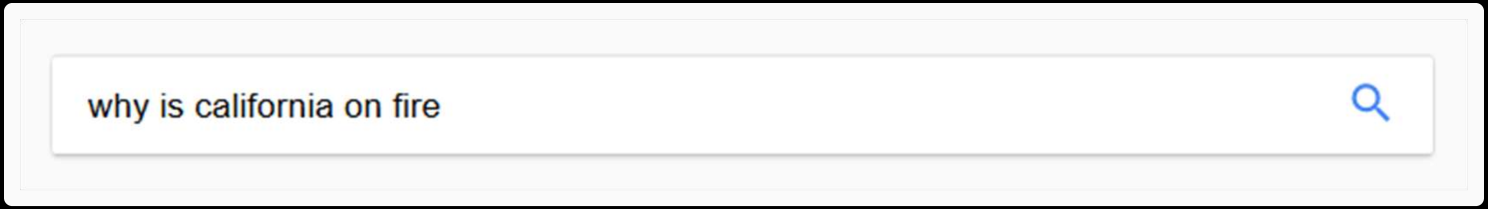
Annotating and Automatically Tagging Constructions of Causal Language

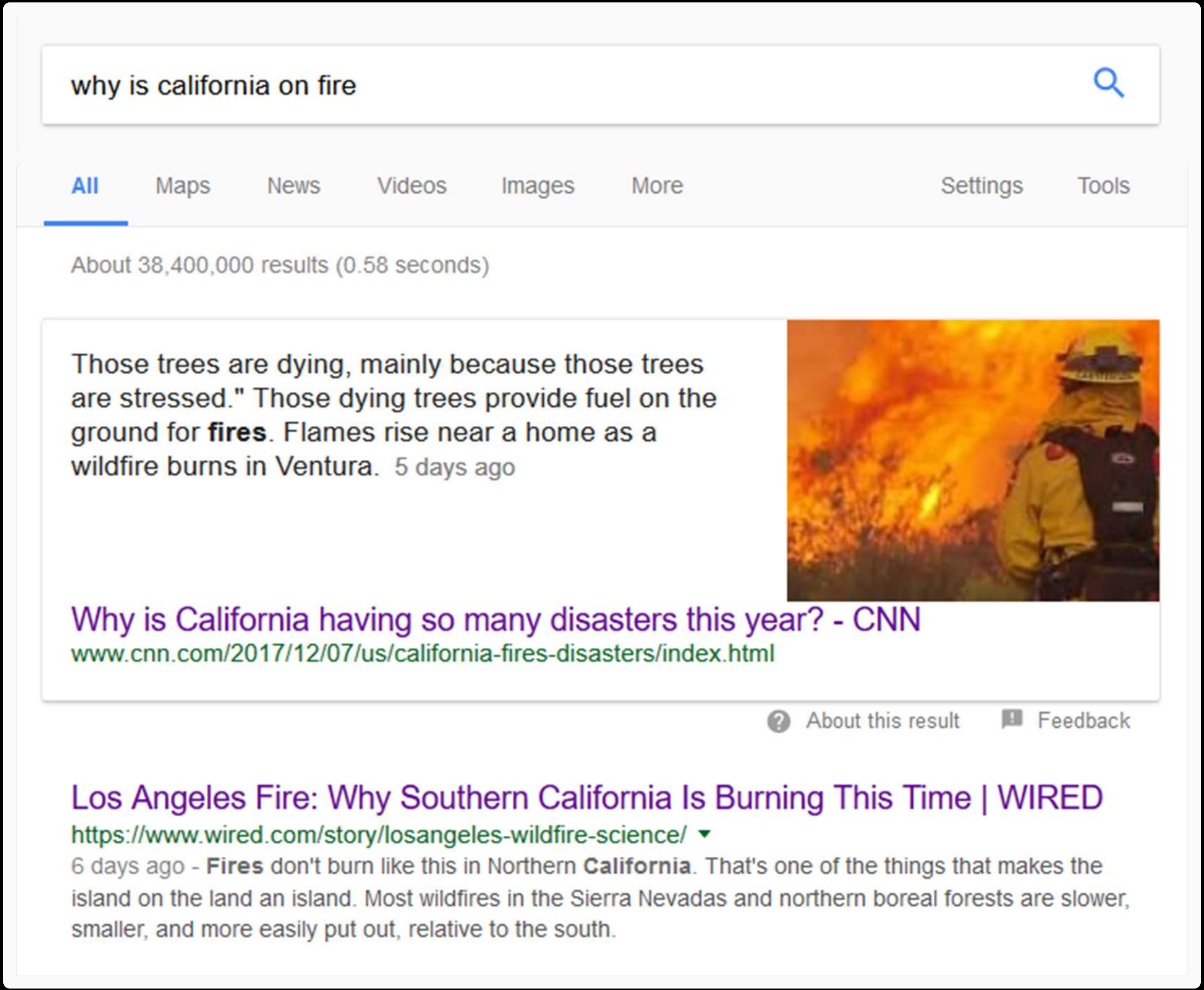
Jesse Dunietz
Thesis oral
December 14, 2017

What Google displays for “why” questions



why is california on fire

What Google displays for “why” questions



why is california on fire

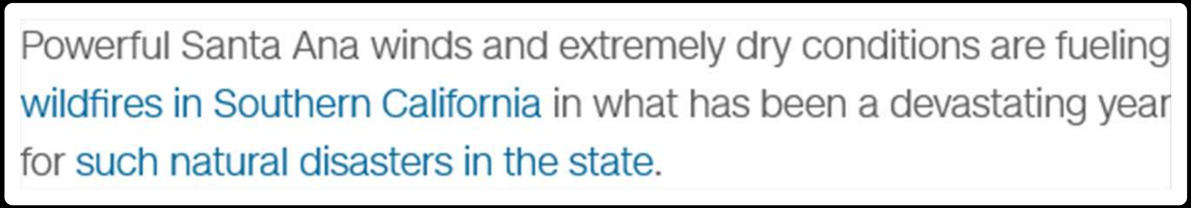
About 38,400,000 results (0.58 seconds)

Those trees are dying, mainly because those trees are stressed." Those dying trees provide fuel on the ground for **fires**. Flames rise near a home as a wildfire burns in Ventura. 5 days ago

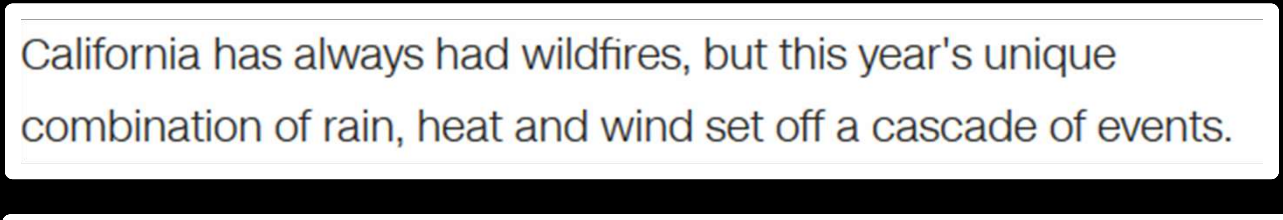
Why is California having so many disasters this year? - CNN
www.cnn.com/2017/12/07/us/california-fires-disasters/index.html

Los Angeles Fire: Why Southern California Is Burning This Time | WIRED
https://www.wired.com/story/losangeles-wildfire-science/ ▾
6 days ago - **Fires** don't burn like this in Northern **California**. That's one of the things that makes the island on the land an island. Most wildfires in the Sierra Nevadas and northern boreal forests are slower, smaller, and more easily put out, relative to the south.

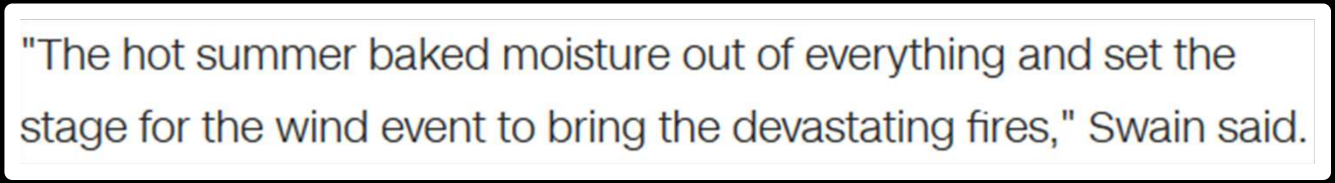
What Google displays for “why” questions could be a lot more helpful.



