Inter-Event Dependencies support Event Extraction from Biomedical Literature

Roman Klinger
Joint Work w/ Sebastian Riedel and Andrew McCallum

9th September 2011
MIND Workshop at ECML-PKDD 2011

Overview

1. The BioNLP 2009 Shared Task on Event Extraction
   - Task Definition
   - Examples
   - Approaches and Motivation
   - Results
2. Imperatively Defined Factor Graphs (IDF)
   - Factor Graphs
   - Templates
   - FACTORIE
3. Document Wide Inference for the BioNLP Shared Task with IDF
   - Variables, Data Structure, Templates, Sampling, Objective
4. Results
5. Summary

Overview

1. The BioNLP 2009 Shared Task on Event Extraction
   - Task Definition
   - Examples
   - Approaches and Motivation
   - Results
2. Imperatively Defined Factor Graphs (IDF)
   - Factor Graphs
   - Templates
   - FACTORIE
3. Document Wide Inference for the BioNLP Shared Task with IDF
   - Variables, Data Structure, Templates, Sampling, Objective
4. Results
5. Summary

The BioNLP 2009 Shared Task on Event Extraction

- BioNLP Competition
- Data Set based on Genia Corpus

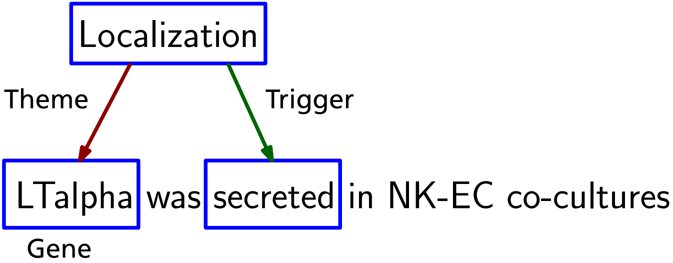
The BioNLP 2009 Shared Task on Event Extraction

- BioNLP Competition
- Data Set based on Genia Corpus

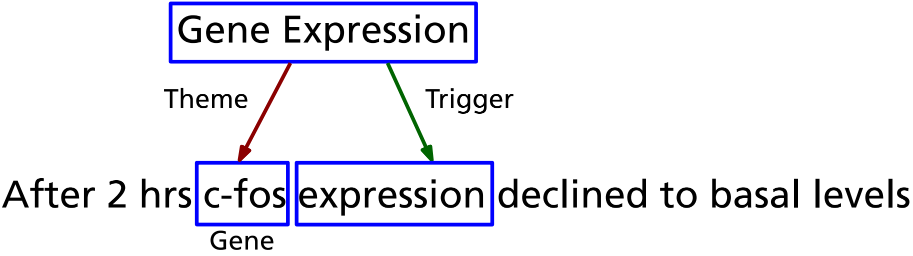
Task

- Extract event descriptions from biomedical abstracts
- Events of interest
  - Gene Expression, Transcription, Protein Catabolism, Phosphorylation, Localization
  - Binding
  - Positive Regulaton, Negative Regulation, Regulation

BioNLP '09 Shared Task – Examples (1)

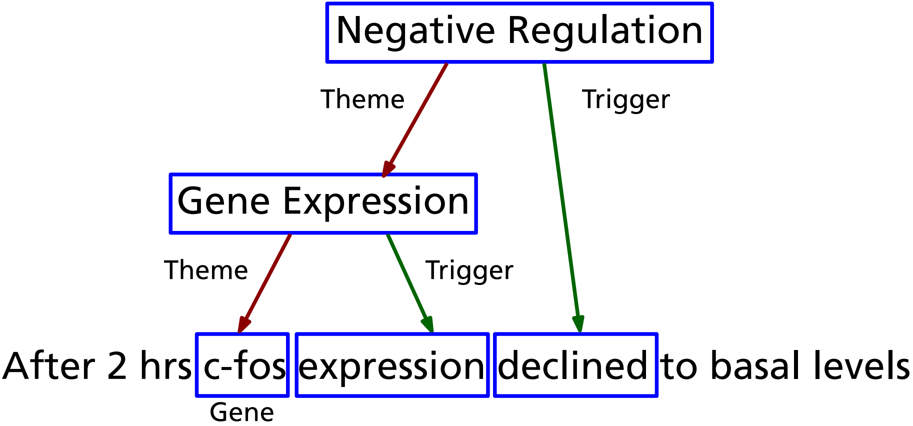


BioNLP '09 Shared Task – Examples (2)



Fraunhofer SCAI UMASS

BioNLP '09 Shared Task – Examples (2)



BioNLP ’09 Shared Task – Examples (3)

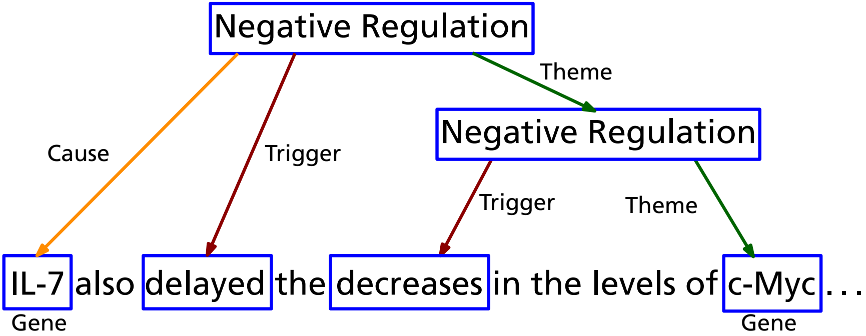
Negative Regulation

Trigger Theme

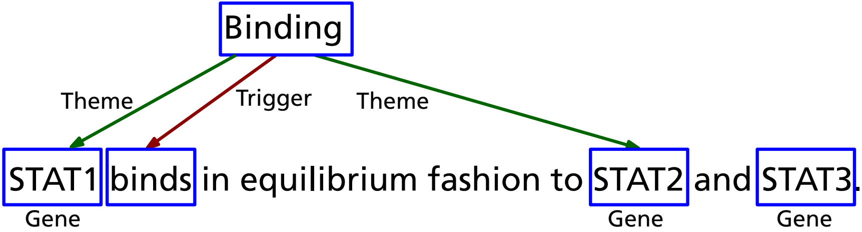
IL-7 also delayed the decreases in the levels of c-Myc . . .

