Extracting Biological Processes with Global Constraints

Author 1
XYZ Company
111 Anywhere Street
Mytown, NY 10000, USA
author1@xyz.org

Author 2
ABC University
900 Main Street
Ourcity, PQ, Canada A1A 1T2
author2@abc.ca

Abstract

Biological processes are complex phenomena involving a series of events that are related to one another through multiple dependencies. Teaching computers to read, understand and reason over text describing biological processes could dramatically improve performance of semantic applications such as question answering (QA). In this paper, we present the task of *process extraction*, in which events of a process and their relations 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 that graph (e.g., the graph is *connected*). Then, we present a method for extracting relations between processes, which exploits these structural properties by performing joint inference over the set of possible extracted relations. On a novel data set (released with this paper), containing 148 descriptions of biological processes 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. On top of that, it describes the role of each entity in each event, 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?"* is only possible given 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 in 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 specific events from a closed set in a single sentence. For example, the BioNLP 2009 and 2011 shared tasks (Kim et al., 2009; Kim et al., 2011) consider nine events types that are relevant for proteins. 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 that span multiple sentences. However, events in processes are tightly coupled in ways that go beyond simple temporal ordering, and these dependencies are central for the task of process extraction. Conse-

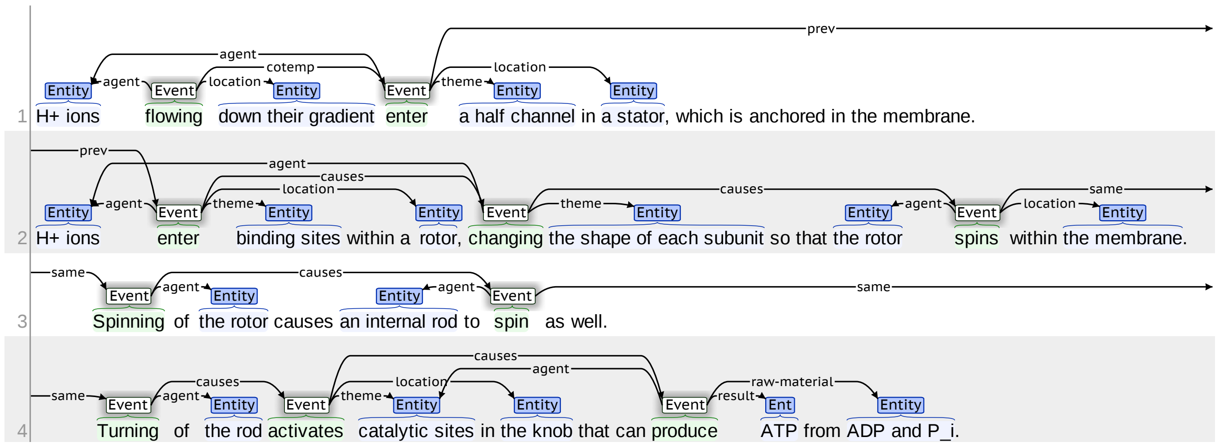


Figure 1: Partial annotation of the ATP synthesis process

quently, capturing process structure requires modeling a larger set of relations that includes, temporal, causal and coreference relations.

In this paper, we formally define the task of process extraction and present automatic extraction methods. Our approach works over multiple sentences and extracts a rich set of event-event relations, where the set of possible event types is open ended. Furthermore, we characterize a set of global properties in process structure that can be utilized during process extraction. For example, in processes all events are somehow connected to one another, and in addition processes usually exhibit a "chain-like" structure corresponding to process progression over time. We show that by incorporating global properties into our model and performing joint inference over the extracted relations, we can significantly improve process quality. Our empirical experiments are performed over a novel data set of 148 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 for domains other than Biology.

The three main contributions of this paper are:

1. We define process extraction and characterize their structural properties.
2. We show that modeling global structural properties significantly improves extraction accuracy.
3. We publicly release a novel data set of 148 fully annotated biological process descriptions.

2 Process Definition and Data Set

A process description is a paragraph or sequence of tokens $\mathbf{x} = \{x_1, ... x_{|\mathbf{x}|}\}$ describing a series of events that are related by various temporal and causal relations. For example, in ATP synthesis, the event in which the rotor spins *causes* the event where an internal rod spins.