Powerful Santa Ana winds and extremely dry conditions are fueling wildfires in Southern California in what has been a devastating year for such natural disasters in the state.

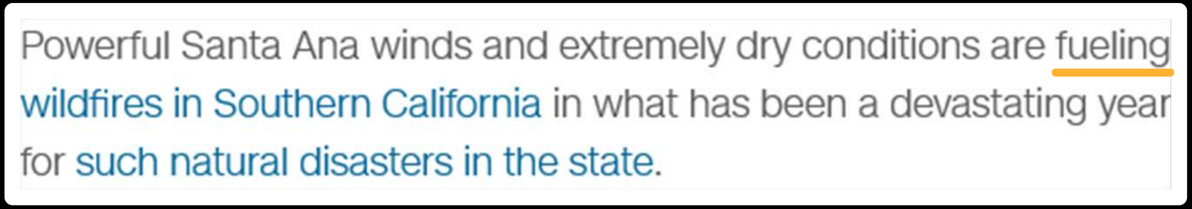


California has always had wildfires, but this year's unique combination of rain, heat and wind set off a cascade of events.

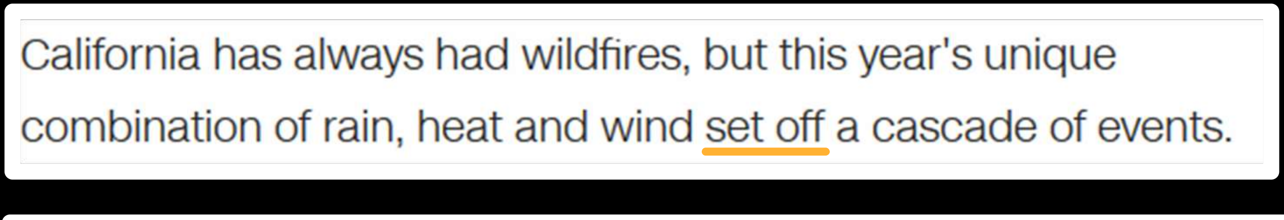


"The hot summer baked moisture out of everything and set the stage for the wind event to bring the devastating fires," Swain said.

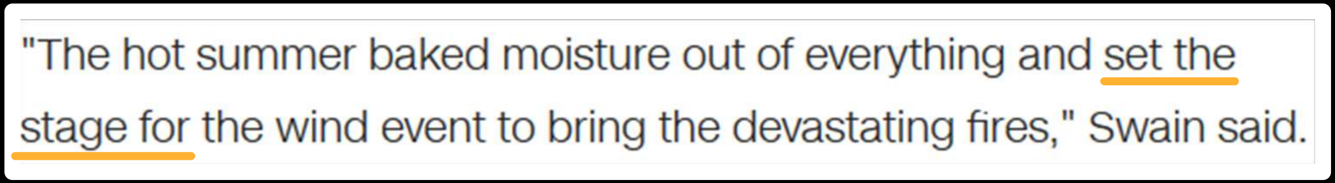
What Google displays for "why" questions could be a lot more helpful.



Powerful Santa Ana winds and extremely dry conditions are fueling wildfires in Southern California in what has been a devastating year for such natural disasters in the state.



California has always had wildfires, but this year's unique combination of rain, heat and wind set off a cascade of events.



"The hot summer baked moisture out of everything and set the stage for the wind event to bring the devastating fires," Swain said.

Such cause-and-effect questions & assertions are far from rare.

33% of explicit relations between French verbs
(Conrath et al. 2011)

12% of explicit discourse connectives
in Penn Discourse Treebank
(Prasad et al., 2008)

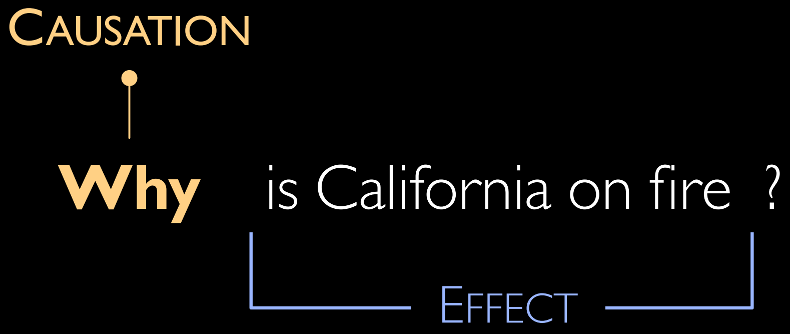
Such cause-and-effect questions & assertions are far from rare.

33% of explicit relations between French verbs
(Conrath et al. 2011)

12% of explicit discourse connectives
in Penn Discourse Treebank
(Prasad et al., 2008)

>5% and among the most complex
of questions asked to question-answering systems
(Verberne et al., 2010)

We'd like to be able to parse causal relationships in text.

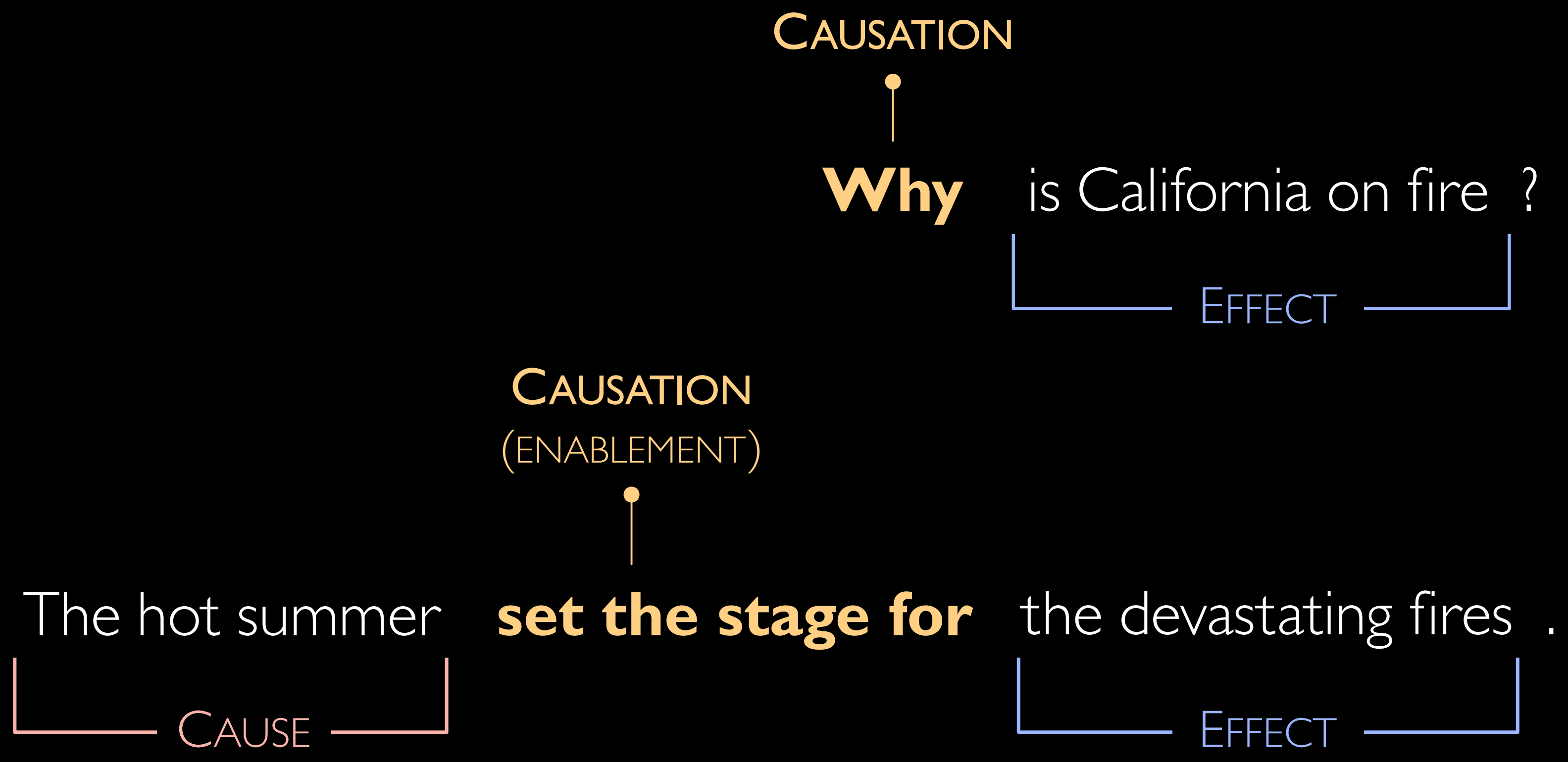


CAUSATION

Why is California on fire ?