Gene

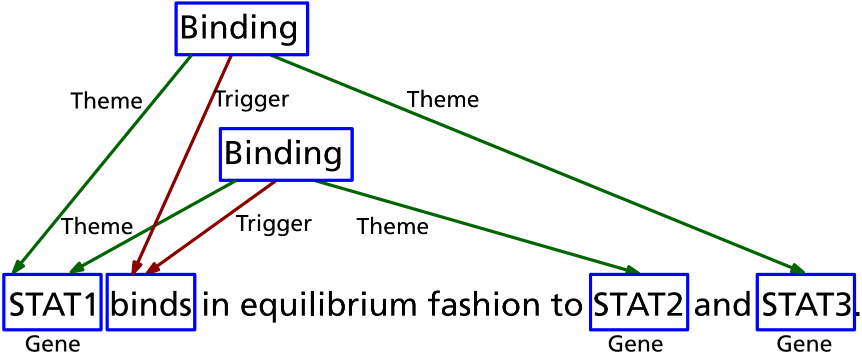
BioNLP '09 Shared Task – Examples (3)



BioNLP '09 Shared Task – Examples (4)



BioNLP '09 Shared Task – Examples (4)



BioNLP '09 Shared Task – Different Approaches

- Multi-Step Workflow using classifiers (SVM. . .)
  1. Detect Trigger Words
  2. Attach Arguments

  - Information flow in one direction

  ⇒ Propagation of errors!

BioNLP '09 Shared Task – Different Approaches

- Multi-Step Workflow using classifiers (SVM. . .)
  1. Detect Trigger Words
  2. Attach Arguments

  - Information flow in one direction

  ⇒ Propagation of errors!
- Models solving the whole task jointly
  - Information flow in all directions

  ⇒ No propagation of errors!

BioNLP '09 Shared Task – Different Approaches

- Multi-Step Workflow using classifiers (SVM...)
  1. Detect Trigger Words
  2. Attach Arguments

  - Information flow in one direction

  $\Rightarrow$ Propagation of errors!
- Models solving the whole task jointly
  - Information flow in all directions

  $\Rightarrow$ No propagation of errors!
- Neglected:
  - Modelling inter-event characteristics

BioNLP 2009 Shared Task – Results

	TEAM	F_1
–	Riedel2011	57.4
–	Miwa2010	56.3
–	Bjoerne2009	52.0
1	UTurku	52.0
–	Poon2010	50.0
–	McClosky2011	48.6
2	JULIELab	46.7
3	ConcordU	44.6
4	UT+DBCLS	44.4
5	VIBGhen	40.5
6	UTokyo	36.9
7	UNSW	34.9
8	UZurich	34.8
9	ASU+HU+BU	32.1

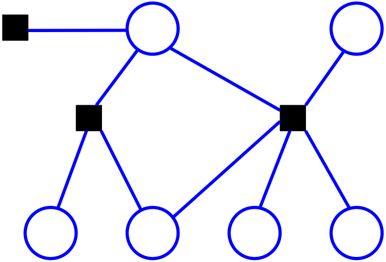
	TEAM	F_1
10	Cam	30.8
11	UAntwerp	30.6
12	UNIMAN	30.6
13	SCAI	30.3
14	UAveiro	29.4
15	Team 24	29.1
16	USzeged	27.2
17	NICTA	24.3
18	CNBMadrid	24.2
19	CCP-BTMG	22.7
20	CIPS-ASU	20.7
21	UMich	19.3
22	PIKB	19.3
23	Team 09	17.0
24	KoreaU	16.3

Overview

1. The BioNLP 2009 Shared Task on Event Extraction
   - Task Definition
   - Examples
   - Approaches and Motivation
   - Results
2. Imperatively Defined Factor Graphs (IDF)
   - Factor Graphs
   - Templates
   - FACTORIE
3. Document Wide Inference for the BioNLP Shared Task with IDF
   - Variables, Data Structure, Templates, Sampling, Objective
4. Results
5. Summary

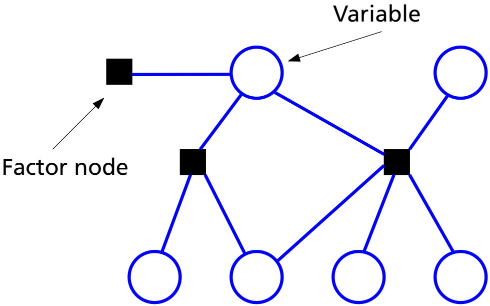
Factor Graph

A Factor Graph is a bipartite graph over factors and variables



Factor Graph

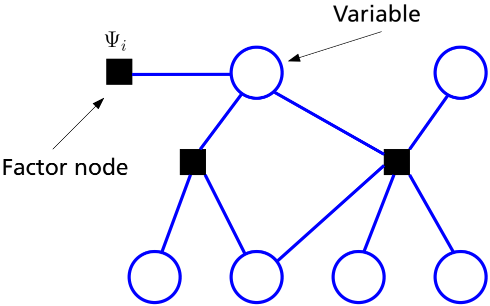
A Factor Graph is a bipartite graph over factors and variables



