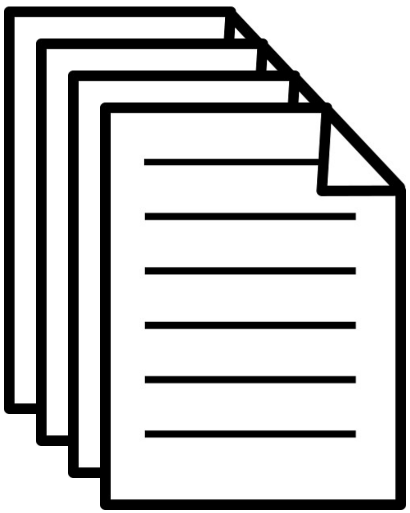
Search-based Structured Prediction for biological event extraction

李辰

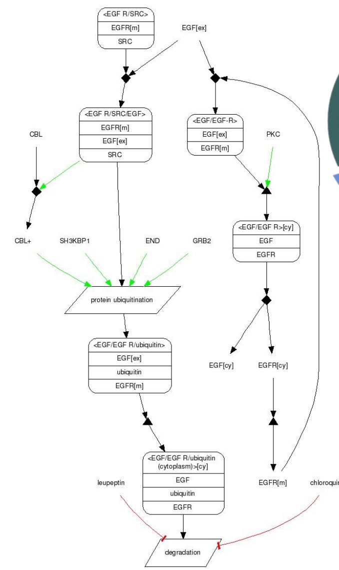
Background

Unstructured knowledge



MEDLINE

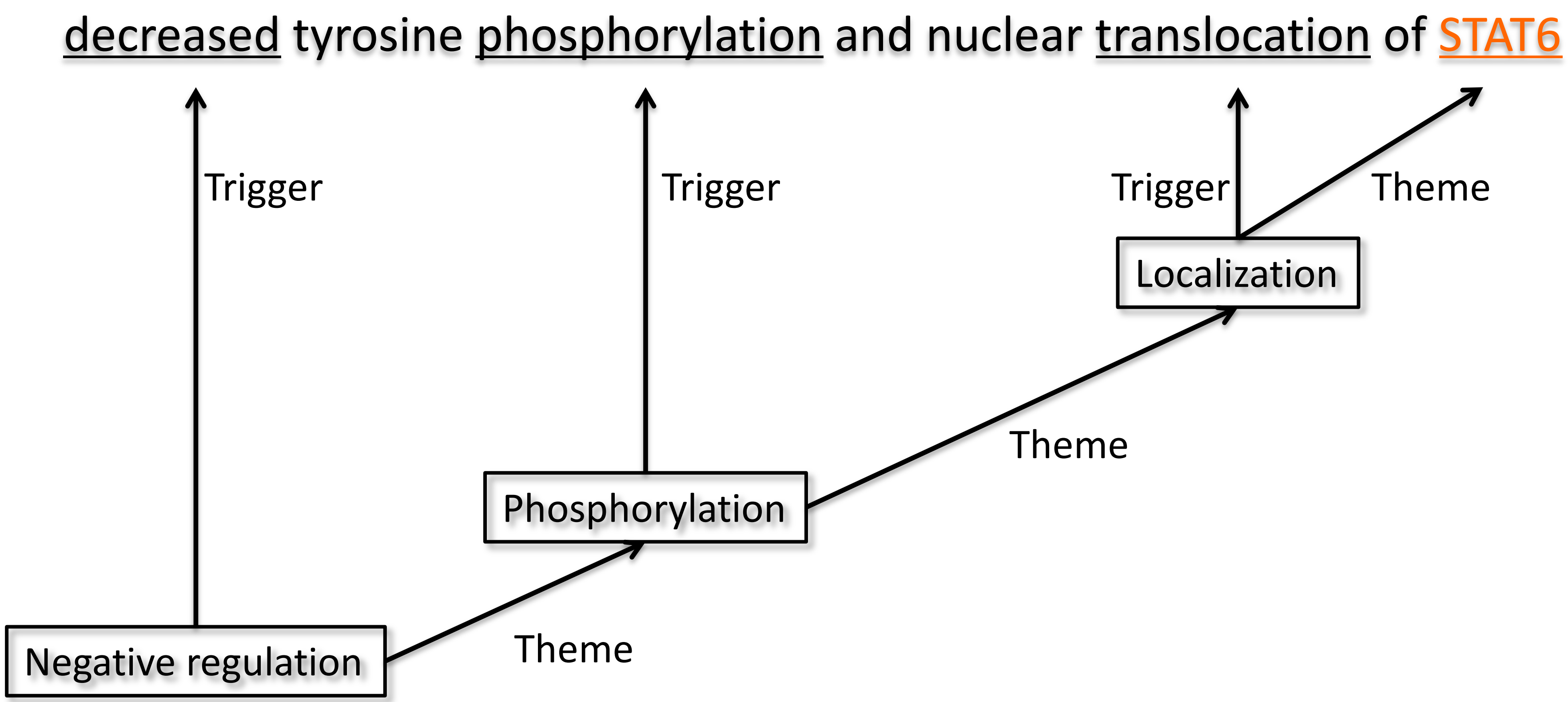
Structured knowledge



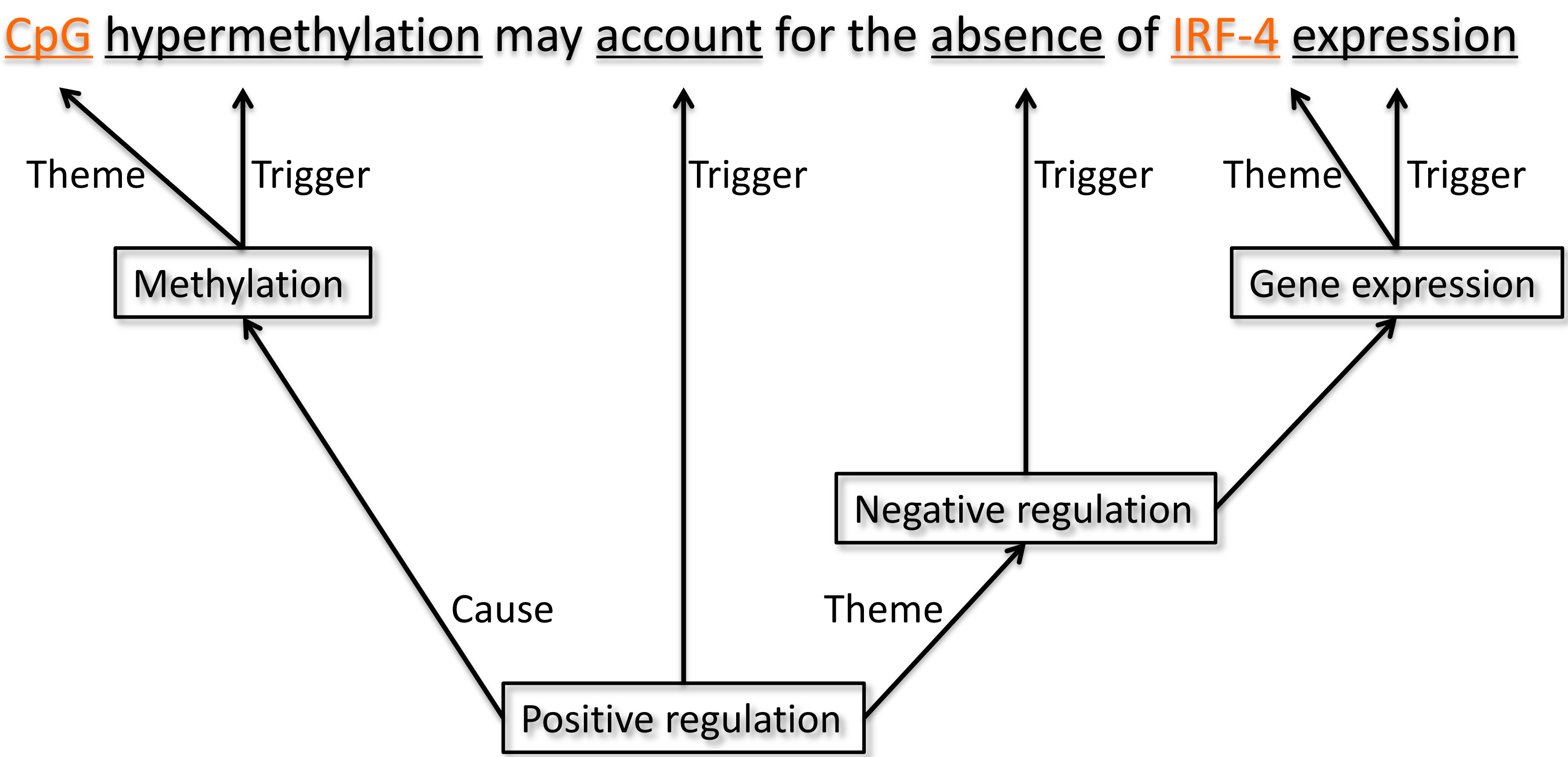
BioModels Database
Li C. et al., BMC systems biology