EFFECT

We'd like to be able to parse causal relationships in text.



CAUSATION

Why is California on fire ?

EFFECT

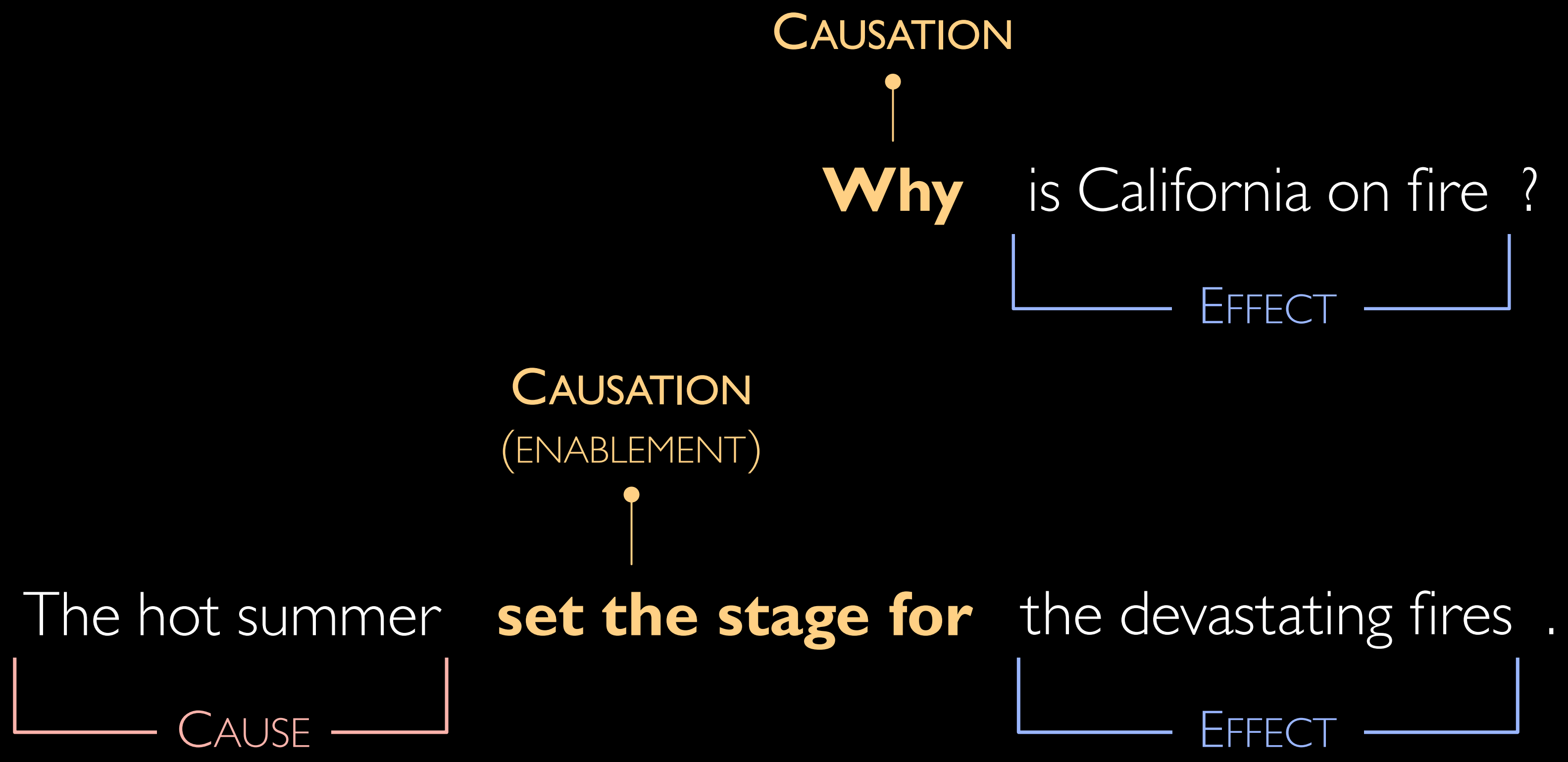
CAUSATION (ENABLEMENT)

The hot summer **set the stage for** the devastating fires .

CAUSE

EFFECT

This style of analysis is known as "shallow semantic parsing."



CAUSATION

Why is California on fire ?

EFFECT

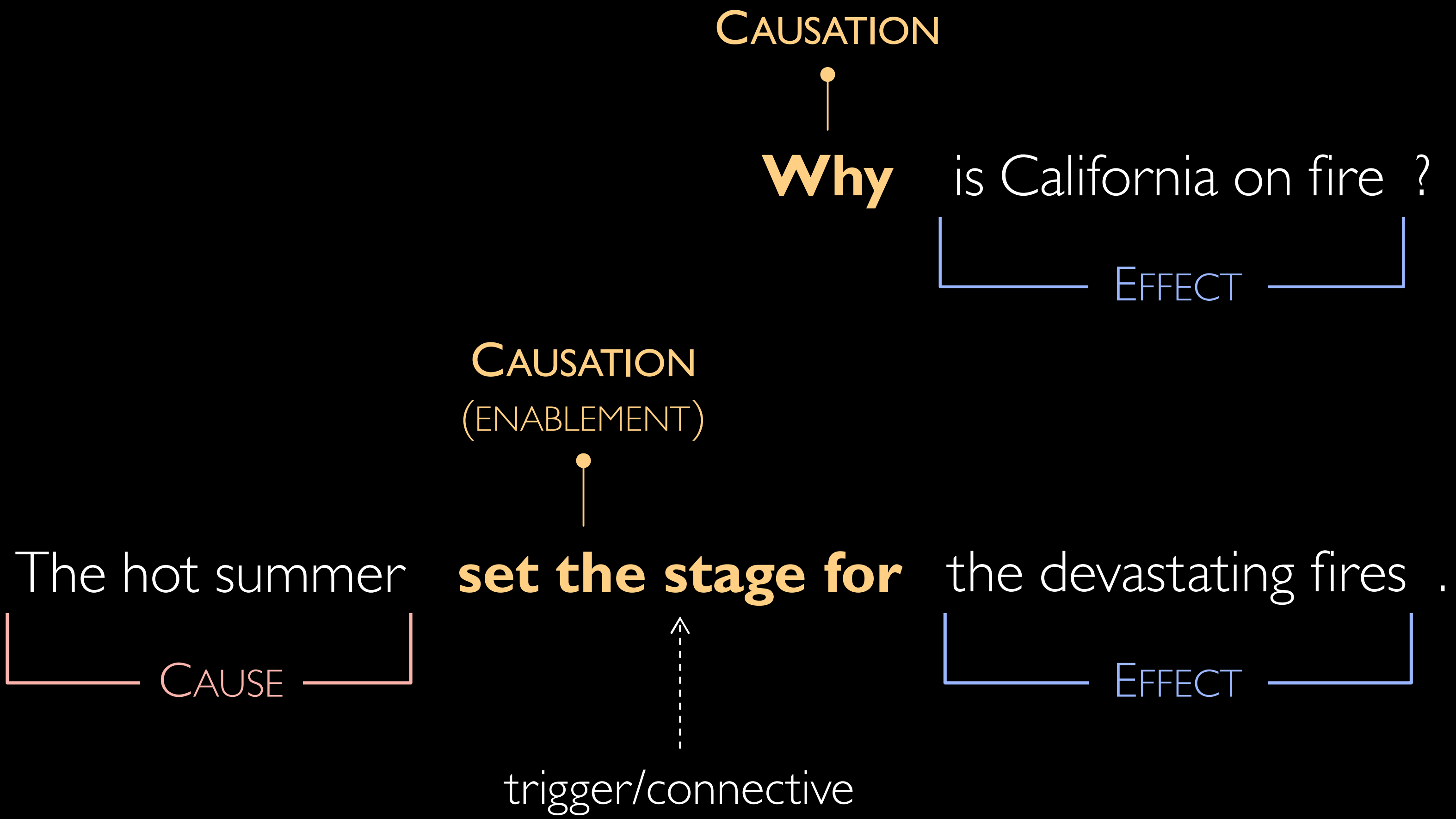
CAUSATION
(ENABLEMENT)

The hot summer **set the stage for** the devastating fires .