Factor Graph

A Factor Graph is a bipartite graph over factors and variables

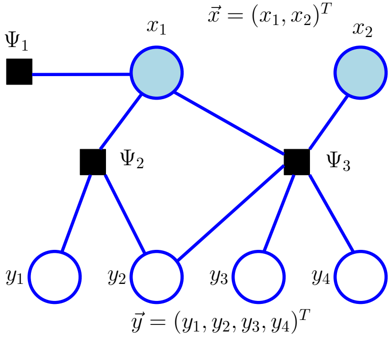
- Factor Ψ_i computes a scalar over all variables



Factor Graph

A Factor Graph is a bipartite graph over factors and variables

- Factor Ψ_i computes a scalar over all variables
- Let $\vec{x}$ be observed variables, $\vec{y}$ output variables

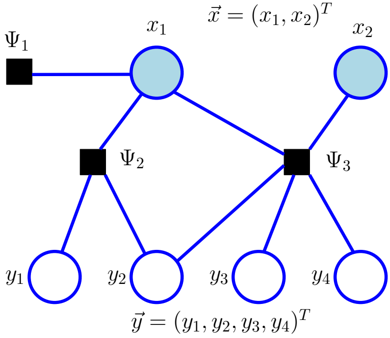


Factor Graph

A Factor Graph is a bipartite graph over factors and variables

- Factor Ψ_i computes a scalar over all variables
- Let $\vec{x}$ be observed variables, $\vec{y}$ output variables
- Common definition:
  $\Psi_i(\vec{x}_i, \vec{y}_i) =$
  $\exp\left(\sum_k \theta_{ki} f_{ki}(\vec{x}_i, \vec{y}_i)\right)$

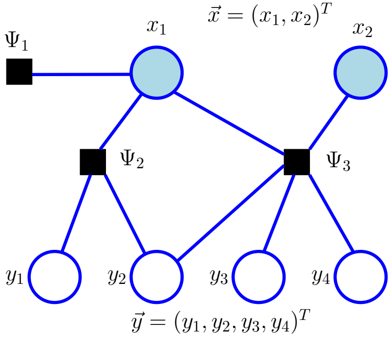
(parameters θ_{ki} and sufficient statistics $f_{ki}(\cdot)$)



Factor Graph

A Factor Graph is a bipartite graph over factors and variables

- Factor Ψ_i computes a scalar over all variables
- Let $\vec{x}$ be observed variables, $\vec{y}$ output variables
- Common definition:
  $\Psi_i(\vec{x}_i, \vec{y}_i) =$
  $\exp\left(\sum_k \theta_{ki} f_{ki}(\vec{x}_i, \vec{y}_i)\right)$
  (parameters θ_{ki} and sufficient statistics $f_{ki}(\cdot)$)
- Probability distribution:
  $p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_i \Psi_i(\vec{x}_i, \vec{y}_i)$



Templates for Factor Graphs

- Probability distribution

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_i \exp\left(\sum_k \theta_{ki} f_{ki}(\vec{x}_i, \vec{y}_i)\right)$$

Templates for Factor Graphs

- Probability distribution

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_i \exp\left(\sum_k \theta_{ki} f_{ki}(\vec{x}_i, \vec{y}_i)\right)$$

- Typically, a lot of parameter tying is applied:

Templates for Factor Graphs

- Probability distribution

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_i \exp\left(\sum_k \theta_{ki} f_{ki}(\vec{x}_i, \vec{y}_i)\right)$$

- Typically, a lot of parameter tying is applied:
- A Factor Template T_j consists of parameters θ_{jk} and statistic functions f_{jk} and some description of variables yielding tupels $(\vec{x}_j, \vec{y}_j)$

Templates for Factor Graphs

- Probability distribution

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_i \exp\left(\sum_k \theta_{ki} f_{ki}(\vec{x}_i, \vec{y}_i)\right)$$

- Typically, a lot of parameter tying is applied:
- A Factor Template T_j consists of parameters θ_{jk} and statistic functions f_{jk} and some description of variables yielding tupels $(\vec{x}_j, \vec{y}_j)$
- Parameters θ_{jk}, feature functions f_{jk} are shared across tupels

Templates for Factor Graphs

- Probability distribution

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_i \exp\left(\sum_k \theta_{ki} f_{ki}(\vec{x}_i, \vec{y}_i) \right)$$

- Typically, a lot of parameter tying is applied:
- A Factor Template T_j consists of parameters θ_{jk} and statistic functions f_{jk} and some description of variables yielding tupels $(\vec{x}_j, \vec{y}_j)$
- Parameters θ_{jk}, feature functions f_{jk} are shared across tupels
- The resulting probability distribution ($\mathcal{T}$ set of templates):

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_{T_j \in \mathcal{T}} \prod_{(\vec{x}_i, \vec{y}_i) \in T_i} \exp\left(\sum_k \theta_{kj} f_{kj}(\vec{x}_i, \vec{y}_i) \right)$$

FACTORIE

- Imperatively Defined Factor Graphs (IDF) allow to define Factor Graphs in an imperatively manner (as the name says. . .)
- FACTORIE is an implementation of IDF in Scala
- Markov Chain Monte Carlo inference
- Only one world is required to be represented
- Variables which do not change, are not evaluated

⇒ Huge graphs possible!

