

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 *Phosphorylation* 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.

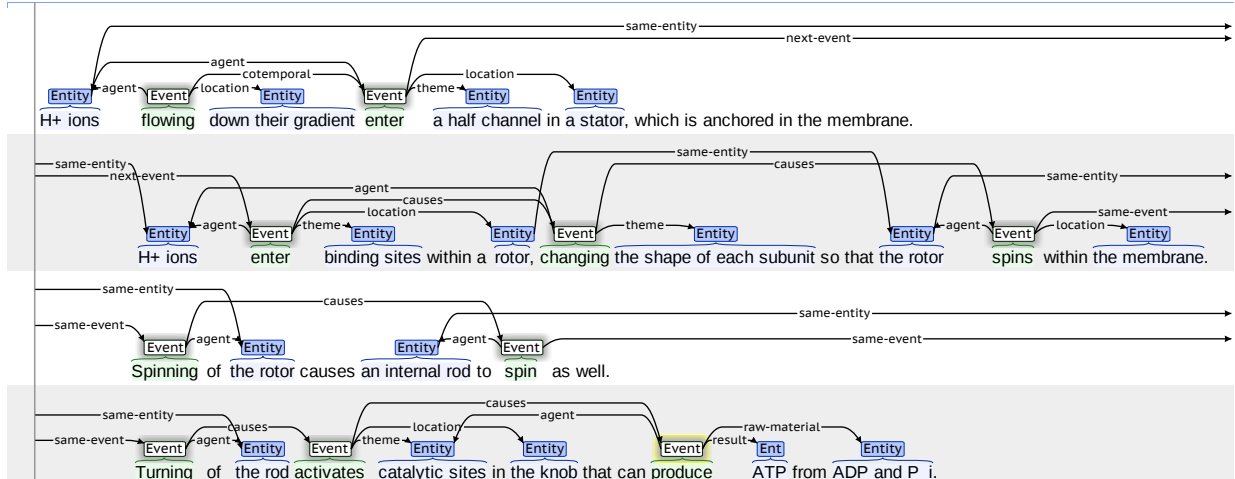


Figure 1: An annotation of the ATP synthesis process

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 $\mathbf{x} = \{x_1, \dots, x_{|x|}\}$ 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 $V = \{1, \dots, |V|\}$ 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}, \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 first event mention of ATP synthesis $t_v = \text{flowing}$, and one of the arguments is $a_v = (\text{H+ ions}, \text{AGENT})$. 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 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 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 $(\text{strikes}, \text{reaches}, \text{NEXTEVENT})$ 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 v . For instance, the process for “During *DNA replication*, DNA polymerases *proofread* each nucleotide...” has the edge $(\text{DNA replication}, \text{proofread}, \text{SUPEREVENT})$.
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

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

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

Table 1: Statistics over the 150 process descriptions

example, the process “... cause cancer cells to *lose* attachments to neighboring cells..., allowing them to *spread* into nearby tissues” has the edge $(\text{lose}, \text{spread}, \text{ENABLES})$.

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 4.3)

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 mention 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

Deg.	Gold	Local	Global
0			
1			
2			
3			

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

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 4.3, 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.

4 Joint Model for Process Extraction

Given a paragraph 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 4.1). Next, we describe novel features specific for process extraction (Section 4.2). Last, we incorporate global constraints into our model in an ILP formulation (Section 4.3).

4.1 Local pairwise classifier

The pairwise classifier predicts relations between all event mention pairs (represented 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, PREVIOUS EVENT, COTEMPORAL, SUPEREVENT, SUBEVENT, CAUSES, CAUSED, ENABLES, ENABLED, SAMEEVENT, NONE, where NONE indicates no relation. Thus, our classifier is a function $f : \mathcal{T} \times \mathcal{T} \rightarrow \mathcal{R}$. Let n be the number of triggers in a process description, and t_i be the i ’th trigger ap-

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
# 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.

pearing in the description, since $f(t_i, t_j)$ completely determines $f(t_j, t_i)$ it suffices to consider only pairs such that $i < j$.

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.

4.2 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 so genetic differences are *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 dependency structure and a clustering of connectives.

[Details on clustering and syntactic feature implementation probably using example above, maybe explain using "marker", then say something that we do something similar for things that are under PP but do not describe for simplicity, and say that for advmod we only use the clustering]

We further extend our features to handle the rich relation set necessary for process extraction. Processes often begin with a trigger that is a SUPEREVENT of the 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 SAMEEVENT: one indicating whether the lemmas of t_i and t_j are equal, 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 SAMEEVENT 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 $\xrightarrow{dobj} \xrightarrow{rmod}$ between *produces* and *divide* will fire in "meiosis produces cells that divide...".

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

4.3 Global Constraints

Naturally, a pairwise classifier can result in a process structure that violates the global properties described in Section 3. Figure ?? shows in black edges the predictions of our local classifier, which result in the trigger [it'd be good to find a structure that combines transitivity violation and triad violation and re-

fer to it] In this section we incorporate constraints that result in a coherent global process structure.