CAUSE

EFFECT

This style of analysis is known as “shallow semantic parsing.”



CAUSATION

Why is California on fire ?

EFFECT

CAUSATION (ENABLEMENT)

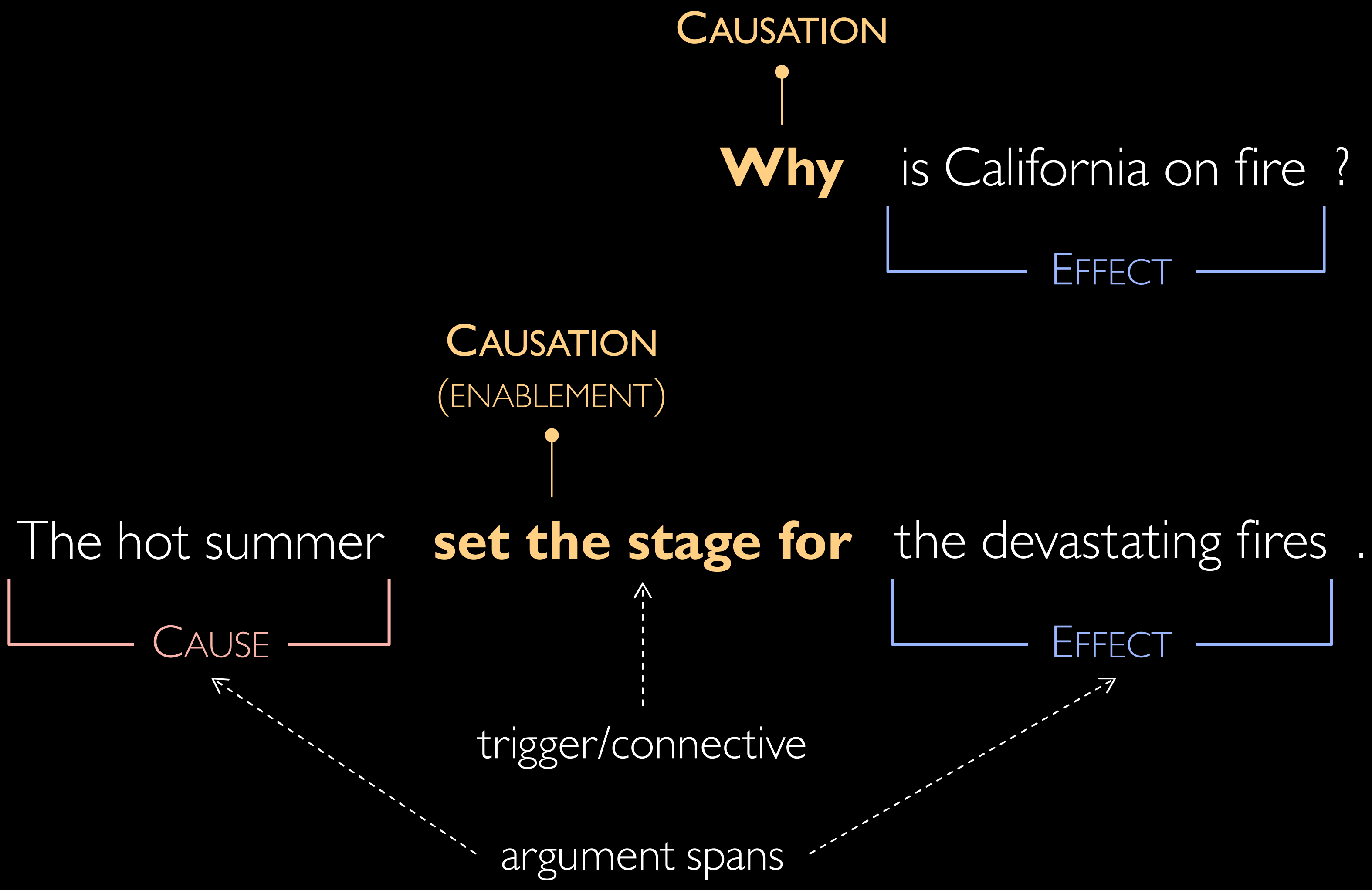
The hot summer **set the stage for** the devastating fires .

CAUSE

EFFECT

trigger/connective

This style of analysis is known as "shallow semantic parsing."



CAUSATION

Why is California on fire ?

EFFECT

CAUSATION

(ENABLEMENT)

The hot summer **set the stage for** the devastating fires .

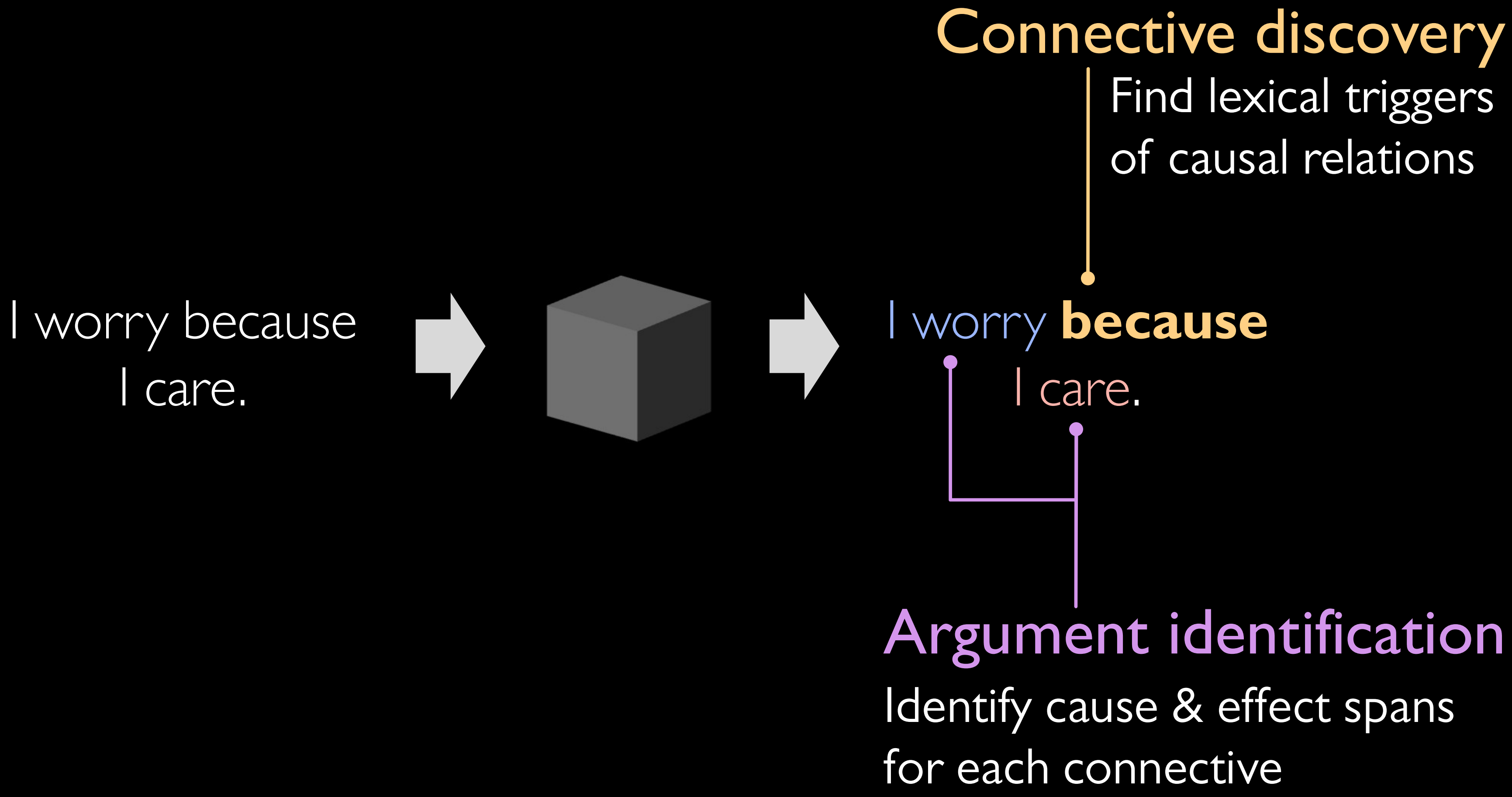
CAUSE

EFFECT

trigger/connective

argument spans

Task definition: connective discovery + argument identification



I worry because I care.

I worry **because** I care.

