method into QA systems and use the extracted process structure to answer non-factoid questions.

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Learning entailment relations by global graph structure optimization. *Journal of Computational Linguistics*, 38(1).

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 Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*.

Pascal Denis and Philippe Muller. 2011. Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. In *Proceedings of IJCAI*.

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of EMNLP-CoNLL*.

Jennifer D'Souza and Vincent Ng. 2013. Classifying temporal relations with rich linguistic knowledge. In *Proceedings of NAACL-HLT*.

Bradley Efron and Robert Tibshirani. 1993. *An introduction to the bootstrap*, volume 57. CRC press.

Christiane Fellbaum, editor. 1998. *WordNet: An electronic lexical database*. MIT Press.

Jenny Rose Finkel and Christopher D. Manning. 2008. Enforcing transitivity in coreference resolution. In *Proceedings of ACL*.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Junichi Tsujii. 2009. Overview of BioNLP 09 shared task on event extraction. In *Proceedings of BioNLP*.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, and Junichi Tsujii. 2011. Overview of BioNLP shared task 2011. In *Proceedings of BioNLP*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*.

André L. Martins, Noah A. Smith, and Eric P. Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of ACL/IJCNLP*, pages 342–350.

David McClosky and Christopher D. Manning. 2012. Learning constraints for consistent timeline extraction. In *Proceedings of EMNLP-CoNLL*, pages 873–882.

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. Castaño, 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 Sontag, Amir Globerson, and Tommi Jaakkola. 2011. Introduction to dual decomposition for inference. In Suvrit Sra, Sebastian Nowozin, and Stephen J. Wright, editors, *Optimization for Machine Learning*. MIT Press.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2011. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics*, 37(2).

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

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

Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. 2009. Jointly identifying temporal relations with Markov logic. In *Proceedings of ACL/IJCNLP*.