- Quantitatively characterise morphology of different types of biological networks in the scientific literature.
- Semantically enrich curated biological networks
- Discover hidden relations

Biomedical event extraction

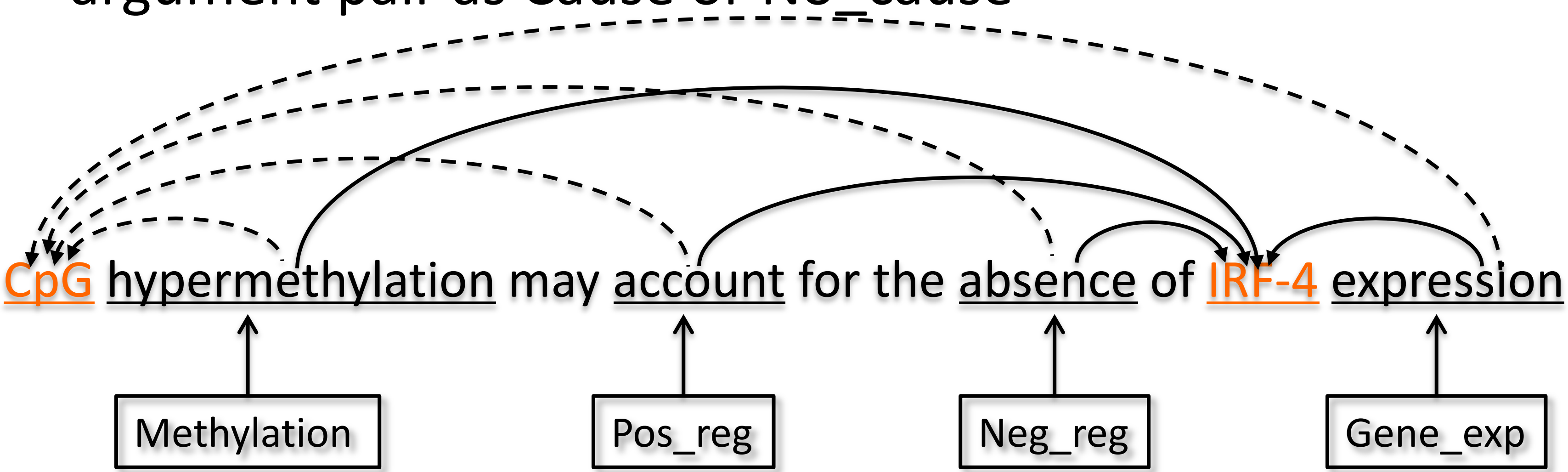


Biomedical event extraction



Modeling event extraction
- Event extraction decomposition

- Trigger recognition: classify each token as one of the event types or No_trigger
- Theme assignment: classify each candidate trigger-argument pair as Theme or No_theme
- Cause assignment: classify each candidate trigger-argument pair as Cause or No_cause



Modeling event extraction
- Event types and features

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction
- Event types and features

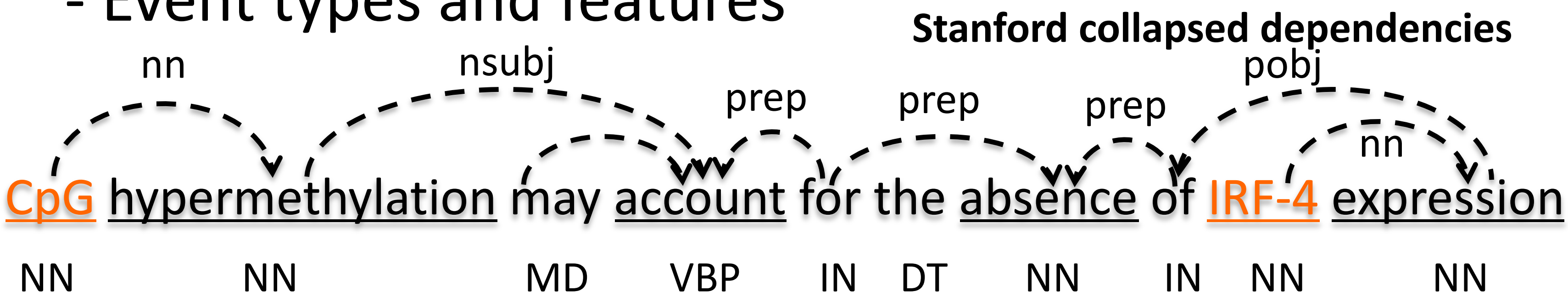
CpG hypermethylation may account for the absence of IRF-4 expression

Methylation Pos_reg Neg_reg Gene_exp

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction

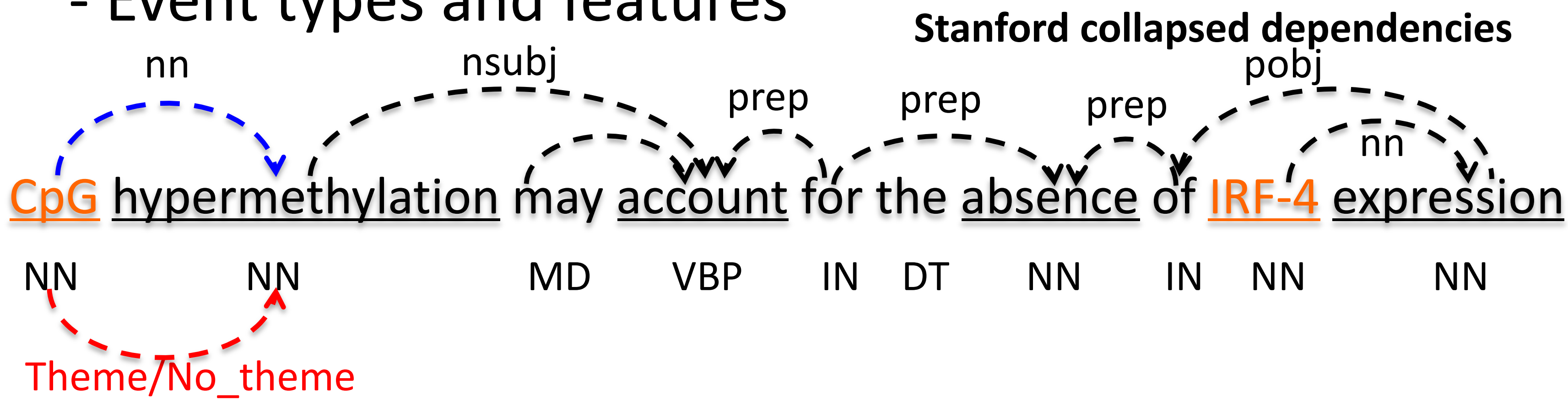
- Event types and features



1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction

- Event types and features



Path: {trigger}$\xrightarrow{\text{nsubj}}${argument}

Lemma_Path: hypermethylation$\xrightarrow{\text{nsubj}}${argument}

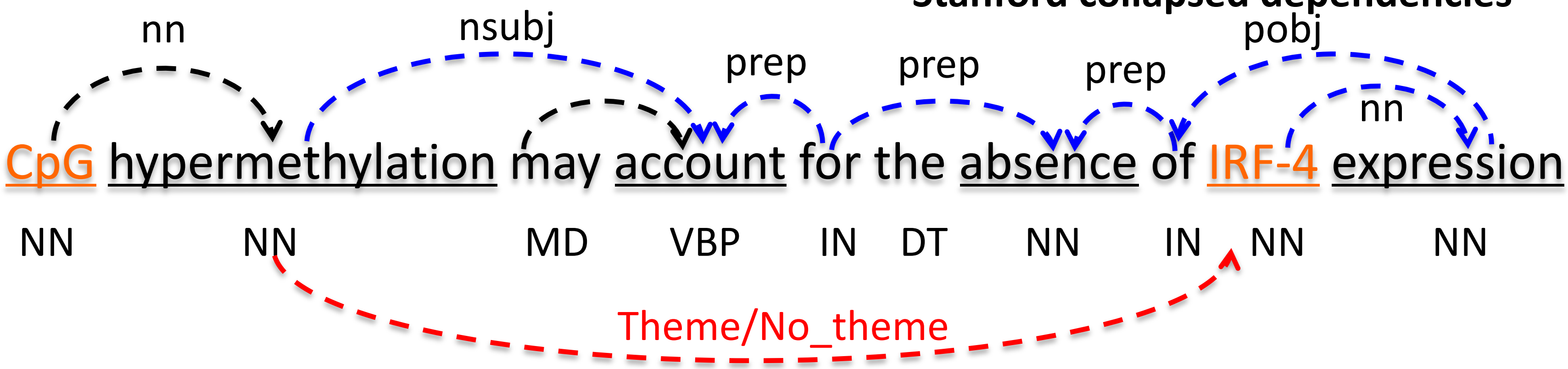
POS_Path: NN$\xrightarrow{\text{nsubj}}${argument}

etc.