We define the process events and their relations by a directed graph $\mathcal{P} = (V, E)$, where the nodes $V = \{1, ..., |V|\}$ represent event mentions and labeled edges correspond to event-event relations. An event mention $v \in V$ is defined by a trigger t_v, which is a span of words $x_i, x_{i+1}, ..., x_j$ 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 first event mention of ATP synthesis t_v = *flowing*, and one of the arguments is a_v = *(H+ ions,* AGENT*)*. 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 two steps 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

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

mentions. In this paper, we focus on the second task, 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 data set, 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, and RAW-MATERIAL describes an entity that is used or consumed during an event. For example, in the last event *'produce'* in Figure 1 *'ATP'* is the RESULT of the event, while *'ADP'* is the RAW-MATERIAL.

The relation set $\mathcal{R}$ contains the following relations (assuming an 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 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 co-refer to the same event (see Figure 1).

Our relation set contains the relations CAUSES and ENABLES, which are important for modeling processes and go beyond just temporal ordering.

	Avg	Min	Max
# of sentences	3.80	1	15
# of tokens	89.98	19	319
# of events	6.20	2	15
# of relations	5.64	1	24

Table 1: Process statistics over 148 process descriptions

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 it can not be for any w, that u is before w, but v is after w (see Section 3.3)

We have annotated 148 process descriptions based on the aforementioned definitions and provide further details on annotation and data set statistics in Section 4.1 and Table 1.

Structural properties of processes Naturally, coherent processes exhibit many structural properties. For example, two argument mentions related to the same event can not 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 node degree to generally be ≤ 2. Indeed, 90% of event mentions have degree ≤ 2, as is demonstrated by the first column of Table 2. Last, if we consider all possible relations between a triple of triggers, clearly some configurations are impossible, while other are quite common (illustrated in Figure 2). In Section 3.3, we will show how by modeling these properties we can improve process

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.

extraction using a joint inference framework.

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 (Section 3.1). Next, we describe novel features specific for process extraction (Section 3.2). Last, we incorporate global constraints into our model in an ILP formulation (Section 3.3).

3.1 Local pairwise classifier

The pairwise classifier predicts relations between all event mention pairs (represented by their triggers). Since some of the relations in $\mathcal{R}$ are directed, we must predict also the direction of these relations. We do this by expanding $\mathcal{R}$ to include the reverse of four directed relations: PREV-NEXT, SUPER-SUB, CAUSES-CAUSED, ENABLES-ENABLED. After adding NONE to indicate no relation, $\mathcal{R}$ contains 11 relations. Hence, the classifier is a function $f : \mathcal{T} \times \mathcal{T} \to \mathcal{R}$, where for instance $f(t_i, t_j) = \text{PREV}$ iff $f(t_j, t_i) = \text{NEXT}$. Let n be the number of triggers in a process description, and t_i be the i'th trigger appearing in the description, since $f(t_i, t_j)$ completely determines $f(t_j, t_i)$ it suffices to consider only pairs such that $i < j$. Note that in this new definition of $\mathcal{R}$ the process graph $\mathcal{P}$ is undirected.

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.

Feature	Description
POS	Pair of POS tags
Lemma	Pair of lemmas
Prep*	Preposition lexeme, if in a prepositional phrase
Words between	For adjacent triggers, content words between triggers
Temp. between	For adjacent triggers, temporal connectives (from a small list) between triggers
Adjacency	Whether two triggers are adjacent
# Sent.	Quantized number of sentences between triggers
# Word.	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

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

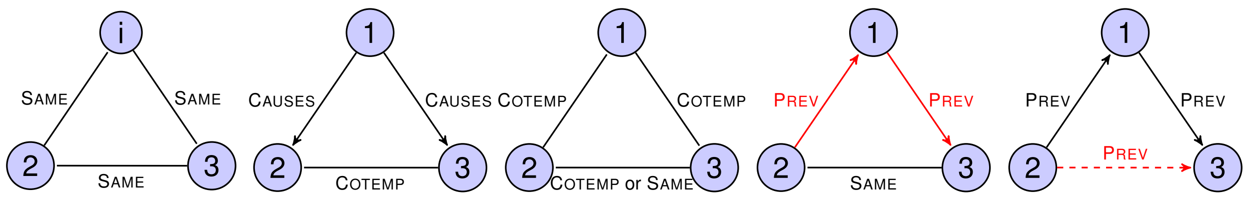
3.2 Classifier extensions

A central source of information for extracting event-event relations from text are *connectives* such as *after*, *during*, etc. However, there is variability in the occurrence of these connectives. Consider 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*.

