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

Reasoning over processes is fundamental for language understanding applications such as Question Answering. In this paper we propose a method for extracting relations between events in a process. We annotate 150 paragraphs describing biological processes and show that by taking advantage of the global structure of a process we can substantially improve performance. In addition, we release our data set.

1 Introduction

Processes describe complicated phenomena that involve a series of events and multiple participants. Automatically extracting the structure of processes is necessary for text understanding applications that require reasoning over process events. Consider, for example, the paragraph in Figure 1, which describes the biological process of ATP synthesis. A human reading this paragraph can create a mental model that allows her to answer questions such as:

1. *How do H+ ions contribute to the production of ATP?*
2. *What causes the rotor to spin?*
3. *In ATP synthesis, what happens if the rotor fails to spin?*

All these questions depend on extraction of the process structure and reasoning over the causal and temporal relations between the process events. Question answering systems that rely on bag-of-words representations will fail to correctly answer such questions.

Process extraction is related to two recent lines of works in Information Extraction – event extraction and timeline construction. The BioNLP 2009 and 2011 shared tasks (Kim et al., 2009; Kim et al., 2011) led to increasing interest in biomedical event extraction (Poon and Vanderwende, 2010; Miwa et al., 2010; Riedel and McCallum, 2011; McClosky et al., 2011; Björne et al., 2011), where given a single sentence annotated with protein mentions, events are identified and relations between events and proteins are extracted. In this shared task participants were asked to consider nine event types that are relevant for proteins (such as *Phosphorilation* and *Transcription*). Processes, on the other hand, are centered around discovering relations between several event mentions. Thus, process descriptions usually span multiple sentences, and must handle both an open-ended set of event types as well as a rich set of event-event relations.

Timeline construction involves identifying temporal relations between a collection of events (Chambers and Jurafsky, 2008; Yoshikawa et al., 2009; Denis and Muller, 2011; Do et al., 2012; McClosky and Manning, 2012), and is thus related to process extraction as both focus on event-event relations that span multiple sentences. However, fully capturing process structure requires handling a rich set of relations such as CAUSES and SUPEREVENT (see Section 3), which are often not addressed in timeline construction. Moreover, processes exhibit particular properties that do not hold generally in temporal ordering. For example, in processes all events are somehow related to one another, a property that can be exploited for improving extraction.

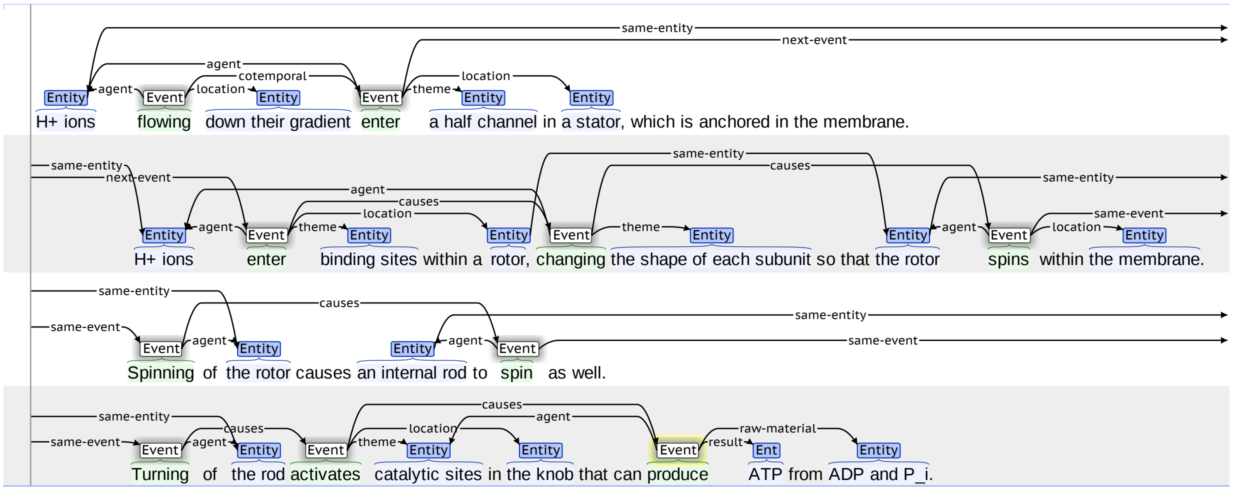
same-entity
next-event
agent
cotemporal
location
theme
Entity
Event
H+ ions flowing down their gradient enter a half channel in a stator, which is anchored in the membrane.
same-entity
next-event
agent
causes
location
theme
same-event
H+ ions enter binding sites within a rotor, changing the shape of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

Figure 1: An annotation of the ATP synthesis process

In this paper, we present the task of process extraction and describe methods for extracting relations between process events. Our method works over multiple sentences and extracts a rich set of event-event relations, where the set of possible event types is open ended. Process structure is characterized by global properties that can be utilized during process extraction. For example, most processes exhibit a “chain-like” structure corresponding to process progression over time, and all process events are connected to one another, as previously noted. We will 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 150 process descriptions from the textbook “Biology” (Campbell and Reece, 2005) that were annotated by trained biologists. We note however that our method does not utilize any domain-specific knowledge and thus can be easily applied to domains other than Biology.