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction
- Event types and features

Stanford collapsed dependencies



Path: {trigger} $\xrightarrow{\text{nsubj}}$ {token} $\xrightarrow{\text{prep}}$ {token} $\xrightarrow{\text{prep}}$ {token} $\xrightarrow{\text{prep}}$ {token} $\xrightarrow{\text{pobj}}$ {token} $\xrightarrow{\text{nn}}$ {argument}

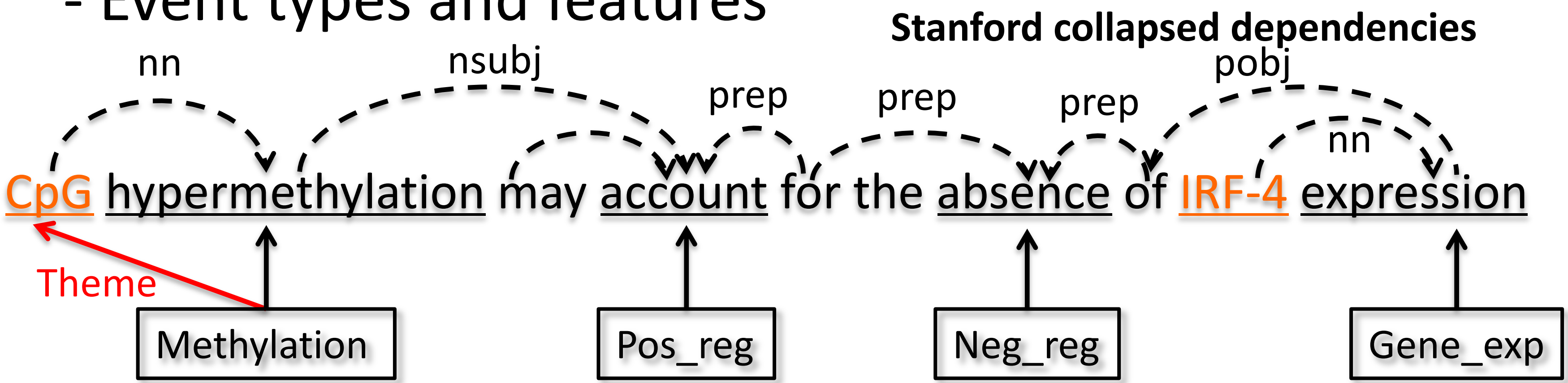
Lemma_Path:
hypermethylation $\xrightarrow{\text{nsubj}}$ {token} $\xrightarrow{\text{prep}}$ {token} $\xrightarrow{\text{prep}}$ {token} $\xrightarrow{\text{prep}}$ {token} $\xrightarrow{\text{pobj}}$ {token} $\xrightarrow{\text{nn}}$ {argument}

POS_Path:
hypermethylation $\xrightarrow{\text{nsubj}}$ {token} $\xrightarrow{\text{prep}}$ {token} $\xrightarrow{\text{prep}}$ {token} $\xrightarrow{\text{prep}}$ {token} $\xrightarrow{\text{pobj}}$ {token} $\xrightarrow{\text{nn}}$ {argument}
etc.

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

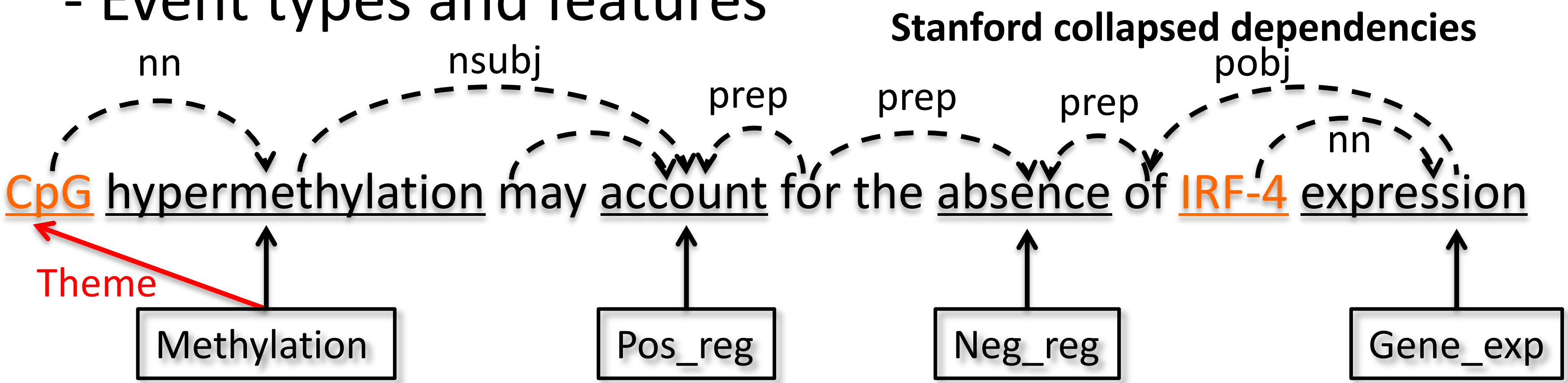
Modeling event extraction

- Event types and features



1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction
- Event types and features

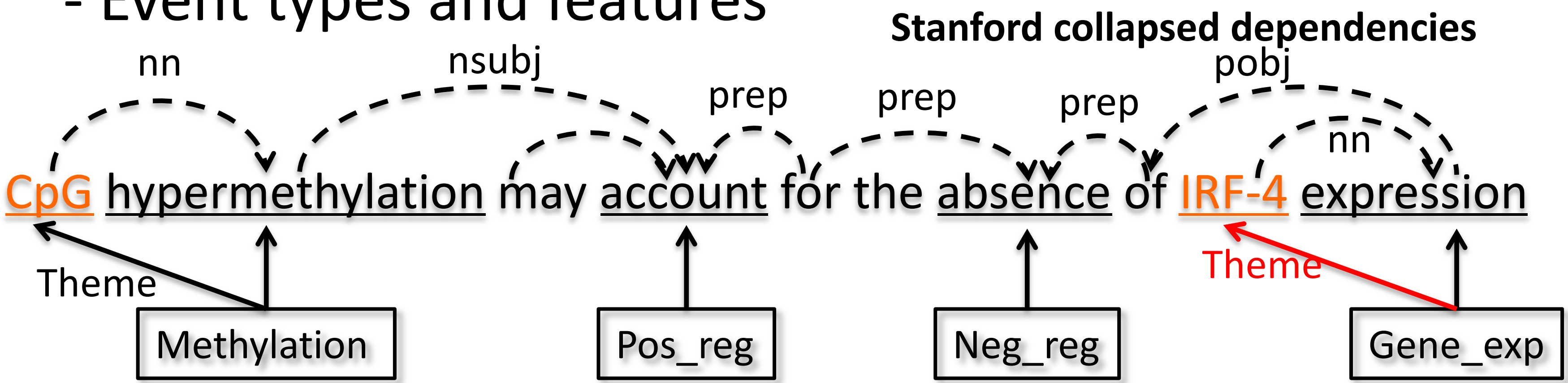


E1	Methylation	hypermethylation	CpG	

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction

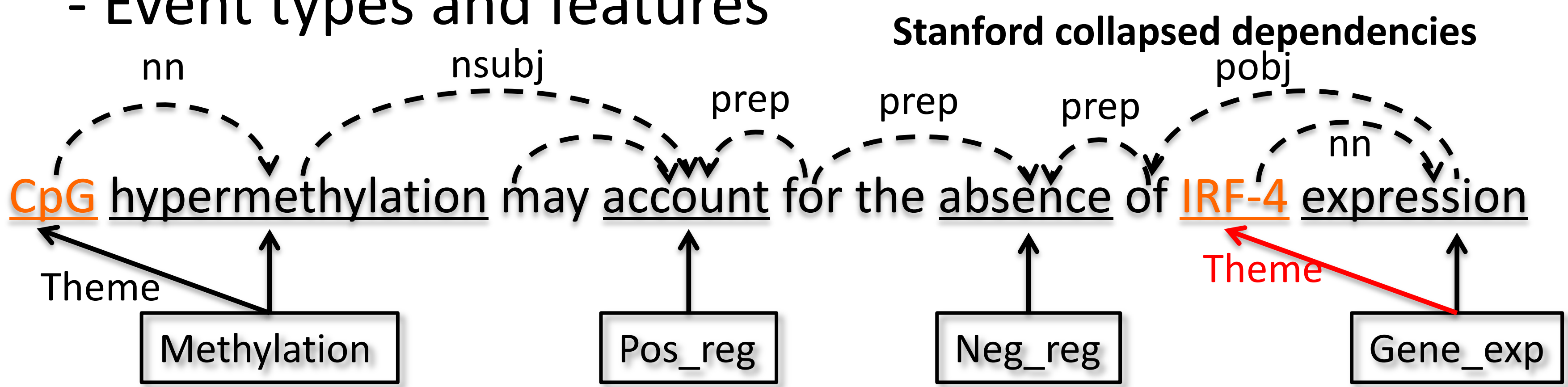
- Event types and features



E1	Methylation	hypermethylation	CpG	

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction
- Event types and features