Let θ_{ijr} denote a score for whether the relation r holds between triggers (t_i, t_j) (e.g, $\theta_{ijr} = \log p_{ijr}$), and y_{ijr} be an indicator for this relation. Our goal is to find an assignment for the indicators $\mathbf{y} = \{y_{ijr} \mid 1 \leq i < j \leq n, r \in \mathcal{R}\}$. Without global constraints this can be formulated with the following ILP:

$$\begin{aligned} \arg \max_{\mathbf{y}} \quad & \sum_{ijr} \theta_{ijr} y_{ijr} \\ \text{s.t.} \quad & \forall_{i,j} \sum_r y_{ijr} = 1 \\ & \forall_{i,j,r} y_{ijr} \in \{0, 1\} \end{aligned} \tag{1}$$

where the constraint ensures each trigger pair is assigned to exactly one relation. We now describe constraints that result in a process that is connected, "chain-like", and contains reasonable relation triangles.

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 would like a connected graph, but not necessarily a tree, however since an undirected graph is connected iff there is a directed tree embedded in it, the resulting formulation is almost identical. This formulation is based on flow constraints that ensure that there is a path from a root in the graph to all other nodes.

Let $\bar{\mathcal{R}}$ be the set of relations except for NONE, i.e., $\mathcal{R} \setminus \text{NONE}$. An edge (t_i, t_j) is in E if $y_{ij} = \sum_{r \in \bar{\mathcal{R}}} y_{ijr} = 1$. For each variable y_{ij} we define two auxiliary binary variables z_{ij} and z_{ji} that correspond to the directed edges in the embedded tree. We tie each auxiliary variable to its corresponding ILP variable by ensuring that it is active only if the ILP variable is active:

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

Next, we add constraints that enforce the structure of the auxiliary variables results in a tree rooted in node 1. For that we need for every $i \neq j$ a flow variable ϕ_{ij} which specifies the amount of flow on the directed edge z_{ij} .

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

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

$$\forall_j \sum_i \phi_{ij} - \sum_k \phi_{jk} = 1 \quad (5)$$

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

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 our 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 demonstrated that indeed process graphs are close to being chains. The following constraint bounds node’s degree by 2:

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

Since graph structures are not always chains we add this as a soft constraint, that is, we penalize the objective for each violation. 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 constraints, $\alpha_k < 0$ 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 5.4.

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 how many time each possible triangle occurs in both the training data and the output of our pairwise classifier, and focused on those for which the classifier and the gold standard disagree. We are interested in

triangles that never occur in the training data but our predicted by the classifier, and triangles that are frequent in the gold standard but do not appear in the classifier output. Figure ?? illustrates the triangles we found and we next describe their ILP formulation:

SAMEEVENT transitivity: SameEvent or coreference is a transitive relation (Figure ??), and this has been exploited in previous work (Finkel and Manning, 2008). We encourage/penalize SAMEEVENT transitivity by introducing a soft constraint... (if this is rewarding then explain that this is integrated in the same way as penalizations).

COTEMPORAL transitivity: When 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 (Figure ??). We encourage this construction by introducing a soft constraint....

CAUSE-COTEMPORAL: When t_i causes both t_j and t_k , then often t_j and t_k are co-temporal (Figure ??). A typical example is “*genetic drift* has led to a *loss* of genetic variation and an *increase* in the frequency of harmful alleles”, where a single event causes two subsequent events that occur simultaneously. We encourage this construction with the constraint

SAMEEVENT contradiction: When t_i is the same event as t_j , then their temporal ordering with respect to a third trigger t_k may result in a contradiction, e.g. if t_i is before t_k , but t_j is after t_k (Figure ??). We derive five temporal categories from \mathcal{R} : PAST (corresponding to PREVIOUS EVENT, CAUSES, and ENABLES), FUTURE (corresponding to NEXT EVENT, CAUSED, and ENABLED), COTEMPORAL, SUPEREVENT, and SUBEVENT, and define a contradiction when the relations (t_i, t_j) and (t_i, t_k) are from different temporal categories. We prohibit such constraints but introducing hard constraints for all possible contradictions, where a representative one is the following:

PREVIOUS EVENT: As mentioned in Section 4.3, if t_i is immediately before t_j , and t_j is immediately before t_k , then it can not be that t_i is immediately before t_k . We prohibit such constructions with a hard constraint:

Overall, our ILP contains $O(n^2 r)$ variables and $O(n^3)$ constraints, and we used the Gurobi opti-

mization package³ to find an exact solution for our ILP. We have also developed an equivalent formulation that is amenable to dual decomposition methods (Reichart and Barzilay, 2012), which is a faster approximation method, but practically we found that solving the problem exactly with Gurobi is quite fast (average/median time per process: XX/YY).

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 always 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.

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

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.
- Christiane Fellbaum, editor. 1998. MIT Press.

³www.gurobi.com

- Jenny Rose Finkel and Christopher D. Manning. 2008. Enforcing transitivity in coreference resolution. In *ACL (Short Papers)*, pages 45–48.
- 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.
- Roi Reichart and Regina Barzilay. 2012. Multi-event extraction guided by global constraints. In *HLT-NAACL*, pages 70–79.
- 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*.