Both sentences express the relation (*exchanged*, *reduced*, CAUSES), but the connective used is different, its linear position with respect to the triggers is different, and in sentence 1 the trigger *gene flow* intervenes between *exchanged* and *reduced*. Since our data set is very small, we would like to identify the triggers related to each connective, and share features between such sentences. We do this using the syntactic structure and a clustering of 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:



(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.

‘

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 ignore the trigger *gene flow* that is linearly closer to the trigger *exchanged*. After locating the relevant pair of triggers, we reduce sparseness utilizing a hand-made clustering of 30 connectives that maps words such as *because* and *since* to a "causality" cluster and fire a feature for this 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 we omit details here 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 in this sentence as well for the adjacent triggers *exchanged* and *reduced*.

We further extend our features to handle the rich relation set necessary for process extraction. Processes often begin with a trigger for an event that includes subsequent triggers, e.g., "The *Calvin cycle* begins by *incorporating*...". Thus, we add a feature for t_i indicating whether $i = 1$ and t_i is a noun. We also add two features targeted at the relation SAME: one indicating whether the lemmas of t_i and t_j are same, and another specifying the determiner of t_j, if it exists. Intuition is that certain determiners indicate that the event triggered had 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., the feature $\xrightarrow{dobj}\xrightarrow{rcmod}$ between *produces* and *divide* will fire in "meiosis produces cells that divide...". In Section 4.2 we will empirically show that our extension to the local classifier substantially improves performance

For our pairwise classifier, we train a maximum entropy classifier that provides 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, a pairwise classifier can result in a process structure that violates global properties. 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 process. In this section we incorporate into our model constraints that lead to a coherent global process structure.

Let θ_{ijr} be a score for the relation r and the triggers (t_i, t_j) (e.g, $\theta_{ijr} = \log p_{ijr}$), and y_{ijr} be the corresponding indicator. 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:

$$\arg\max_{\mathbf{y}} \sum_{ijr} \theta_{ijr} y_{ijr} \tag{1}$$

$$\text{s.t.} \forall_{i,j} \sum_{r} y_{ijr} = 1$$

where the constraint ensures each trigger pair is assigned exactly one relation. We now describe constraints that result in a process with a coherent global structure:

Connectivity Our formulation for enforcing connectivity is a minor variation on 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 is a directed tree that is a subgraph of $\mathcal{P}$ when edge directions are ignored. Thus the resulting formulation is almost identical. This formulation is based on flow constraints that 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 if there is some none-NONE relation between t_i and t_j: $y_{ij} = \sum_{r \in \bar{\mathcal{R}}} y_{ijr} = 1$. 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:

$$\forall_{i<j}\ z_{ij} \leq y_{ij}, z_{ji} \leq y_{ij} \qquad (2)$$

Next, we add constraints that enforce the graph structure induced by the auxiliary variables is a tree rooted in an arbitrary node 1 (The choice of root doesn't 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}.

$$\sum_i z_{i1} = 0, \forall_{j \neq 1} \sum_i z_{ij} = 1 \qquad (3)$$

$$\sum_i \phi_{1i} = n - 1 \qquad (4)$$

$$\forall_{j \neq 1}\ \sum_i \phi_{ij} - \sum_k \phi_{jk} = 1 \qquad (5)$$

$$\forall_{i \neq j}\ \phi_{ij} \leq n \cdot z_{ij} \qquad (6)$$

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 consumes exactly one flow unit. 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 connected graph where the degree of all nodes is ≤ 2 is a chain. Table 2 presents nodes' degree and demonstrates that indeed process graphs are close to being chains. The following constraint bounds nodes' degree by 2:

$$\forall_j \sum_{i<j} y_{ij} + \sum_{j<k} y_{jk} \leq 2 \qquad (7)$$

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. 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, and C_k indicates whether a constraint is violated. We tune the parameters α_k on a development set, as explained in Section 4.1.