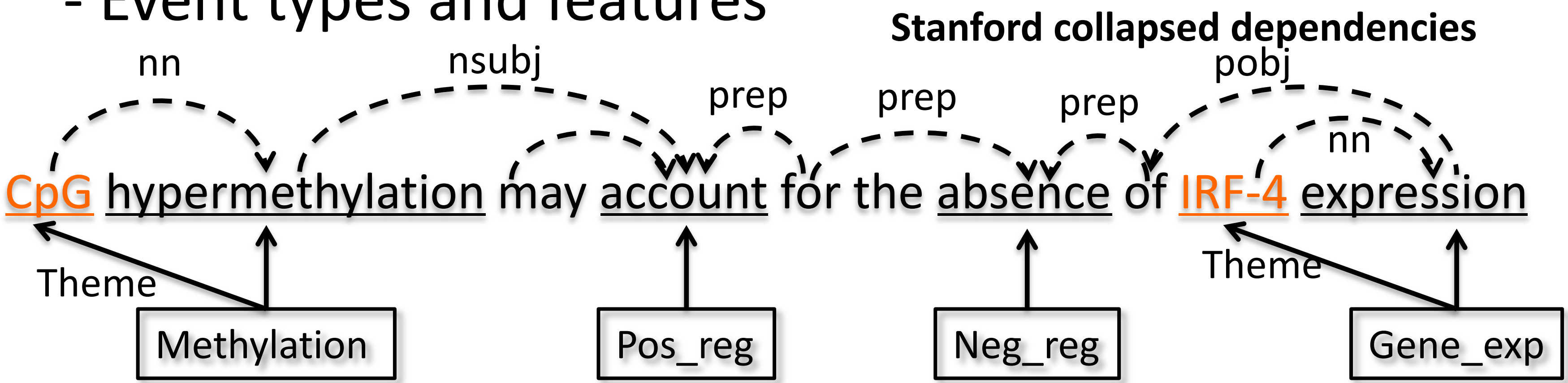


E1	Methylation	hypermethylation	CpG	
E2	Gene expression	expression	IRF-4	

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction

- Event types and features



E1	Methylation	hypermethylation	CpG	
E2	Gene expression	expression	IRF-4	

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

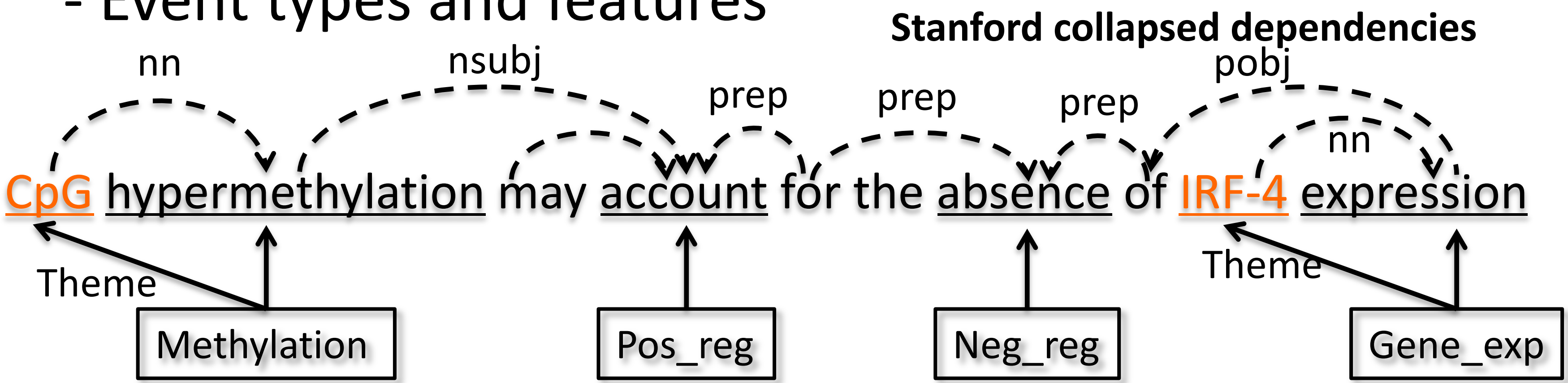
Modeling event extraction
- Event types and features

binding of A, B and C

1st case	Binding	A
2nd case	Binding	B
3rd case	Binding	C
4th case	Binding	AB
5th case	Binding	BC
6th case	Binding	AC
7th case	Binding	ABC

Modeling event extraction

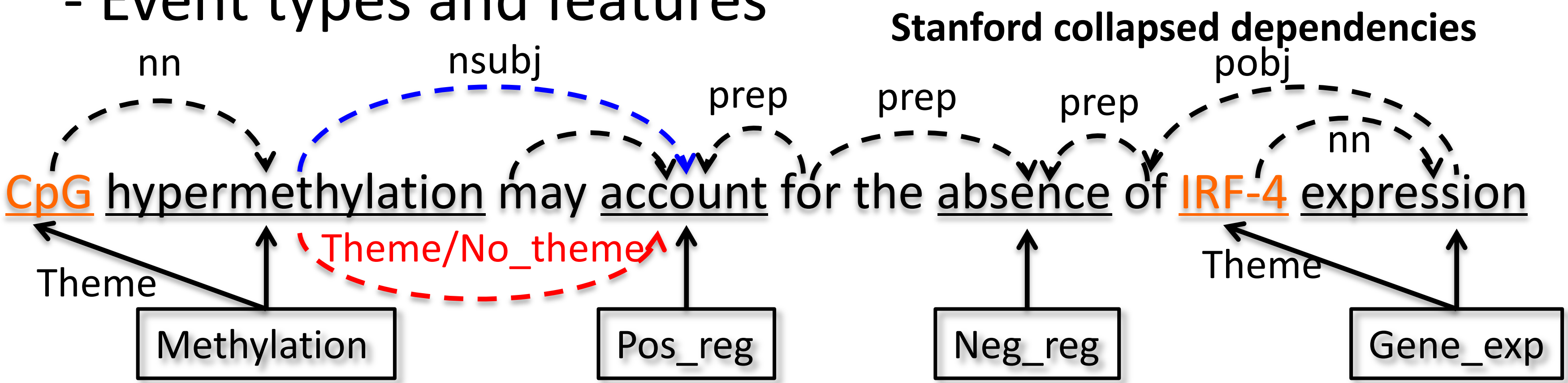
- Event types and features



E1	Methylation	hypermethylation	CpG	
E2	Gene expression	expression	IRF-4	

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

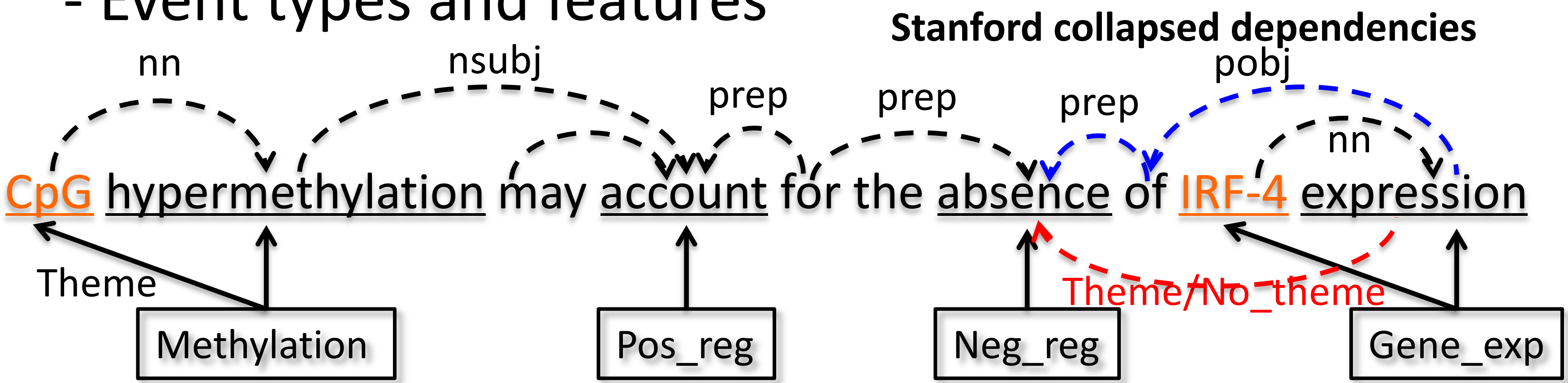
Modeling event extraction
- Event types and features