Connective discovery

Find lexical triggers of causal relations

Argument identification

Identify cause & effect spans for each connective

The catch:
Causality is expressed
in an enormous variety of ways.

The catch:
Causality is expressed in an enormous variety of ways.

Such swelling can **impede** breathing. (Verbs)

They moved **because of** the schools. (Prepositions)

We're running late, **so** let's move quickly. (Conjunctions)

The catch:
Causality is expressed in an enormous variety of ways.

Such swelling can **impede** breathing.	(Verbs)
They moved **because of** the schools.	(Prepositions)
We're running late, **so** let's move quickly.	(Conjunctions)
This **opens the way for** broader regulation.	(Multi-word expr.s)

The catch:
Causality is expressed in an enormous variety of ways.

Such swelling can **impede** breathing.	(Verbs)
They moved **because of** the schools.	(Prepositions)
We’re running late, **so** let’s move quickly.	(Conjunctions)
This **opens the way for** broader regulation.	(Multi-word expr.s)
Judy’s comments were **so** offensive **that** I left.	(Complex)

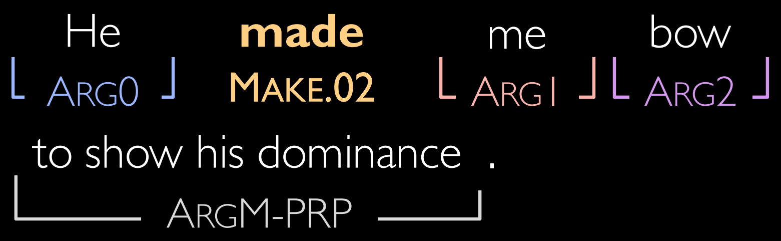
The catch:
Causality is expressed in an enormous variety of ways.

Such swelling can **impede** breathing.	(Verbs)
They moved **because of** the schools.	(Prepositions)
We're running late, **so** let's move quickly.	(Conjunctions)
This **opens the way for** broader regulation.	(Multi-word expr.s)
Judy's comments were **so** offensive **that** I left.	(Complex)
After a drink, she felt much better.	(Temporal)
The more I read his work, **the less** I like it.	(Correlation)

Each existing semantic parsing representation handles only a portion of this space.

Each existing semantic parsing representation handles only a portion of this space.

PropBank
(Palmer et al., 2005)



He **made** me bow to show his dominance .

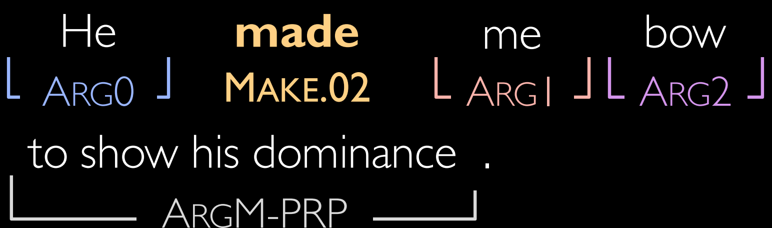
ARG0 MAKE.02 ARG1 ARG2

ARGM-PRP

Verbs only

Each existing semantic parsing representation handles only a portion of this space.

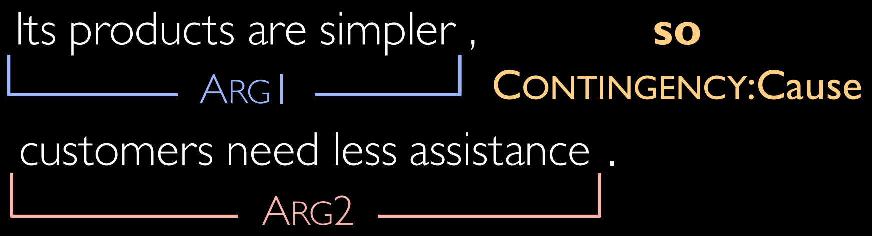
PropBank
(Palmer et al., 2005)


He **made** me bow to show his dominance .

ARG0 MAKE.02 ARG1 ARG2 ARGM-PRP

Verbs only

Penn Discourse Treebank
(Prasad et al., 2008)


Its products are simpler , **so** customers need less assistance .

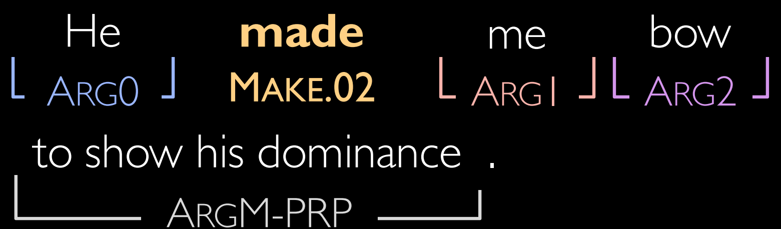
ARG1 CONTINGENCY:Cause ARG2

Conjunctions & adverbs only

Each existing semantic parsing representation handles only a portion of this space.

PropBank

(Palmer et al., 2005)



He **made** me bow

ARG0 MAKE.02 ARG1 ARG2

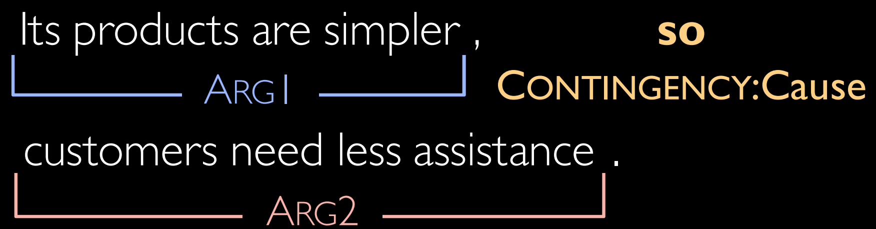
to show his dominance .

ARGM-PRP

Verbs only

Penn Discourse Treebank

(Prasad et al., 2008)



Its products are simpler , **so**

ARG1 CONTINGENCY:Cause

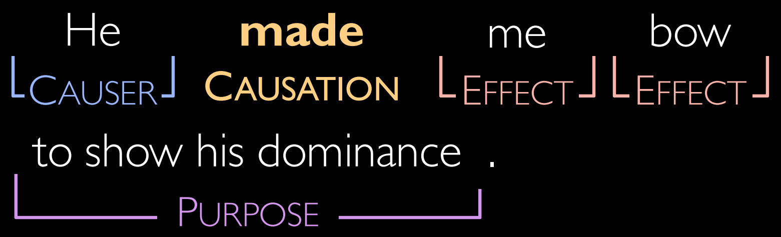
customers need less assistance .

ARG2

Conjunctions & adverbs only

FrameNet

(Fillmore & Baker, 2010; Ruppenhofer et al., 2016)



He **made** me bow

CAUSER CAUSATION EFFECT EFFECT

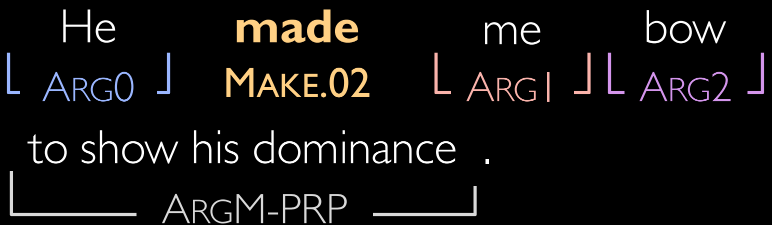
to show his dominance .

PURPOSE

Words or constituents only

Each existing semantic parsing representation handles only a portion of this space.

PropBank
(Palmer et al., 2005)

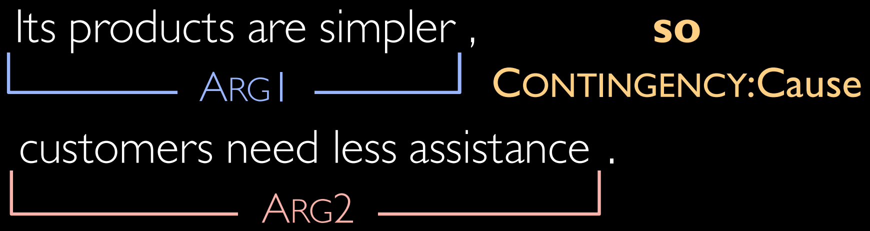

He **made** me bow to show his dominance .