Relation triangles A triangle is a 3-tuple of relations $(f(t_i, t_j), f(t_j, t_k), f(t_i, t_k))$. Clearly, some triangles are impossible while others are quite common. In order to look for triangles that could potentially improve process extraction we counted the frequency of all possible triangles in both the training data and the output of our pairwise classifier, and focused on those for which the classifier and the gold standard disagreed. We are interested in triangles that never occur in the training data but are predicted by the classifier, and vice versa. Figure 2 illustrates the triangles found and Equations 8-12 provide the corresponding ILP formulation. Soft constraints were incorporated by defining a reward α_k for each triangle type and expanding the set $\mathcal{K}$ accordingly[2].

1. SAME transitivity (Figure 2a): Co-reference transitivity has been used in past work (Finkel and Manning, 2008) and we incorporate it as a soft constraint that encourages triangles that respect transitivity:

$$y_{ijSAME} + y_{jkSAME} + y_{ikSAME} \geq 3 \qquad (8)$$

2. CAUSE-COTEMP (Figure 2b): 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

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

variation and an *increase* in the frequency of harmful alleles", a single event causes two subsequent events that occur simultaneously. We formulate this as a soft constraint:

$$y_{ijCAUSES} + y_{ikCAUSES} + y_{jkCOTEMP} \geq 3 \quad (9)$$

3. COTEMP transitivity (Figure 2c): 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. We formulate this as a soft constraint:

$$y_{ijCOTEMP} + y_{jkCOTEMP} + y_{ikCOTEMP} + y_{ikSAME} \geq 3 \quad (10)$$

4. SAME contradiction (Figure 2d): 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_i is before t_j, but t_k is after t_j. We define 5 temporal categories that generate $\binom{5}{2}$ possible contradictions, but for brevity present just one representative hard constraint. Note that this constraint depends on co-reference and temporal relations being predicted jointly.

$$y_{ijPREV} + y_{jkPREV} + y_{ikSAME} \leq 2 \quad (11)$$

5. PREV contradiction (Figure 2e): As mentioned (Section 3.3), if t_i is immediately before t_j, and t_j is immediately before t_k, then t_i is not immediately before t_k (hard constraint).

$$y_{ijPREV} + y_{jkPREV} - y_{ikNONE} \leq 1 \quad (12)$$

.

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

[3] www.gurobi.com

4 Experimental Evaluation

4.1 Experimental setup

Our data set consists of 148 process descriptions annotated by a biologist. The annotator was presented with annotation guidelines, annotated 20 descriptions and then annotations were discussed with the authors, after which all process descriptions were annotated. After training a second biologist, we measured inter-annotator agreement on 30 random process descriptions, resulting in agreement $\kappa = 0.69$.

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 triangle 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. Last, for our global model we defined $\theta_{ijr} = \log p_{ijr}$, where p_{ijr} is the probability given by the local classifier.

We test the following systems: (a) *All-Prev*: since the most common process structure is a chain of consecutive events we simply predict NEXT for every two adjacent triggers. (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

	Temporal			Full		
	P	R	F_1	P	R	F_1
All-Prev	62.2	58.3	60.2	34.1	32.0	33.0
$Local_{base}$	65.6	55.3	60.0	52.1	43.9	47.6
Local	66.2	58.3	62.0	54.7	48.3	51.3
Global	**67.1**	**64.5***	**65.8***	**56.2**	**54.0***	**55.0***

Table 4: Test set results on all experiments. Asterisk (*) denotes statistical significance ($p < 0.01$) against all other baselines. TODO compute stat significant for real

method (Efron and Tibshirani, 1993).

4.2 Results

Table 4 presents performance of all systems. Our main result is 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 full *Local* classifier substantially outperforms $Local_{base}$. 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 NEXT, CAUSES, and ENABLES when the amount of training data is small.