E1	Methylation	hypermethylation	CpG	
E2	Gene expression	expression	IRF-4	

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

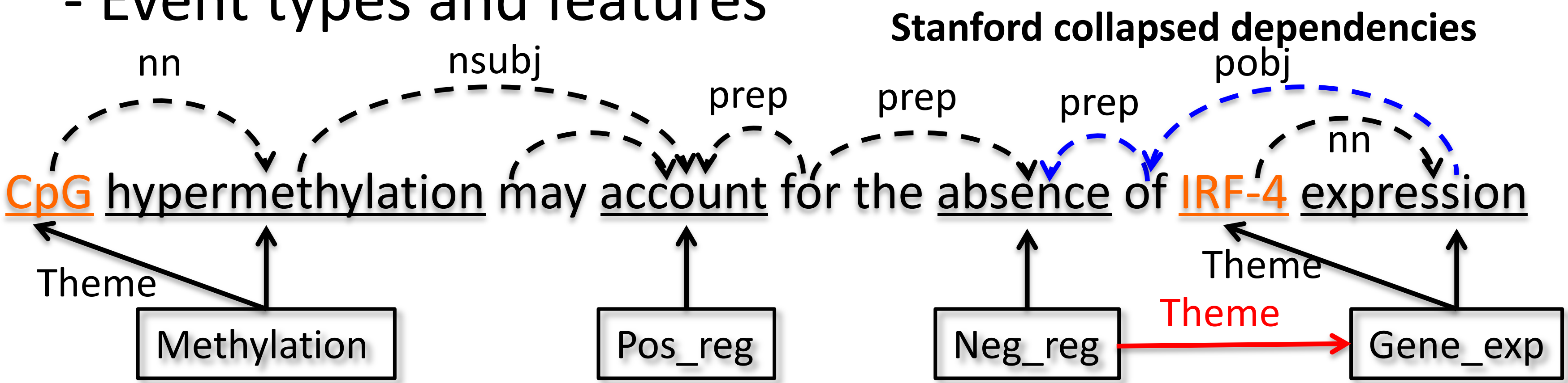
Modeling event extraction
- Event types and features



E1	Methylation	hypermethylation	CpG	
E2	Gene expression	expression	IRF-4	

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction
- Event types and features

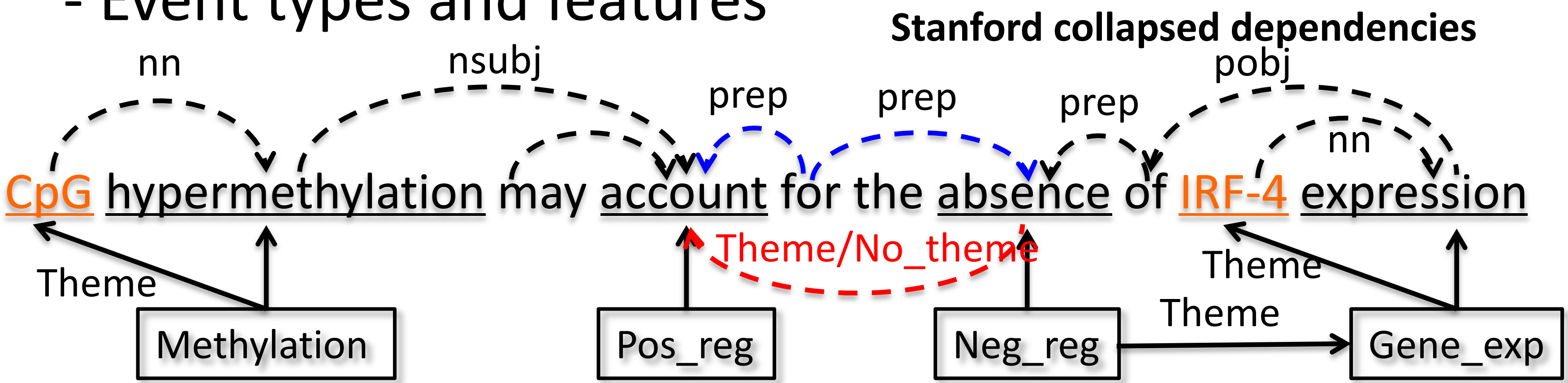


E1	Methylation	hypermethylation	CpG	
E2	Gene expression	expression	IRF-4	
E3	Negative regulation	absence	E2	

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction

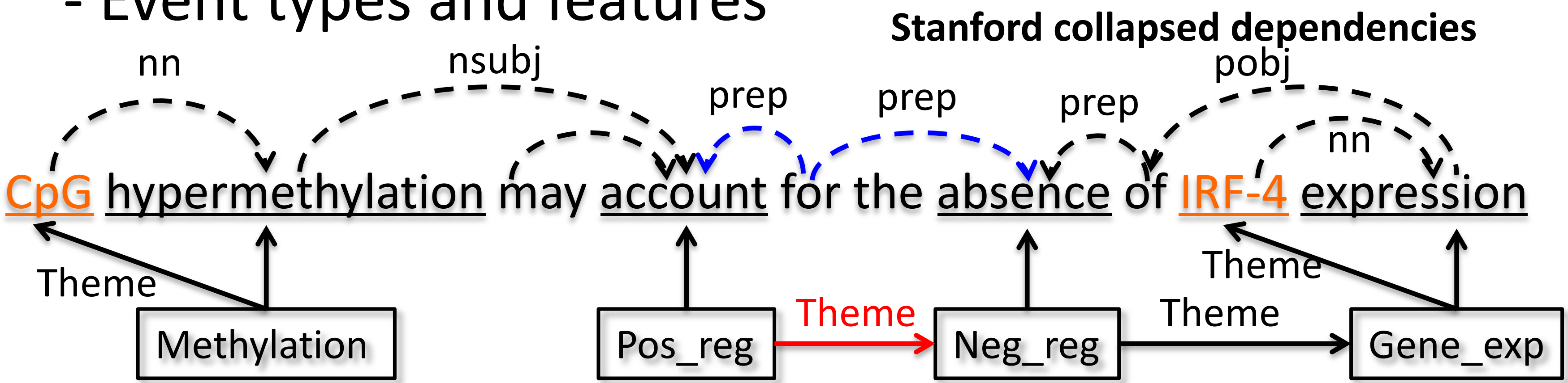
- Event types and features



E1	Methylation	hypermethylation	CpG	
E2	Gene expression	expression	IRF-4	
E3	Negative regulation	absence	E2	

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction
- Event types and features

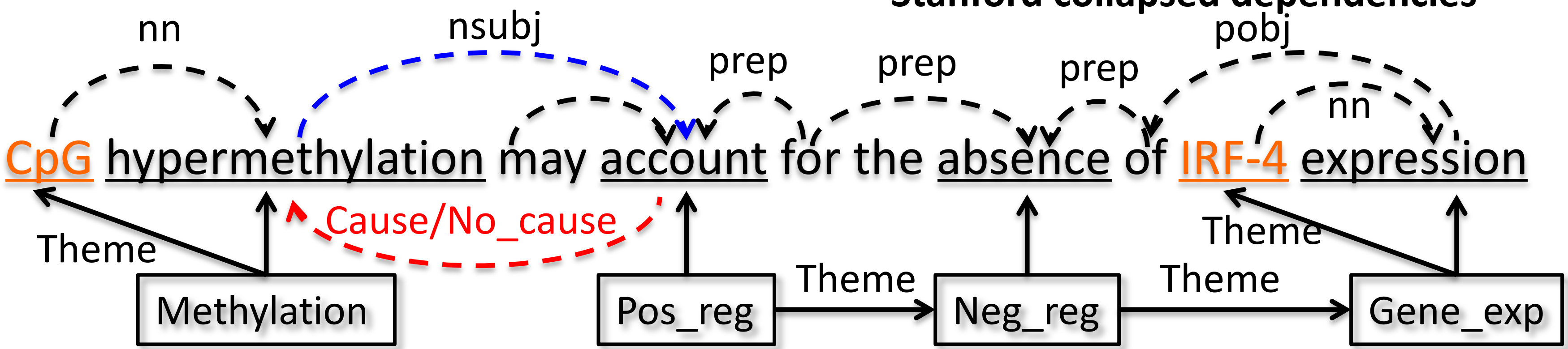


E1	Methylation	hypermethylation	CpG	
E2	Gene expression	expression	IRF-4	
E3	Negative regulation	absence	E2	
E4	Positive regulation	account	E3	

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction
- Event types and features