ARG0 MAKE.02 ARG1 ARG2

ARGM-PRP

Verbs only

Penn Discourse Treebank
(Prasad et al., 2008)

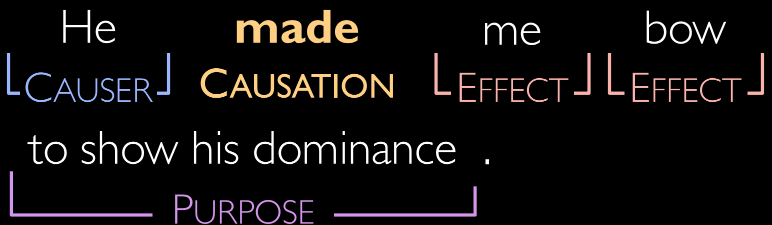

Its products are simpler , **so** customers need less assistance .

ARG1 CONTINGENCY:Cause

ARG2

Conjunctions & adverbs only

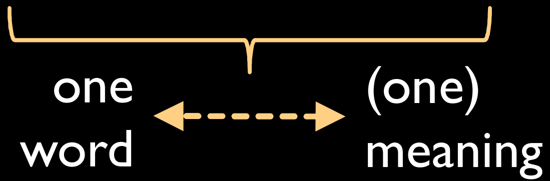
FrameNet
(Fillmore & Baker, 2010;
Ruppenhofer et al., 2016)


He **made** me bow to show his dominance .

CAUSER CAUSATION EFFECT EFFECT

PURPOSE

Words or constituents only


one word <- - - - - -> (one) meaning

Construction grammar (CxG) offers a way forward.

Construction

(Fillmore et al., 1988; Goldberg, 1995)

Construction grammar (CxG) offers a way forward.

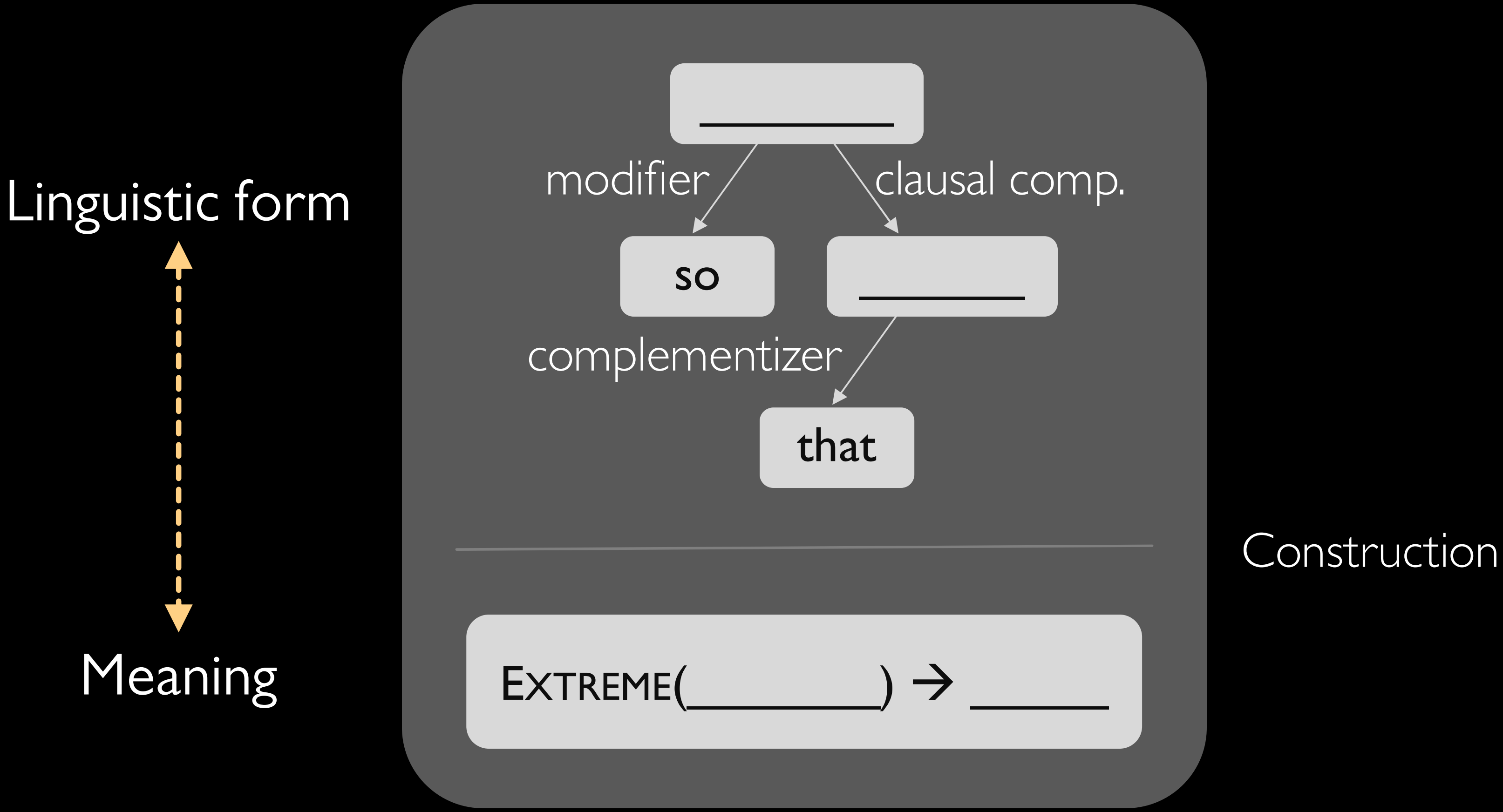
Linguistic form

Meaning

Construction

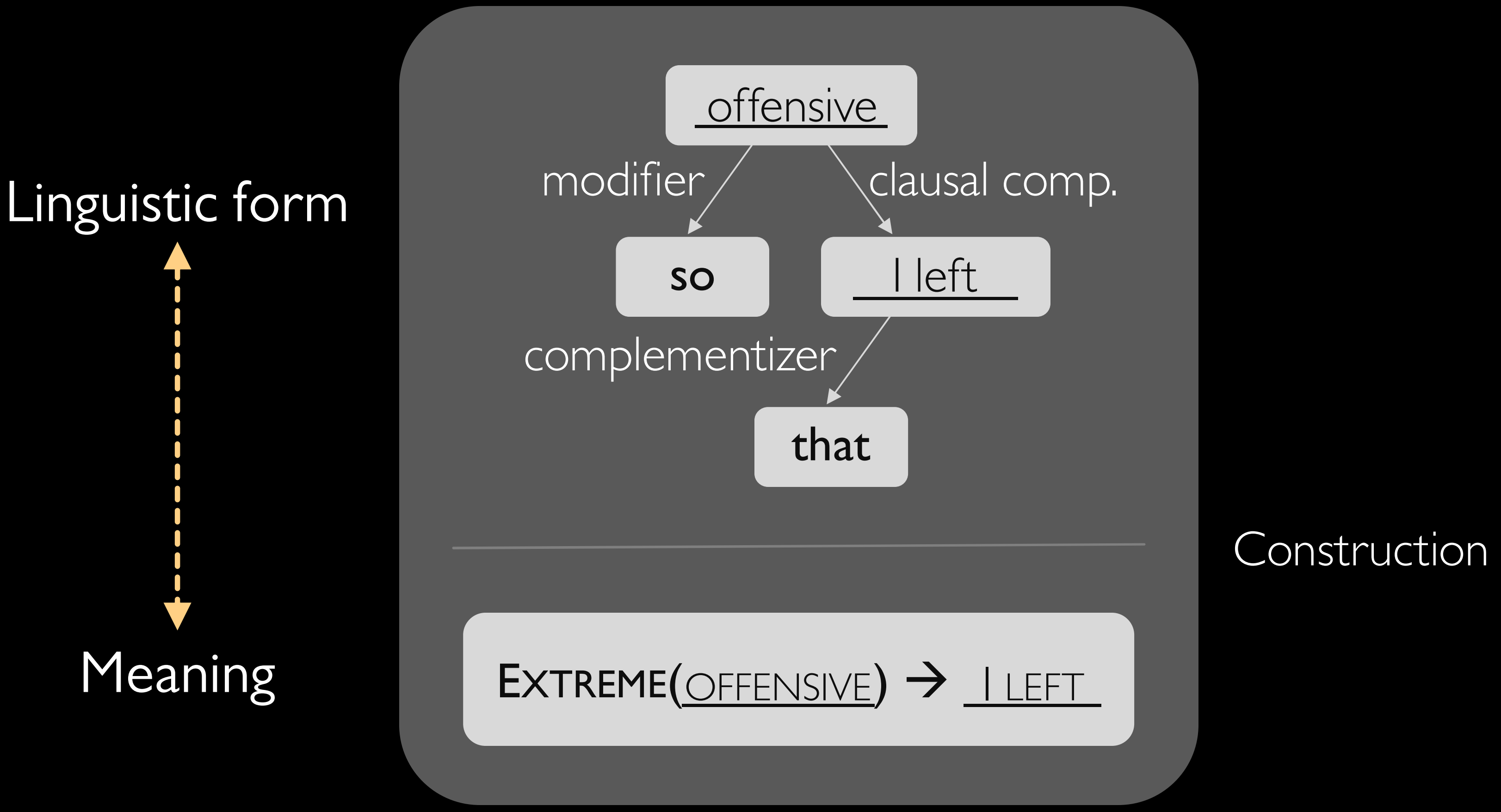
(Fillmore et al., 1988; Goldberg, 1995)

Construction grammar (CxG) offers a way forward.