FACTORIE

IDF programming typically consists of 4 stages:

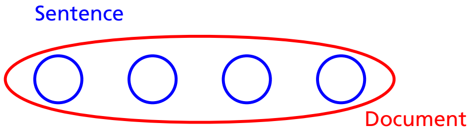
1. Design a data representation, assign variables
2. Design templates T_j which define the graphical structure
3. Implement application specific sampling (speed up inference)
4. Read data, learn parameters, test, evaluate

Overview

1. The BioNLP 2009 Shared Task on Event Extraction
   - Task Definition
   - Examples
   - Approaches and Motivation
   - Results
2. Imperatively Defined Factor Graphs (IDF)
   - Factor Graphs
   - Templates
   - FACTORIE
3. Document Wide Inference for the BioNLP Shared Task with IDF
   - Variables, Data Structure, Templates, Sampling, Objective
4. Results
5. Summary

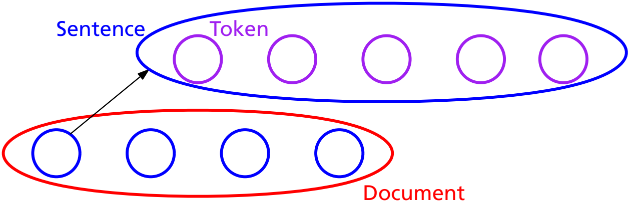
Variables and Data Structure

- Document is sequence of Sentences



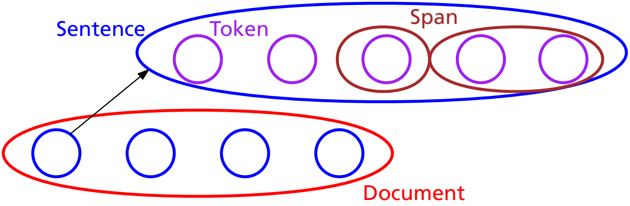
Variables and Data Structure

- Document is sequence of Sentences
- Sentence is sequence of Tokens
- Token represents token in Sentence



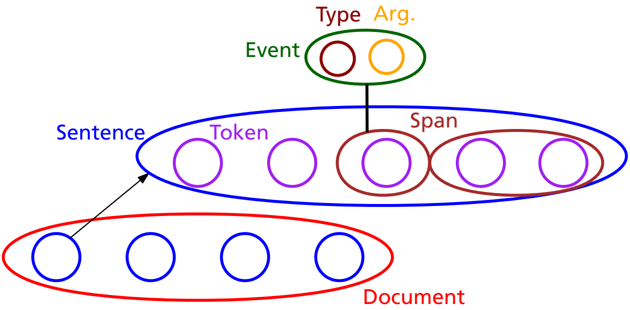
Variables and Data Structure

- Document is sequence of Sentences
- Sentence is sequence of Tokens
- Token represents token in Sentence
- Span defines subset of consecutive Tokens



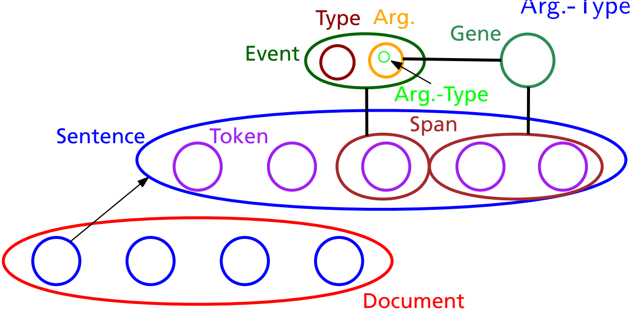
Variables and Data Structure

- Document is sequence of Sentences
- Sentence is sequence of Tokens
- Token represents token in Sentence
- Span defines subset of consecutive Tokens
- Event is on Span, has Arguments and Type



Variables and Data Structure

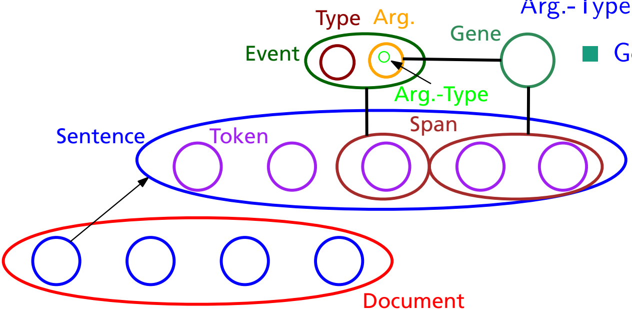
- Document is sequence of Sentences
- Sentence is sequence of Tokens
- Token represents token in Sentence
- Span defines subset of consecutive Tokens
- Event is on Span, has Arguments and Type
- Argument has Gene/Event and Arg.-Type



Fraunhofer SCAI UMASS

Variables and Data Structure

- Document is sequence of Sentences
- Sentence is sequence of Tokens
- Token represents token in Sentence
- Span defines subset of consecutive Tokens
- Event is on Span, has Arguments and Type
- Argument has Gene/Event and Arg.-Type
- Gene is on Span



Templates and Graphical Structure

- Templates define the graphical structure!

Templates and Graphical Structure

- Templates define the graphical structure!

Single Event Templates

- Measuring only on a single, isolated event (and its attributes)
- Features (conjunctions with each other in each category)

Fraunhofer SCAI

Templates and Graphical Structure

- Templates define the graphical structure!

Single Event Templates

- Measuring only on a single, isolated event (and its attributes)
- Features (conjunctions with each other in each category)

  Trigger: String, Stem, Dictionary, Pre-Hyphen, Event-Type, Normalized Event-Type

Templates and Graphical Structure

- Templates define the graphical structure!

Single Event Templates

- Measuring only on a single, isolated event (and its attributes)
- Features (conjunctions with each other in each category)

  Trigger String, Stem, Dictionary, Pre-Hyphen, Event-Type, Normalized Event-Type

  Trigger and Argument Trigger features with dependency path, argument, argument type

Templates and Graphical Structure

- Templates define the graphical structure!

Single Event Templates

- Measuring only on a single, isolated event (and its attributes)
- Features (conjunctions with each other in each category)

Trigger: String, Stem, Dictionary, Pre-Hyphen, Event-Type, Normalized Event-Type

Trigger and Argument: Trigger features with dependency path, argument, argument type

Argument Pair: Dependency Path, position to trigger, with trigger features

Templates and Graphical Structure

- Templates define the graphical structure!