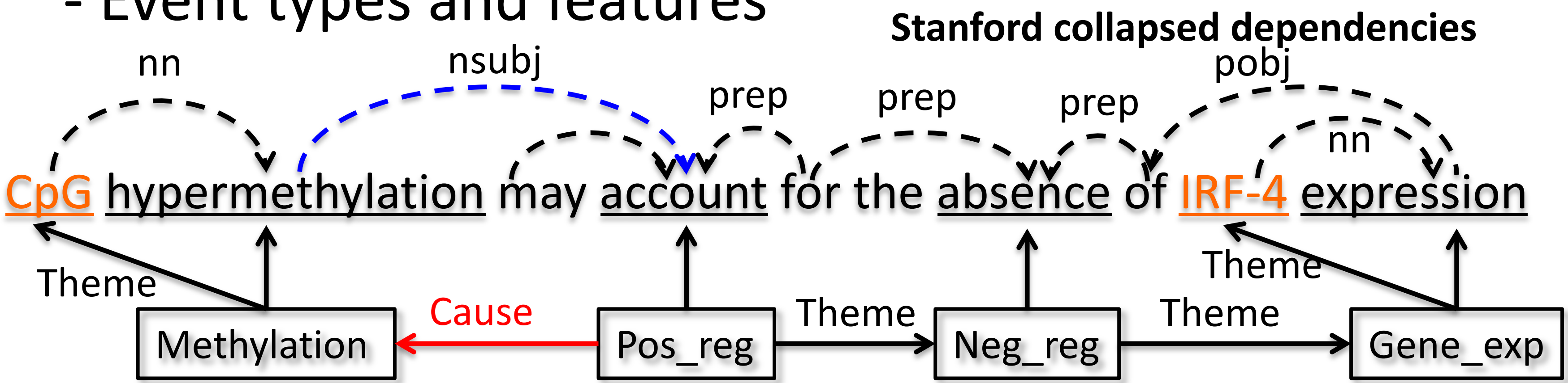
Stanford collapsed dependencies



E1	Methylation	hypermethylation	CpG	
E2	Gene expression	expression	IRF-4	
E3	Negative regulation	absence	E2	
E4	Positive regulation	account	E3	

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

Modeling event extraction
- Event types and features



E1	Methylation	hypermethylation	CpG	
E2	Gene expression	expression	IRF-4	
E3	Negative regulation	absence	E2	
E4	Positive regulation	account	E3	E1

1. Simple events: an event takes one theme, e.g. gene expression
2. Binding: an event has multi-theme
3. Complex events: an event has both theme and cause
4. Recursive events: an event has theme and/or cause, which can be another event

BioNLP shared task

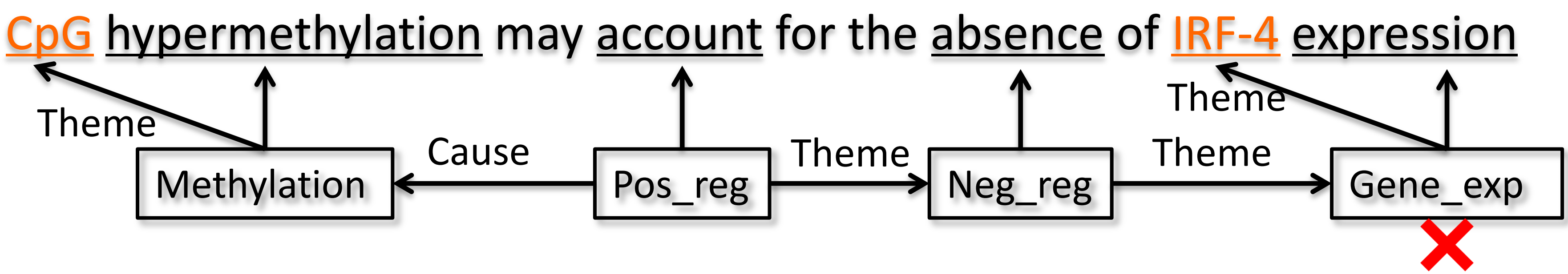
- Organized in 2009, 2011 and 2013 (next in 2015)
- Emphasis on
  - Expressive structured models of extracted information
  - “High-level” information extraction: directionality and polarity of reactions

BioNLP shared task (*cont.*)

- Event structure is defined closer to reactions in biological networks.
  - Each event is associated with an event type
  - Event is explicitly mentioned in the text
  - Reactant number may vary in each event
  - Each reactant has different role, e.g. cause, theme, site etc.
  - Event can play a role in another event

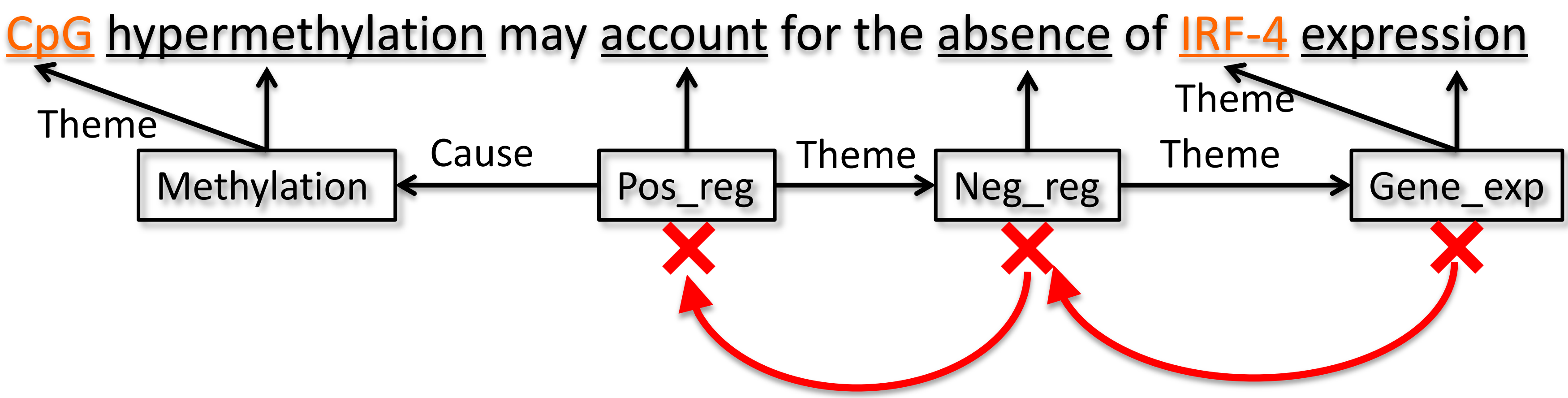
BioNLP shared task (*cont.*)

- Structural evaluation



BioNLP shared task (*cont.*)

- Structural evaluation



3 false positives and 3 false negatives

Structured prediction requirements

- Labeled data
- Loss function
- Optimal policy
- A cost-sensitive classification

Structured prediction (*algorithm*)

```
Initialise
  Structured instances S,
  optimal policy π,
  cost sensitive learning algorithm CSCL
  loss function ℓ

Train
  current policy h = π
  while h depends significantly on π do
    Examples E = ∅
    for s in S do
      Predict h(s) = ŷ_1 ... ŷ_T
      for ŷ_t in h(s) do
        Extract features Φ_t = f(s, ŷ_{1:t-1})
        for each possible action y_t^i do
          Predict y'_{t+1:T} = h(s | ŷ_{1:t-1}, y_t^i)
          Estimate c_t^i = ℓ(ŷ_{1:t-1}, y_t^i, y'_{t+1,T})
        Add (Φ_t, c_t) to E
    Learn a classifier h_new = CSCL(E)
    h = β h_new + (1 + β) h
Return policy h
```

Structured prediction (*algorithm*)

Initialise
  Structured instances S,
  optimal policy π,
  cost sensitive learning algorithm $CSCL$
  loss function ℓ

Train
  current policy $h = \pi$
  while h depends significantly on π do
    Examples $E = \emptyset$
    for s in S do
      Predict $h(s) = \hat{y}_1 \ldots \hat{y}_T$
      for $\hat{y}_t$ in $h(s)$ do
        Extract features $\Phi_t = f(s, \hat{y}_{1:t-1})$
        for each possible action y_t^i do
          Predict $y'_{t+1:T} = h(s|\hat{y}_{1:t-1}, y_t^i)$
          Estimate $c_t^i = \ell(\hat{y}_{1:t-1}, y_t^i, y'_{t+1,T})$
        Add (Φ_t, c_t) to E
    Learn a classifier $h_{new} = CSCL(E)$
    $h = \beta h_{new} + (1+\beta)h$
Return policy h

Structured prediction (*algorithm*)

Initialise
  Structured instances S,
  optimal policy π,
  cost sensitive learning algorithm $CSCL$
  loss function ℓ

Train
  current policy $h = \pi$
  while h depends significantly on π do
    Examples $E = \emptyset$
    for s in S do
      Predict $h(s) = \hat{y}_1 \ldots \hat{y}_T$
      for $\hat{y}_t$ in $h(s)$ do
        Extract features $\Phi_t = f(s, \hat{y}_{1:t-1})$
        for each possible action y_t^i do
          Predict $y'_{t+1:T} = h(s|\hat{y}_{1:t-1}, y_t^i)$
          Estimate $c_t^i = \ell(\hat{y}_{1:t-1}, y_t^i, y'_{t+1,T})$
        Add (Φ_t, c_t) to E
    Learn a classifier $h_{new} = CSCL(E)$
    $h = \beta h_{new} + (1 + \beta) h$
Return policy h

Structured prediction (*algorithm*)

Initialise
  Structured instances S,
  optimal policy π,
  cost sensitive learning algorithm $CSCL$
  loss function ℓ