(Fillmore et al., 1988; Goldberg, 1995)

Construction grammar (CxG) offers a way forward.



(Fillmore et al., 1988; Goldberg, 1995)

(It’s not just causality, either.)

(It’s not just causality, either.)

Comparatives

You’re **as** bad **as** my mom!

More boys wanted to participate **than** girls.

Andrew is **as** annoying **as** he **is** useless.

(It’s not just causality, either.)

Comparatives

You’re **as** bad **as** my mom!

More boys wanted to participate **than** girls.

Andrew is **as** annoying **as** he **is** useless.

Concessives

We headed out **in spite of** the awful weather.

We value any contribution, **no matter** its size.

Strange **as** it seems,
there’s been a run of crazy dreams!

Full CxG theory means “constructions all the way down”:

so offensive that I left

(see Goldberg, 2006)

Full CxG theory means “constructions all the way down”:

so offensive that I left

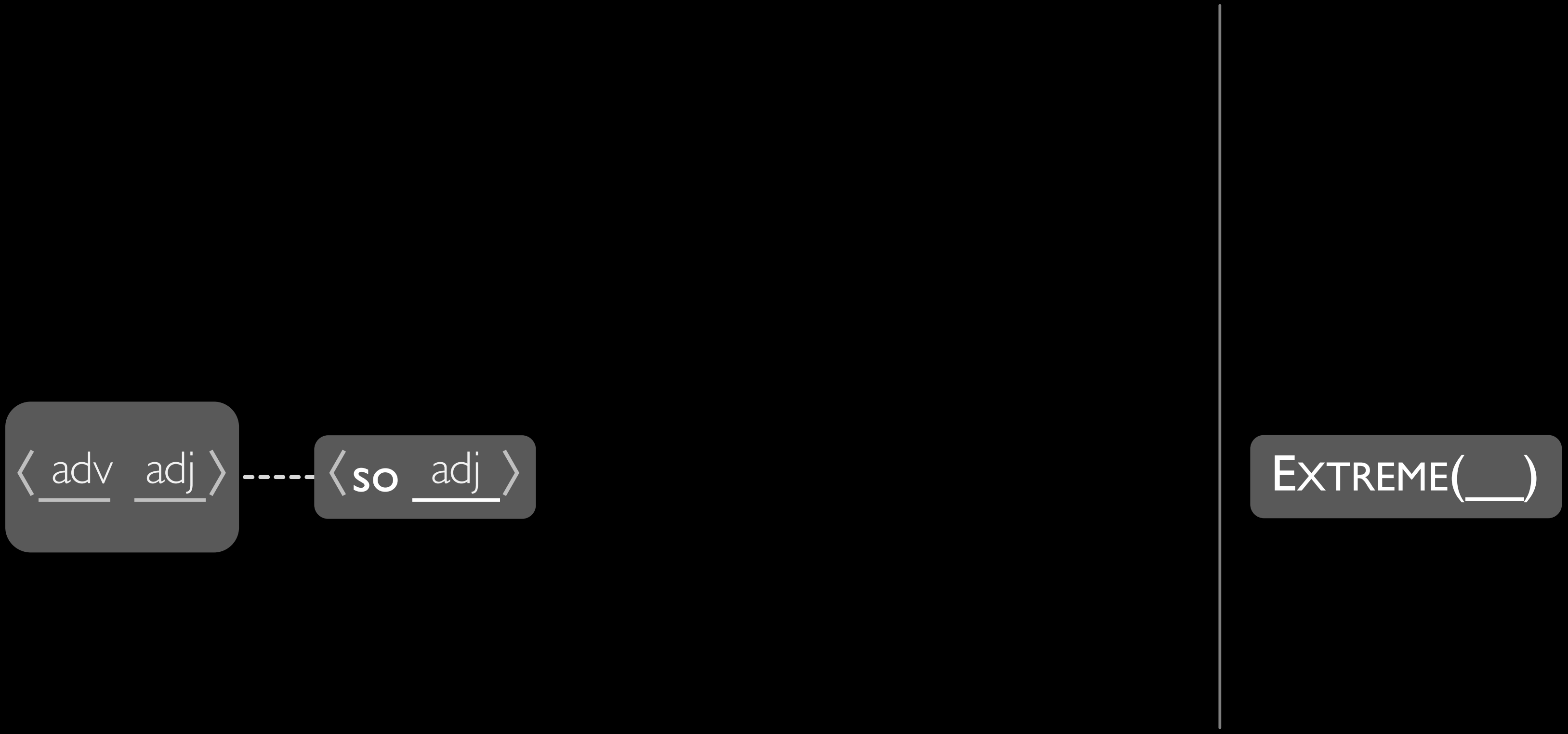
⟨so adj⟩

EXTREME(___)

(see Goldberg, 2006)

Full CxG theory means “constructions all the way down”:

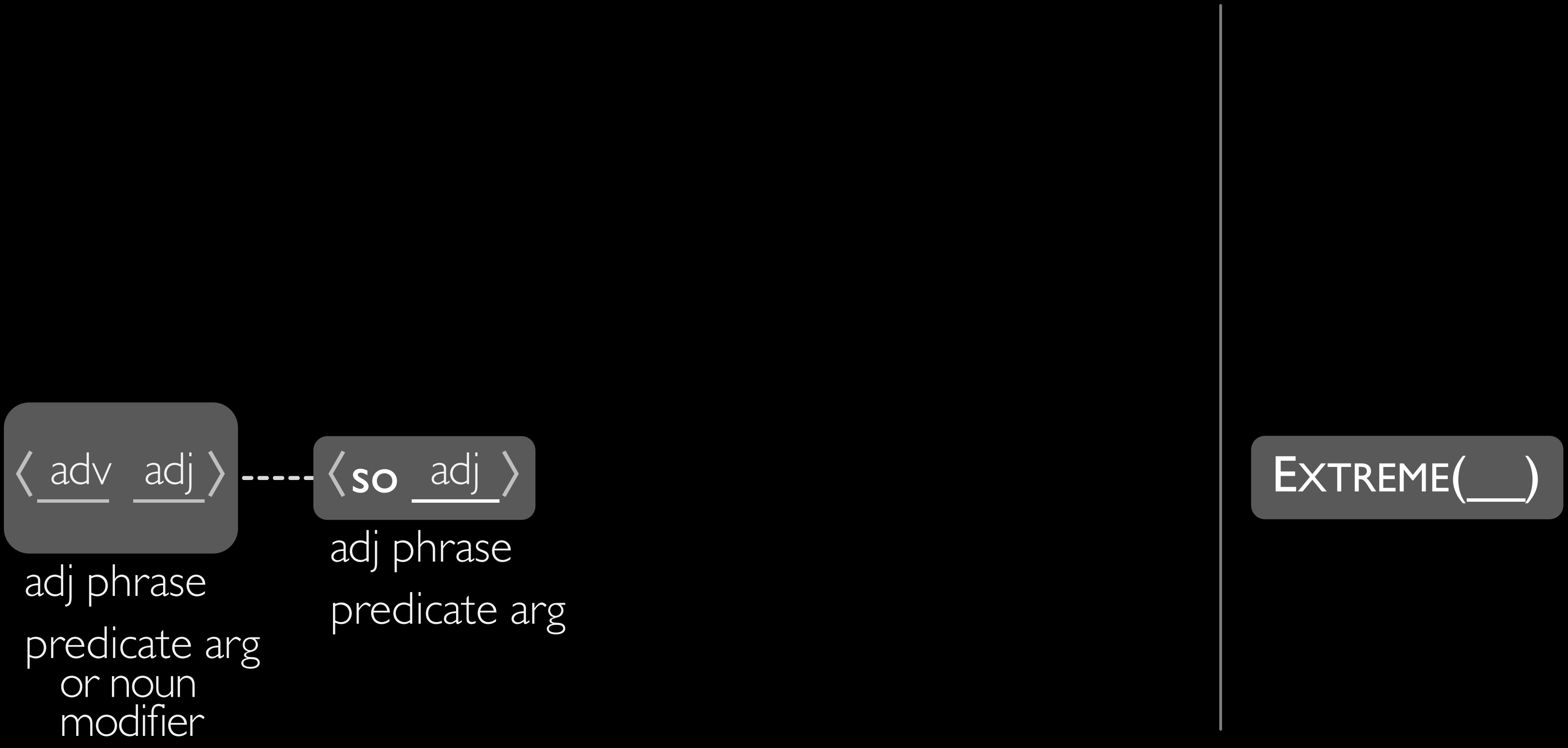
so offensive that I left



(see Goldberg, 2006)