Single Event Templates

- Measuring only on a single, isolated event (and its attributes)
- Features (conjunctions with each other in each category)

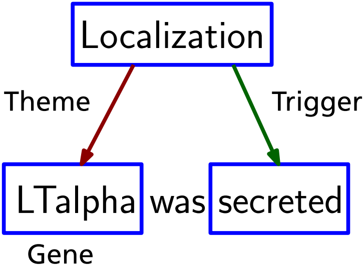
Trigger String, Stem, Dictionary, Pre-Hyphen, Event-Type, Normalized Event-Type

Trigger and Argument Trigger features with dependency path, argument, argument type

Argument Pair Dependency Path, position to trigger, with trigger features

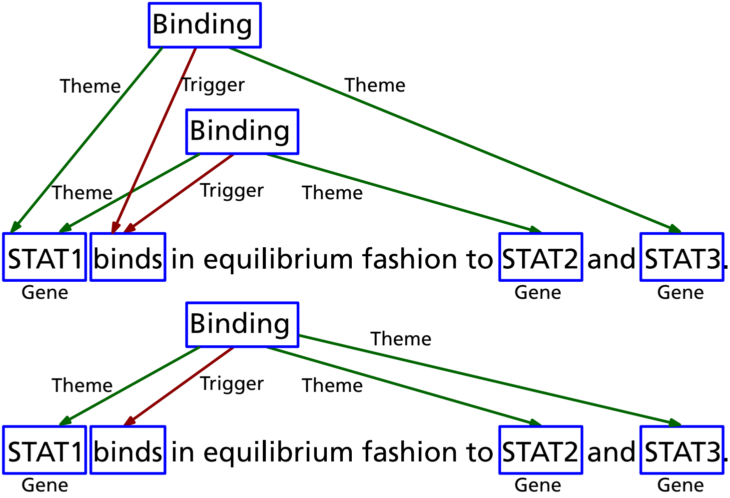
⇒ n-gram dependency sub-paths used additionally
(used parser: Charniak-Johnson reranking parser with McClosky-Charniak biomedical parsing model)

Templates and Graphical Structure – Examples

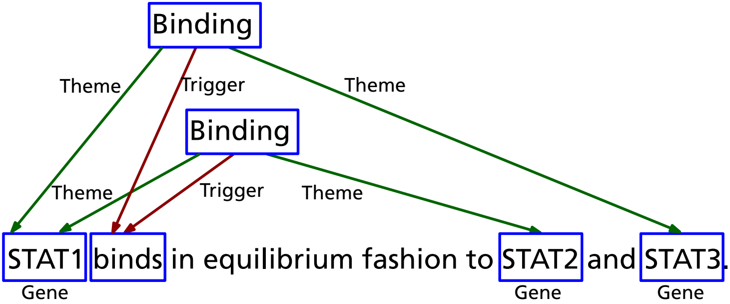


- Trigger
  - secret
  - secreted
  - secret+Localization
  - In-Dict
- Trigger and Argument
  - secret+Theme+Gene
  - secret+Theme+Gene+Localization
  - nsubjpass$\Uparrow$
  - nsubjpass$\Uparrow$+Localization

Templates and Graphical Structure – Examples



Templates and Graphical Structure – Examples



- Argument-Pair (example for STAT1—STAT2)
  - left-right-binding
  - nsubj⇓⇒prep_to⇑
- Argument-Pair (example for STAT2—STAT3)
  - right-right-binding
  - conj_and⇑

Fraunhofer SCAI UMASS

Templates and Graphical Structure

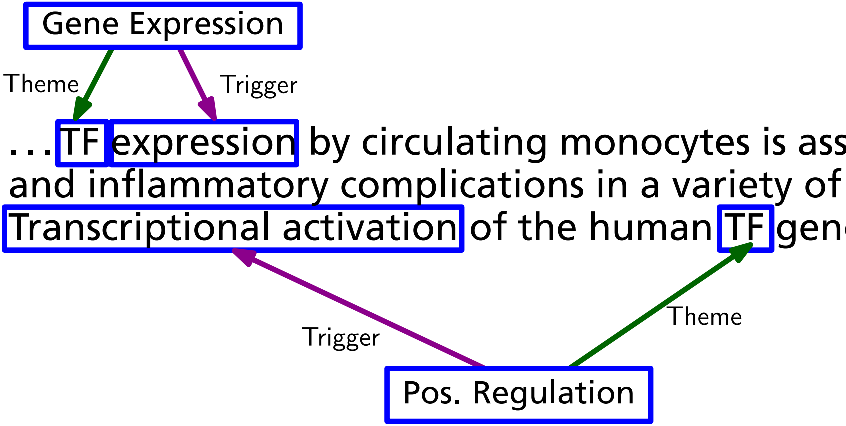
Event Pair Templates

- Measuring the relation between two events
- Features (again with conjunctions)

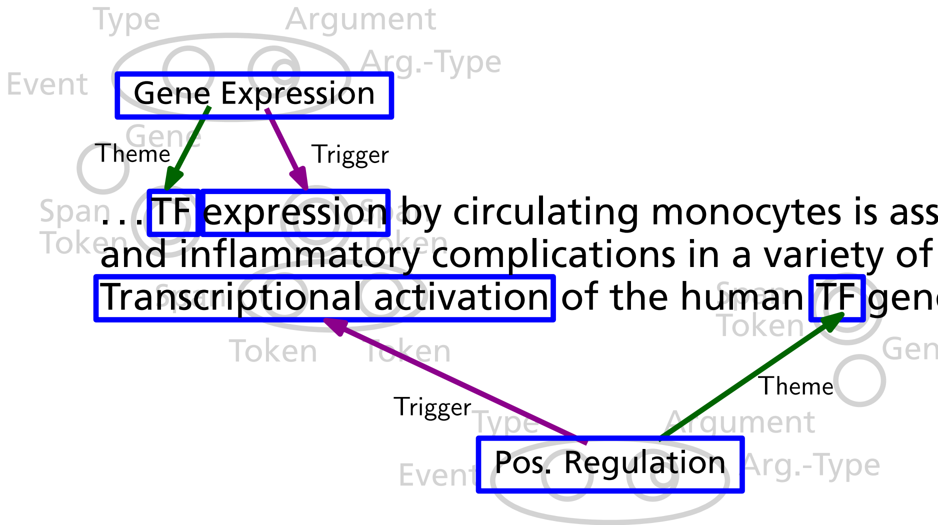
Parent-Child Unrolls for every event-A—event-B pair where A is argument of B
$\Rightarrow$ features as in trigger-argument feature