Train
  current policy $h = \pi$
  while h depends significantly on π do
    Examples $E = \emptyset$
    for s in S do
      Predict $h(s) = \hat{y}_1 \ldots \hat{y}_T$
      for $\hat{y}_t$ in $h(s)$ do
        Extract features $\Phi_t = f(s, \hat{y}_{1:t-1})$
        for each possible action y_t^i do
          Predict $y'_{t+1:T} = h(s|\hat{y}_{1:t-1}, y_t^i)$
          Estimate $c_t^i = \ell(\hat{y}_{1:t-1}, y_t^i, y'_{t+1,T})$
        Add (Φ_t, c_t) to E
    Learn a classifier $h_{new} = CSCL(E)$
    $h = \beta h_{new} + (1 + \beta) h$
Return policy h

Structured prediction (*algorithm*)

Initialise
  Structured instances S,
  optimal policy π,
  cost sensitive learning algorithm $CSCL$
  loss function ℓ

Train
  current policy $h = \pi$
  while h depends significantly on π do
    Examples $E = \emptyset$
    for s in S do
      Predict $h(s) = \hat{y}_1 \ldots \hat{y}_T$
      for $\hat{y}_t$ in $h(s)$ do
        Extract features $\Phi_t = f(s, \hat{y}_{1:t-1})$
        for each possible action y_t^i do
          Predict $y'_{t+1:T} = h(s|\hat{y}_{1:t-1}, y_t^i)$
          Estimate $c_t^i = \ell(\hat{y}_{1:t-1}, y_t^i, y'_{t+1,T})$
        Add (Φ_t, c_t) to E
    Learn a classifier $h_{new} = CSCL(E)$
    $h = \beta h_{new} + (1 + \beta) h$
Return policy h

Structured prediction (*algorithm*)

Initialise
  Structured instances S,
  optimal policy π,
  cost sensitive learning algorithm $CSCL$
  loss function ℓ

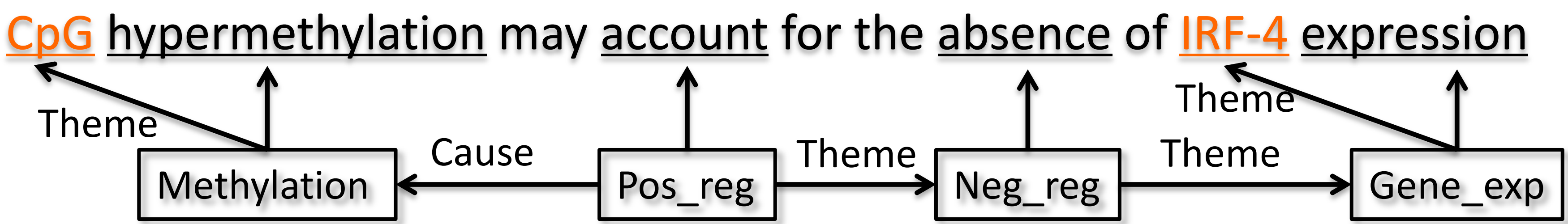
Train
  current policy $h = \pi$
  while h depends significantly on π do
    Examples $E = \emptyset$
    for s in S do
      Predict $h(s) = \hat{y}_1 \ldots \hat{y}_T$
      for $\hat{y}_t$ in $h(s)$ do
        Extract features $\Phi_t = f(s, \hat{y}_{1:t-1})$
        for each possible action y_t^i do
          Predict $y'_{t+1:T} = h(s|\hat{y}_{1:t-1}, y_t^i)$
          Estimate $c_t^i = \ell(\hat{y}_{1:t-1}, y_t^i, y'_{t+1,T})$
        Add (Φ_t, c_t) to E
    Learn a classifier $h_{new} = CSCL(E)$
    $h = \beta h_{new} + (1 + \beta) h$
Return policy h

Structured prediction (*algorithm*)

Initialise
  Structured instances S,
  optimal policy π,
  cost sensitive learning algorithm $CSCL$
  loss function ℓ

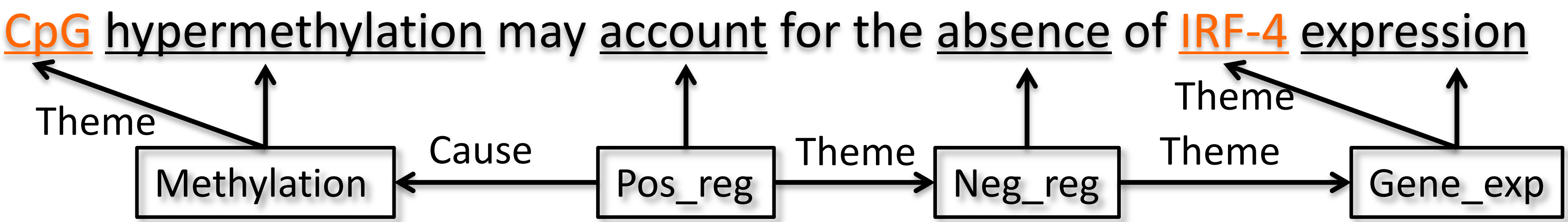
Train
  current policy $h = \pi$
  while h depends significantly on π do
    Examples $E = \emptyset$
    for s in S do
      Predict $h(s) = \hat{y}_1 \ldots \hat{y}_T$
      for $\hat{y}_t$ in $h(s)$ do
        Extract features $\Phi_t = f(s, \hat{y}_{1:t-1})$
        for each possible action y_t^i do
          Predict $y'_{t+1:T} = h(s | \hat{y}_{1:t-1}, y_t^i)$
          Estimate $c_t^i = \ell(\hat{y}_{1:t-1}, y_t^i, y'_{t+1,T})$
        Add (Φ_t, c_t) to E
    Learn a classifier $h_{new} = CSCL(E)$
    $h = \beta h_{new} + (1 + \beta) h$
Return policy h

Example of training



Example of training

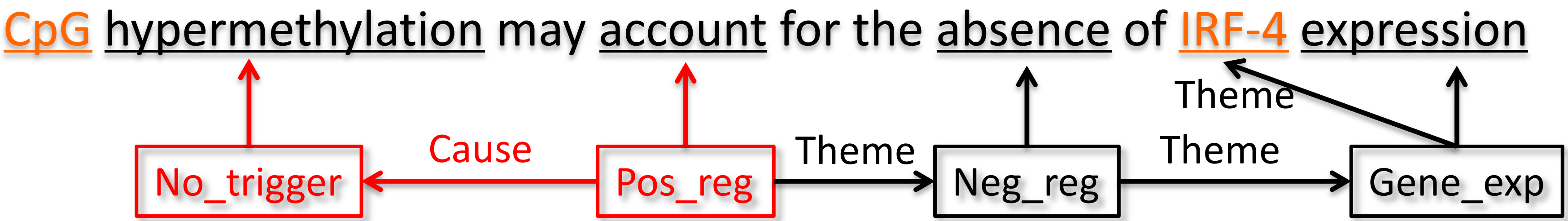
- 1st iteration -- Using the optimal policy



	Non_trigger	Pos_reg	Neg_reg	Methylation	Gene_exp
hypermethylation					
may					
account					
for					
absence					
expression					

Example of training

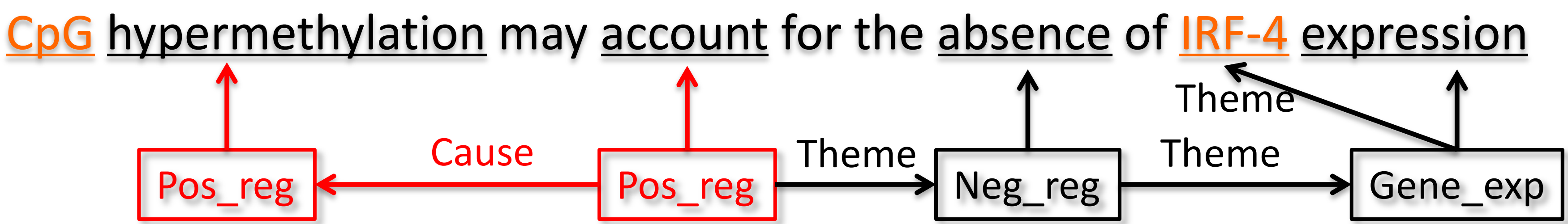
- 1^{st} iteration -- Using the optimal policy



	Non_trigger	Pos_reg	Neg_reg	Methylation	Gene_exp
hypermethylation	3				
may					
account					
for					
absence					
expression					