To conclude, this paper presents the following three contributions:

1. We define the task of process extraction and characterize the structural properties of processes.

2. We show that by modeling structural properties we can significantly improve the quality of extracted processes comparing to several baselines.

3. We publicly release a novel data set of 150 fully annotated biological process descriptions.

2 Related Work

BioNLP work

Timeline construction work.

Scripts work - Chambers, Poon 2013.

Work that uses global constraints with ILP or dual decomposition or whatever.

3 Process Definition and Data Set

A process description is a paragraph or sequence of tokens $\mathbf{x} = \{x_1, ...x_{|x|}\}$ 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 an event 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 structure 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 event 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 task, where given a set of event triggers $\mathcal{T}$, we find all event-event relations. For completeness, we now describe the set of 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 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. NextEvent denotes that v is an event immediately following u. Thus, the edges $(u, v, \text{NextEvent})$ and $(v, w, \text{NextEvent})$, preclude the edge $(u, w, \text{NextEvent})$. For example, in “When a photon *strikes* ... energy is *passed* ... until it *reaches* ...”, there is no edge (*strikes*, *reaches*, NextEvent) due to the intervening event *‘passed’*.

2. Cotemporal denotes that events u and v overlap over time (e.g., the first two event mentions in Figure 1).

3. SuperEvent denotes that event u is included in event u. For instance, the process for “During *DNA replication*, DNA polymerases *proofread* each nucleotide...” has the edge (*DNA replication*, *proofread*, SuperEvent).

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 process “... cause cancer cells to *lose* attachments to neighboring cells..., allowing them to *spread* into nearby tissues” has the edge (*lose*, *spread*, Enables).

	Avg	Min	Max
# of tokens			
# of events			
# of relations			

Table 1: Statistics over the 150 process descriptions

6. SameEvent 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 temporal ordering only. We defined that whenever these two relations apply they override the temporal relation (which is invariably NextEvent). The SuperEvent 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 (SameEvent) 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 SameEvent 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 $(u, v, \text{SameEvent})$, then it can not be for any w that u is before w, but v is after w (see Section 4.2)

We have annotated 150 process descriptions based on the aforementioned definitions and provide further details on annotation and data set statistics in Section 5.4 and Table 1.

Structural properties of processes Naturally, coherent processes exhibit many structural properties. For example, two argument mentions related to the same event trigger can not overlap – a constraint that has been used in the past in SRL (Toutanova et al.,

[1] Argument mentions can also be related by coreference, but we neglect that since it is not central to this paper.

Deg.	Gold	Local	Global
0			
1			
2			
3			

Table 2: Node degree count for event mentions across the process descriptions

2008). In this paper we focus on three main structural properties of the graph $\mathcal{P}$. First, in a coherent process all event 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. Thus, we expect node degree to generally be ≤ 2, and this is indeed the case as demonstrated by the first column in Table 2. Last, if we consider all possible relation triangles, clearly some triangles are impossible, while other are common, which is illustrated in Figure **??**. In Section 4.2, we will show how using these properties we can improve process extraction, by formulating the problem as an ILP with both hard and soft constraints, and performing joint inference.

Next we describe our model for extracting event-event relations, given a set of event triggers $\mathcal{T}$. We first describe a pairwise model that classifies each pair of events independently of other, and then present our full model that performs joint inference over $\mathcal{R}$.

4 Joint Model for Process Extraction

Our task is given a paragraph $\mathbf{x}$ and a set of event mention triggers $\mathcal{T}$, to extract all event-event relations E. Similar to Do et al. (2012) our model consists of two parts. In section 4.1, we use a local pairwise classifier that considers each pair of triggers, and then in Section 4.2 we incorporate global constraints in an ILP formulation.

4.1 Local pairwise classifier

The pairwise classifier predicts relations between all event mention pairs (represented only by their triggers). Since relations in $\mathcal{R}$ are directed, we must predict also the direction of each relation. We do this by expanding $\mathcal{R}$ to include reverse relations and so we re-define $\mathcal{R}$ to include 11 relations: NextEvent, PreviousEvent, Cotemporal, SuperEvent, SubEvent, Causes, Caused, Enables, Enabled, SameEvent, None, where None indicates no relation. Thus, our classifier is a function $w : \mathcal{T} \times \mathcal{T} \rightarrow \mathcal{R}$. Let n be the number of event triggers in a process description, and t_i be the i'th trigger appearing in the description, since $w(t_i, t_j)$ completely determines $w(t_j, t_i)$ it suffices to consider only pairs such that $i < j$.

We have constructed an initial set of features based on previous work in temporal ordering (Chambers and Jurafsky, 2008; Do et al., 2012). However, since our training set is quite small and we consider a larger set of relations we have defined some novel features that improved performance (see Section 5.4. We note that contrary to work in BioNLP, we did not employ any biological dictionaries.