Document-Wide Unrolls for all events *in a document* which share the same gene (simple co-reference)
$\Rightarrow$ features are the transitions of the event types

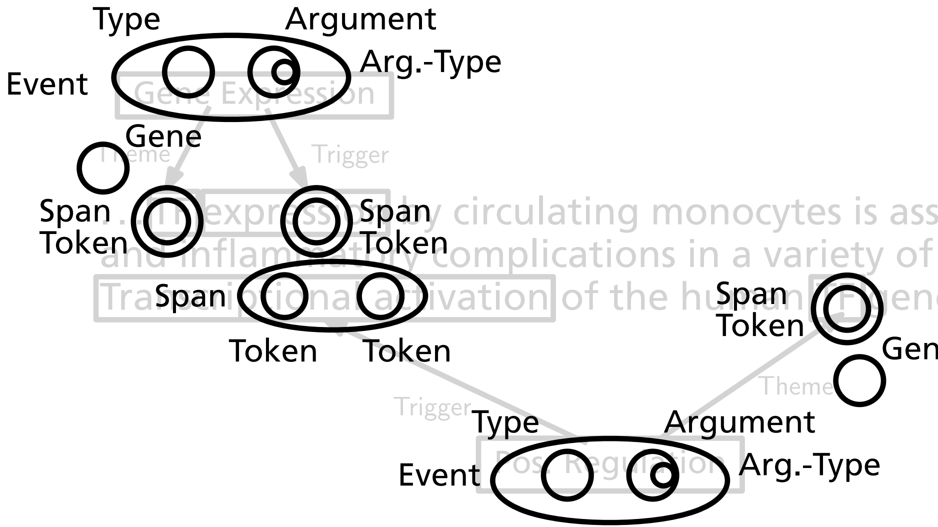
Templates and Graphical Structure – Example



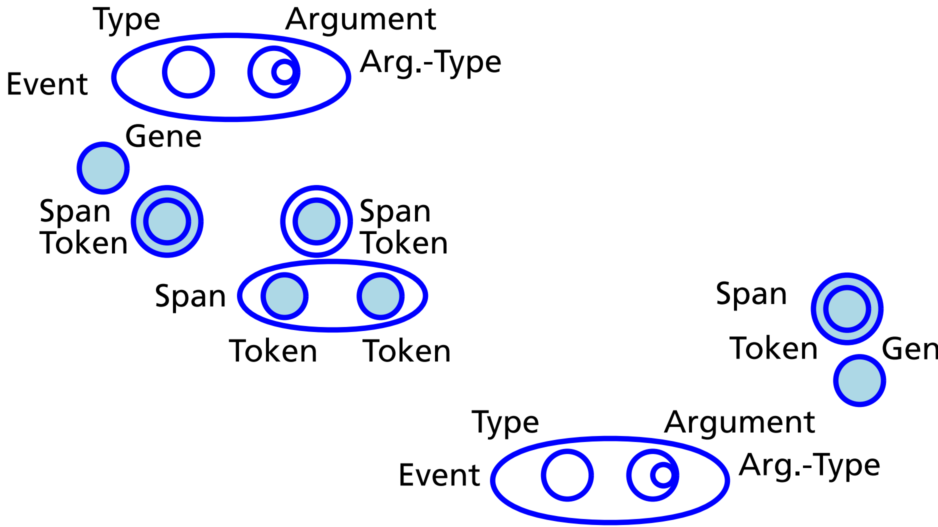
Templates and Graphical Structure – Example



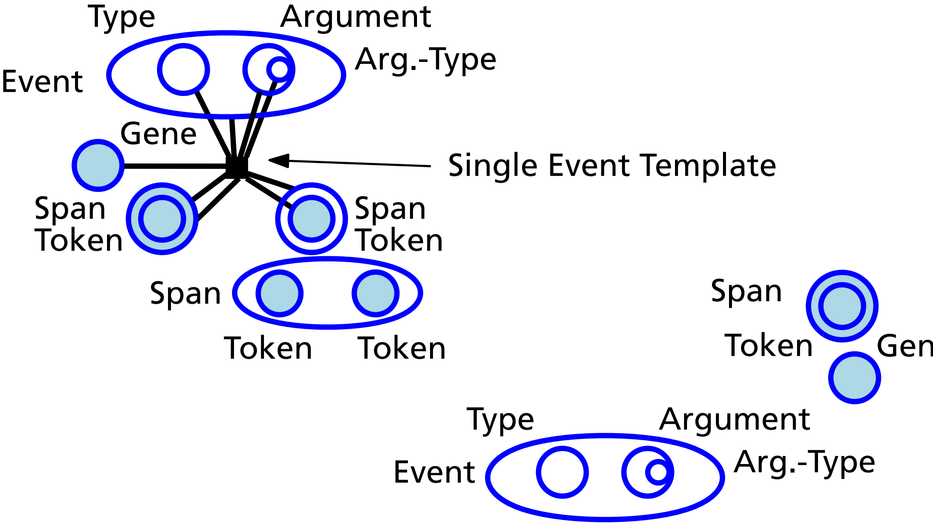
Templates and Graphical Structure – Example