Example of training

- 1st iteration -- Using the optimal policy

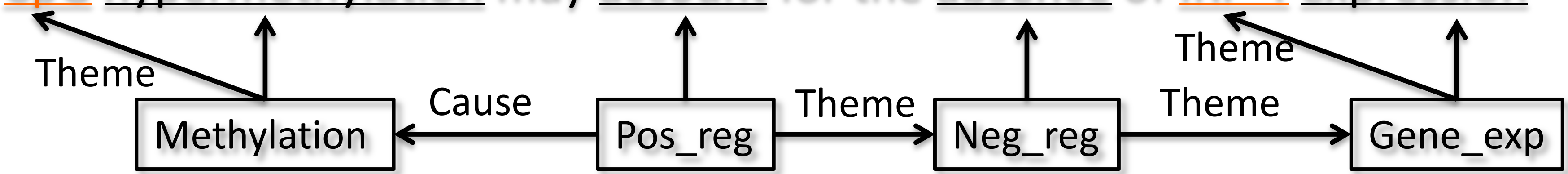


	Non_trigger	Pos_reg	Neg_reg	Methylation	Gene_exp
hypermethylation	3	4			
may					
account					
for					
absence					
expression					

Example of training

- 1st iteration -- Using the optimal policy

CpG hypermethylation may account for the absence of IRF-4 expression

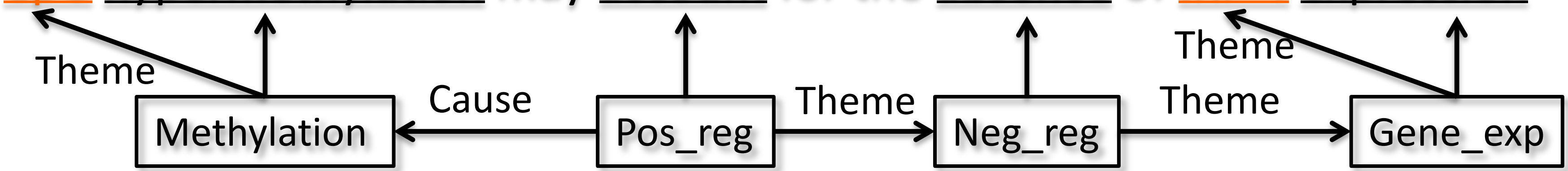


	Non_trigger	Pos_reg	Neg_reg	Methylation	Gene_exp
hypermethylation	3	4	4	0	4
may					
account					
for					
absence					
expression					

Example of training

- 1st iteration -- Using the optimal policy

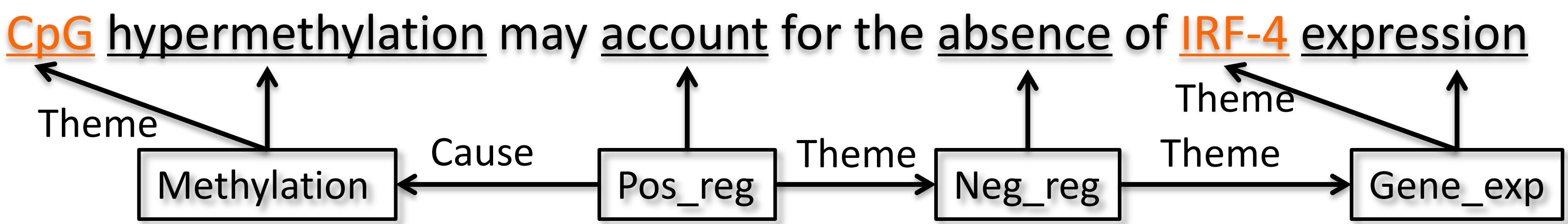
CpG hypermethylation may account for the absence of IRF-4 expression



	Non_trigger	Pos_reg	Neg_reg	Methylation	Gene_exp
hypermethylation	3	4	4	0	4
may	0	0	0	0	0
account					
for					
absence					
expression					

Example of training

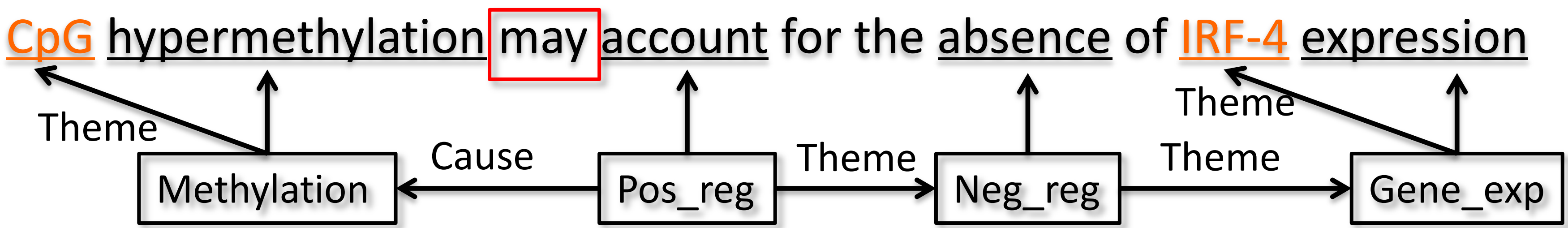
- 1st iteration -- Using the optimal policy



	Non_trigger	Pos_reg	Neg_reg	Methylation	Gene_exp
hypermethylation	3	4	4	0	4
may	0	0	0	0	0
account	1	0	2	2	2
for	0	0	0	0	0
absence	3	4	0	4	4
expression	5	6	6	6	0

Example of training

- 2nd iteration -- Learned hypothesis

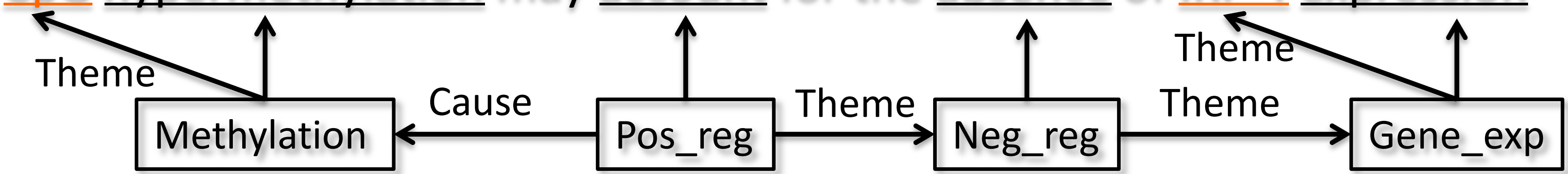


	Non_trigger	Pos_reg	Neg_reg	Methylation	Gene_exp
hypermethylation	3	4	4	0	4
may	0	1	0	0	0
account					
for					
absence					
expression					

Example of training

- 2nd iteration -- Learned hypothesis

CpG hypermethylation may account for the absence of IRF-4 expression



	Non_trigger	Pos_reg	Neg_reg	Methylation	Gene_exp
hypermethylation	3	4	4	0	4
may	0	1	0	0	0
account	1	0	2	2	2
for	0	0	0	0	0
absence	3	4	0	4	4
expression	5	6	6	6	0

Averaged perceptron

Input

  training examples $X = x_1 \dots x_T$,

  cost vectors $c_1 \dots c_T$,

  initialise weights $w_0^{(k)} = (0, \dots, 0)$,

  for $x_t \in X$ **do**

    predict label $\hat{y}_t = argmax_k(w_t^{(k)} \times x_t)$

    receive cost vector c_t

    if $c_t^{(\hat{y}_t)} > 0$ **then**

      loss $\ell_t = w_t^{(\hat{y}_t)} \times x_t - w_t^{(y_t)} \times x_t + \sqrt{c_t^{(\hat{y}_t)}}$

      learning rate $\tau_t = \frac{\ell_t}{\|x_t\|^2 + \frac{1}{2c}}$

      weight $w_{t+1}^{(y_t)} = w_t + \tau_t x_t$

      $w_{t+1}^{(\hat{y}_t)} = w_t - \tau_t x_t$

  average $w_{avg} = \frac{\sum_{i=0}^{T \times R} w_i}{T \times R}$

Averaged perceptron

Input
  training examples $X = x_1 \ldots x_T$,
  cost vectors $c_1 \ldots c_T$,
  initialise weights $w_0^{(k)} = (0, \ldots, 0)$,
  for $x_t \in X$ do
    predict label $\hat{y}_t = argmax_k(w_t^{(k)} \times x_t)$
    receive cost vector c_t
    if $c_t^{(\hat{y}_t)} > 0$ then
      loss $\ell_t = w_t^{(\hat{y}_t)} \times x_t - w_t^{(y_t)} \times x_t + \sqrt{c_t^{(\hat{y}_t)}}$
      learning rate $\tau_t = \frac{\ell_t}{\|x_t\|^2 + \frac{1}{2c}}$
      weight $w_{t+1}^{(y_t)} = w_t + \tau_t x_t$
        $w_{t+1}^{(\hat{y}_t)} = w_t - \tau_t x_t$
  average $w_{avg} = \frac{\sum_{i=0}^{T \times R} w_i}{T \times R}$

Conclusion

- Biological networks are described in complicated structures in the scientific literature.
- Structured prediction for a complex structured prediction task
- Structured prediction achieved state-of-the-art performance
- A flexible system for extracting more types of biological networks

Questions?