Baseline features[2]

- lemma pairs including WordNet normalization to verbs, POS paris
- for consecutive event mentions - words between
- for consecutive event mentions - temporal connectives
- POS pair
- quantized num of sentences and words
- lowest common ancestor
- one dominates the other
- share child

Novel features

- for consecutive event mentions - clusterings
- existence of and (not important)
- - first and nominalization - for SuperEvent
- - same lemma - for SameEvent (includes nominialization)

[2] Some features from (Chambers and Jurafsky, 2008; Do et al., 2012) were not included since they did not improve performance on the development set.

- determiner before the second event - for SameEvent
- dependency path between them
- syntactic marker features including clustering

Talk about the fact that it is important to have the clustering and syntactic features to share stats.

4.2 Global Constraints

[I think we should have a short experiment with soft constraints on the degree of nodes, I think this will add some substance even if our intuition is that it might not work I don't think it is a lot of work]

Motivation paragraph - naturally there are cases where local decision can lead to global structures that don't make sense. Give examples - one for something that is a hard constraint and something for soft constraints. Maybe we should have a figure with examples for bad local predictions - connectivity and triads.

Define the notations for formulating the objective function and formulate the objective function (variables are indicators e_{ijr}). Our formulation will probably have in the objective the local model scores and the soft constraints. Our hard constraints are those to make the formulation make sense and the hard constraints. Say that we use log probabilities from the pairwise classifier as our weights in the model.

Then we describe the modeling constraints.

Connectivity - short explanation. and then show the formulation which is a slight variation on (Martins et al., 2009). Refer to the motivating figure.

chain-like - we did not implement this because we didn't think it would help but I think it is easy to formulate and experiment with this. Explain the motivation. Refer to the table that shows that the local classifier is doing already pretty well and say that later we show whether this helps or not in the process of choosing soft constraints.

Triads - some triangles are not meaningful. To better understand what things are predicted by the model but are not in the gold and vice versa we counted and compared. Table... shows the top K of these. These guided us to engineer constraints that will improve the local classifier

Now we go over each one of the triad constraints we experimented with (even if some got 0 weight at the end) We explain the motivation and show the hard constraint formulation mentioning that turning them into soft is easy. We have to not make this boring so try to explain with examples.

Say something about the number of variables and constraints. Say that we use Gurobi ILP solver. Say that in principle one can use dual decomposition methods but in practice for this work we found ILP was fast enough.

Tuning of the soft constraints parameters - should we talk about this here or in the experimental section - probably in the experimental setting part

5 Experimental Evaluation

5.1 Experimental setup

Annotation Talk a bit about annotation of the data. Talk about the split to train and dev. Explain that the dev was used for feature selection in the local classifier and for tuning the parameters for global constraints. Explain that these parameters were chosen with coordinate ascent. Explain what are the values we tried and what are the values that were chosen - if we want we can have a table for the way performance increased on the dev set and what were the values that were chosen. might not be crucial.

Talk about the baselines (A) always next (B) Simple local (C) full local (D) local with chain structure (E) global model

Talk about evaluation measures. (a) full (b) collapsed. Maybe talk about the double-counting problem and we do nothing about it. We have to decide if to use only micro or also macro.

5.2 Results

Have a table with all results and discuss. We can see if interesting to have train/dev/test results. Maybe we can also have a confusion matrix for the final model to see that there are difficulties distinguishing cause-next-cotemp which are harder to do with global constraints and require more work on local features or more data.

Maybe we can have a table with ablations for the new features we added to see which helps? not sure necessary.

what other tables and figures can we have?

5.3 Analysis and Discussion

What other interesting stats we can put? I think it would be good to have some interesting example for something that got corrected and also something that we did not correct. It's alway nice to have some manual error analysis for intuition.

5.4 Full pipeline

If we have this we can briefly explain about our first step system and show some results. This is good to say we do everything and bad if this really sucks.

6 Conclusion

In this paper we presented the task of process extraction and a method for extracting processes. We focused on extracting relations between event triggers. We also release publicly a data set for the scientific community. We have shown that by taking advantage of the global structure of a process we can improve performance.

Future work - adding more constraints - Mengqiu's idea. This may results in inference problems (it does) and so we can try think of smarter inference. There is the problem of very little data and we can think about using data from other domains and do adaptations. We want to do the full pipeline jointly.

References

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

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 *EMNLP*.

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

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *EMNLP-CoNLL*, pages 677–687.

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*.

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

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

David McClosky, Mihai Surdeanu, and Christopher D. Manning. 2011. Event extraction as dependency parsing. In *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 *HLT-NAACL*, pages 813–821.

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*, pages 28–34.

Sebastian Riedel and Andrew McCallum. 2011. Fast and robust joint models for biomedical event extraction. In *Proceedings of the Conference on Empirical methods in natural language processing (EMNLP '11)*.

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

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