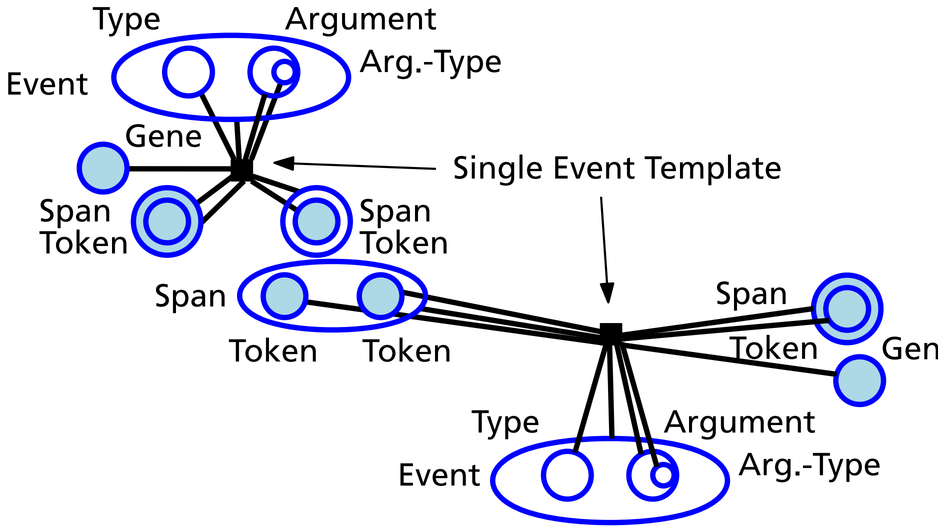
Templates and Graphical Structure – Example



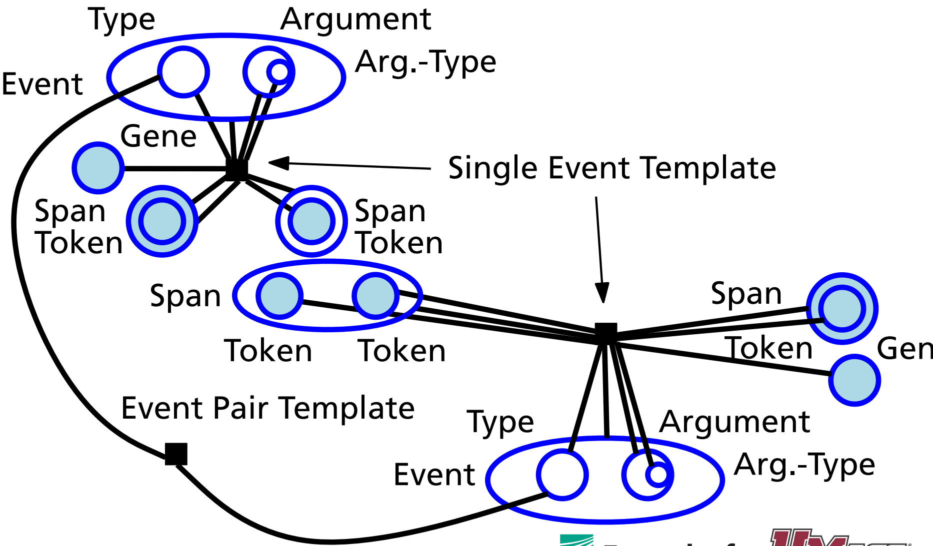
Templates and Graphical Structure – Example



Templates and Graphical Structure – Example



Templates and Graphical Structure – Example



Sampling and Inference

How to change the structures in a random walk like manner?

1st phase: Over all Spans (which are not of a Gene)

1. Add Events with all possible types
2. Keep best fitting (according to model score)
3. Add all possible argument structures
4. Keep best fitting or remove whole Event

Sampling and Inference

How to change the structures in a random walk like manner?

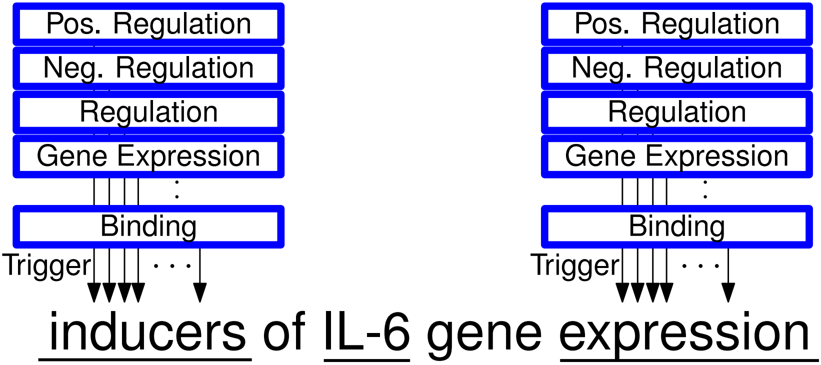
1st phase: Over all Spans (which are not of a Gene)

1. Add Events with all possible types
2. Keep best fitting (according to model score)
3. Add all possible argument structures
4. Keep best fitting or remove whole Event

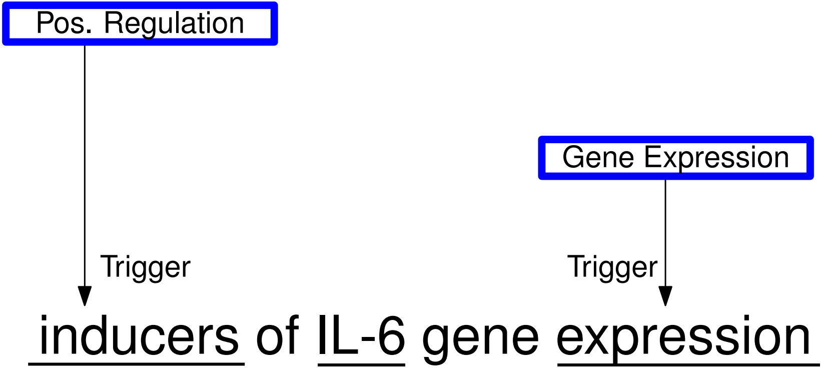
2nd phase: Over all Events

- Add argument (Cause or another Theme, dependent on event type)
- Remove argument
- Exchange argument with another event
- Remove whole event