The *All-Prev* baseline performs quite badly in the full evaluation, but in temporal evaluation it performs reasonably well. This demonstrates the strong tendency process descriptions have to proceed linearly from one event to the other, which is a general 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 global constraints were introduced into the 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, which can be explained by examining the distribution of degrees in Table 2. The predictions of *Local* do not have many nodes with degree > 2 and thus the effect of this constraint is smaller. As for triangle constraints, we see that four constraints are included in the model but one is discarded since it did not improve F_1 on the development set.

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.

4.3 Qualitative Analysis

Figure 3 shows two examples where global constraints corrected predictions made by *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 a connectivity constraint, *Global* was able to connect the triggers *deleted* and *duplicated* to the other triggers in the process using the correct relations.

In Figure 3, right, *Local* predicts a triangle that violates the "SAME contradiction" constraint. The triggers *bind* and *binds* cannot denote the same event if a third trigger *secrete* is temporally between them. However, since *bind* and *binds* share the same lemma, *Local* predicts that they are co-referring triggers. Global constraints prohibit this solution and as a result *Global* modifies the solution and correctly predicts NONE.

In order to understand the performance of the local model, we analyzed the confusion matrix generated based on the predictions of the local model.

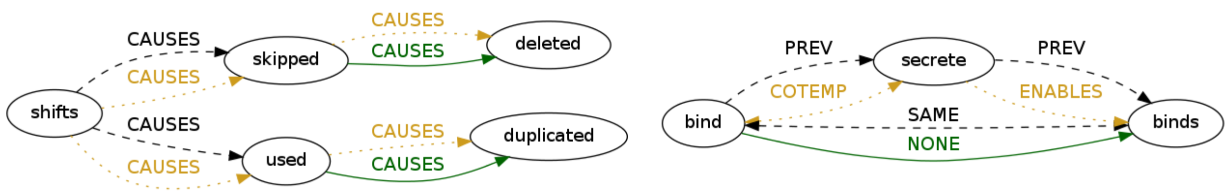


Figure 3: Fragments of process graphs. Black edges (dashed) are predictions of *Local*, green edges (solid) indicate edges modified by *Global*, and gold edges (dotted) represent gold standard edges. Original text, Left: "... the template *shifts* with respect to the new complementary strand, and a part 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."

Even though it is a 11 class classification task, most of the mass is concentrated along the diagonal, indicating that the model performs well. On examining the most common mistakes made by the model, we saw that 17.5% of all errors were confusion between NONE-PREV, 11.1% between PREV-CAUSES, and 8.6% were confusion between PREV-COTEMP. This is because the subtle distinctions between PREV, CAUSES and COTEMP are sometimes hard to capture using features in the local model. At the same time, once the local model makes incorrect predictions, the global model cannot effect a lot of changes to correct everything. As a result, as future work, focus should be given on obtaining more data that will help the model learn stronger features to capture the small differences between classes. In addition, we can also enhance the local model with more features like common phrases denoting temporal/causal relations.

Another interesting point of discussion is how the global constraints can affect the overall result. The performance of the global model largely depends on the predictions made by the local classifier. Also, enforcing the global constraints does not always guarantee improvement on generated graph structures. For instance, if the local classifier predicts a graph structure that is not connected (like *deleted* in Figure 3, left), the connectivity constraint of the global model will force an edge from/to any other node. Now, if the new edge results in an incorrect prediction, we get penalized twice - one for the false positive of the predicted class and one for the false negative of the actual class, instead of just one before. In spite of this caveat, the global model gives us an improvement of 3.7 F_1 points on the test set.

5 Related Work

As previously mentioned, 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 proposed to predict 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 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. Do et al. (2012) modeedl a set of global temporal order constraints also using ILP for timeline construction. There is 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 read and extract meaning from process descriptions is an important step towards applications that require deep reasoning, such as non-factoid QA. 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 mutual dependencies. We have shown that by exploiting these structural dependencies and performing joint inference over all event mentions we can significantly improve accuracy comparing to several baselines. We have also released a new data set containing 148 fully annotated descriptions of biological processes.

We assumed in this paper that event triggers are given as input. In future work we would like to perform trigger identification jointly with extraction of event-event relations. Because data annotation is expensive, another important direction is to reduce annotation burden by using data from similar domains or large unannotated corpora. Last, we would like to combine our method in a QA system that uses the extracted structure to answer non-factoid questions that are unanswerable by current state-of-the-art systems.

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*.