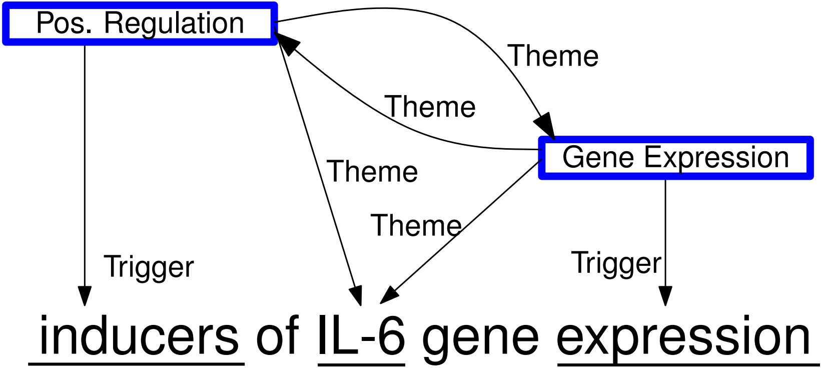
Sampling and Inference



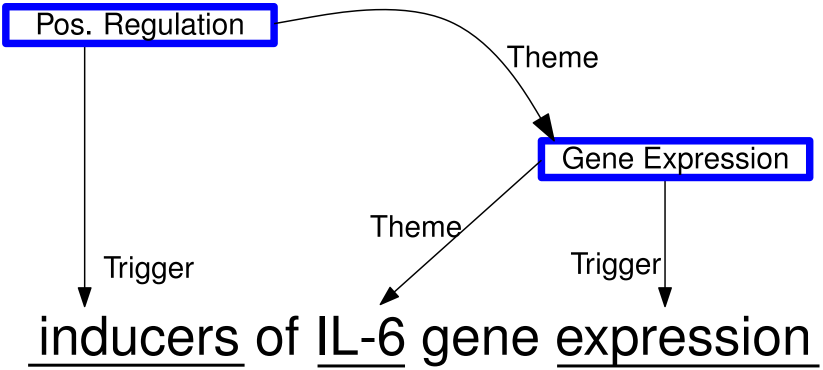
Sampling and Inference



Sampling and Inference



Sampling and Inference



Fraunhofer SCAI UMASS

Two Objective Functions

Objective on a single Event

$$f_1(e) = \mathrm{TP}(e) - \mathrm{FP}(e)$$
$$\mathrm{TP}(e) = \mathbf{1}_{\mathrm{TriggType}} + \mathrm{TP}_{\mathrm{ArgTrigg}} + \mathrm{TP}_{\mathrm{Arg}}$$

$\Rightarrow$ Cannot handle multiple, identical events

Objective on a sentence

$$f_2(s) = \mathrm{TP}_{\mathrm{TriggTypeTheme}} - \mathrm{FP}_{\mathrm{TriggTypeTheme}}$$

Overview

Fraunhofer SCAI UMASS

Results – Document Wide Template

	GE	L	Ph	PC	T	B	R	PR	NR
G. expression	**0.5**				0.1	0.1		0.2	0.1
Localization	0.2	**0.4**				0.1		0.1	0.1
Phosphorylation			**0.6**	0.1		0.3			
Prot. Catabolism	0.1		0.3	**0.4**		0.2			
Transcription	0.2	0.1			0.1		0.1	**0.4**	
Binding	0.1					**0.6**	0.1	0.1	
Regulation	0.1					0.2	0.1	**0.4**	0.1
Pos. Regulation	**0.1**					**0.1**	0.1	0.1	
Neg. Regulation	0.1						0.1	0.3	**0.4**

Development Set

Configuration	Precision	Recall	F_1	F_1 Reg.	
No Arg-Pair	68.4	45.3	54.5	43.6	*
No parent event	68.9	45.8	55.0	43.8	
No doc-wide	68.3	45.3	54.4	43.6	*
Best	68.5	46.7	55.6	45.6	
Miwa 2010	–	–	55.6		
Riedel 2011	67.9	51.8	58.7		

Fraunhofer SCAI UMASS

Test Set

Event Class	Prec.	Rec.	F_1
Gene Expression	78.7	62.7	69.8
Transcription	71.0	16.1	26.2
Protein Catabolism	85.7	42.9	57.1
Phosphorylation	79.3	79.3	79.3
Localization	93.3	40.2	56.2
Binding	56.7	34.0	42.5
Regulation	45.0	23.0	30.6
Positive Regulation	56.9	31.8	40.8
Negative Regulation	51.5	31.1	38.8
Total	65.0	40.0	49.6

Overview

Summary

- BioNLP Shared Task is especially difficult because of nested structures (goes beyond often addressed Protein-Protein-Interaction)

Summary

- BioNLP Shared Task is especially difficult because of nested structures (goes beyond often addressed Protein-Protein-Interaction)
- FACTORIE is a powerful and intuitive framework for probabilistic programming and allows to address the BioNLP shared task in a joint manner

Summary

- BioNLP Shared Task is especially difficult because of nested structures (goes beyond often addressed Protein-Protein-Interaction)
- FACTORIE is a powerful and intuitive framework for probabilistic programming and allows to address the BioNLP shared task in a joint manner
- Inter-Event, especially document-wide features have a positive impact!

Thank YOU for your attention!