Full CxG theory means "constructions all the way down":

so offensive that I left



⟨ adv ___ adj ___ ⟩ ----- ⟨ so ___ adj ⟩

adj phrase

predicate arg or noun modifier

adj phrase

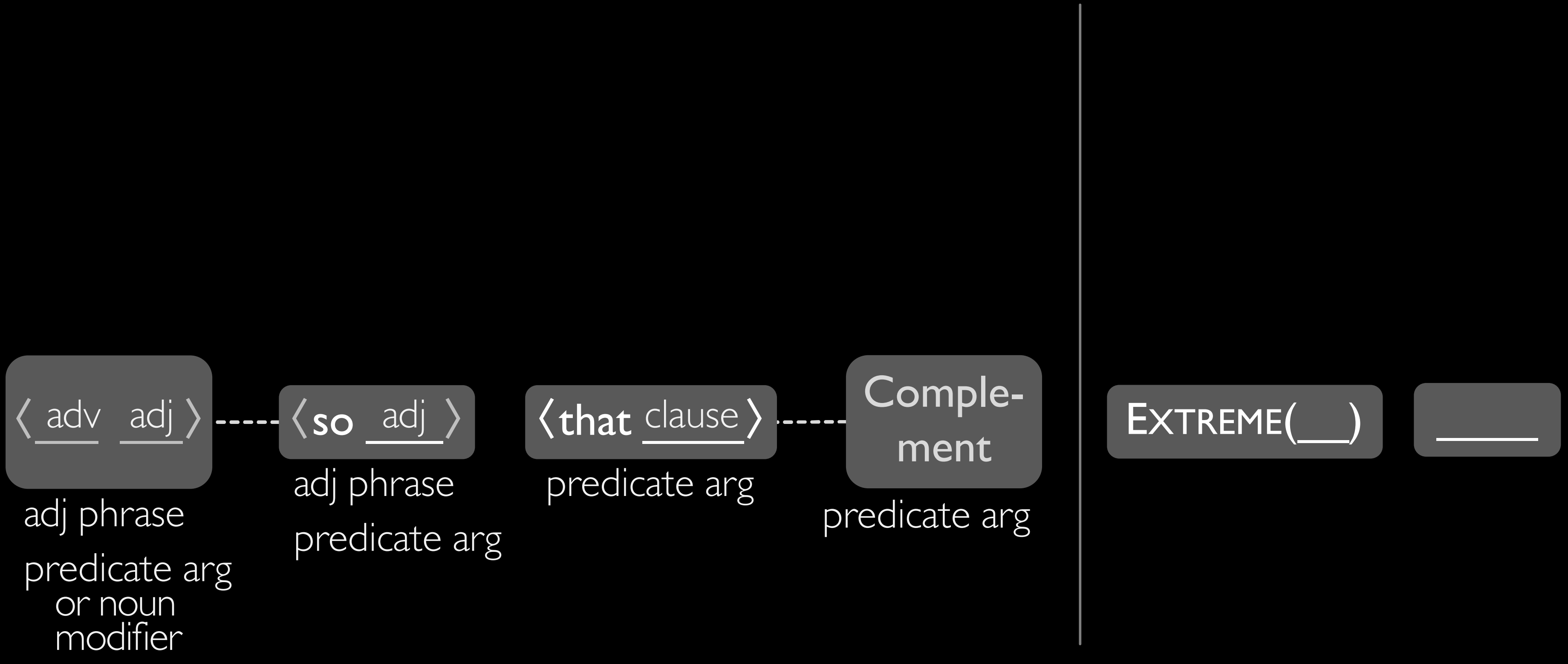
predicate arg

EXTREME(___)

(see Goldberg, 2006)

Full CxG theory means “constructions all the way down”:

so offensive that I left



⟨ adv adj ⟩

adj phrase

predicate arg or noun modifier

⟨so adj ⟩

adj phrase

predicate arg

⟨that clause⟩

predicate arg

Comple-ment

predicate arg

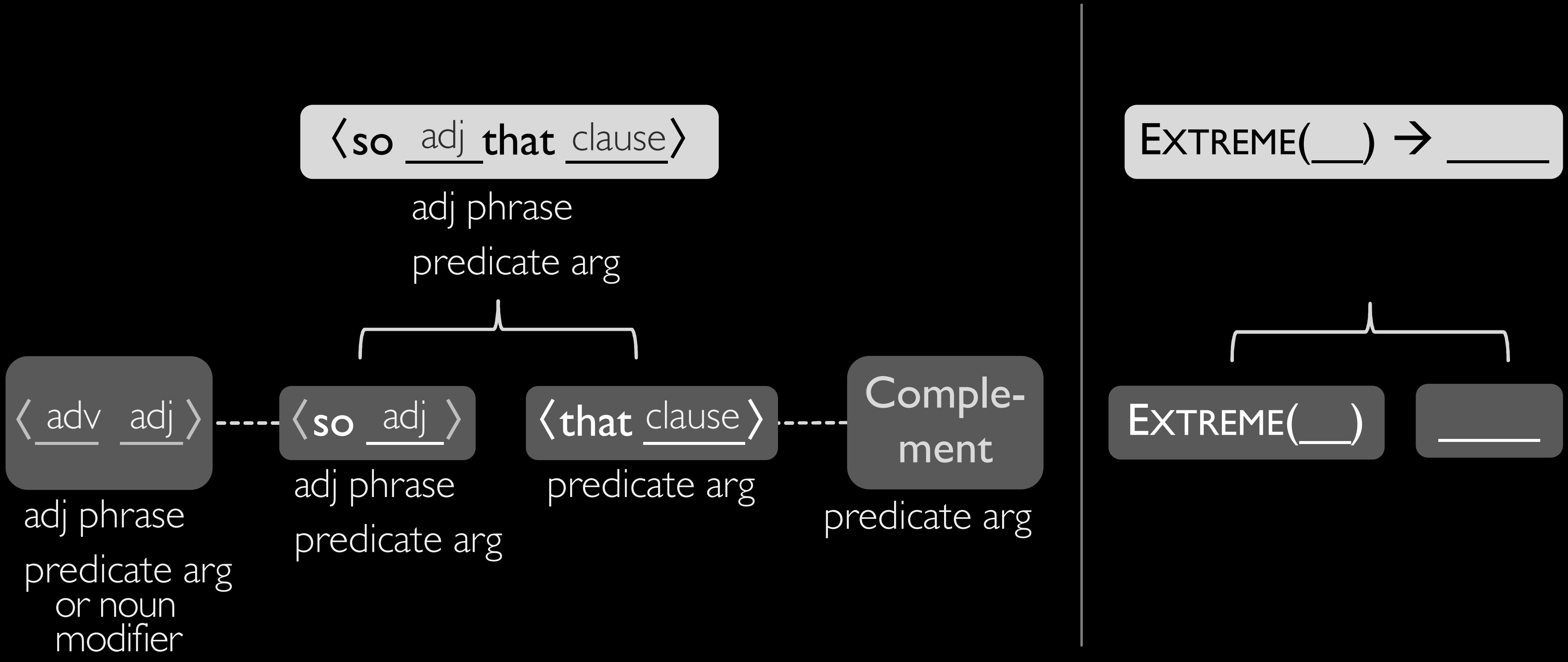
EXTREME(__)

____

(see Goldberg, 2006)

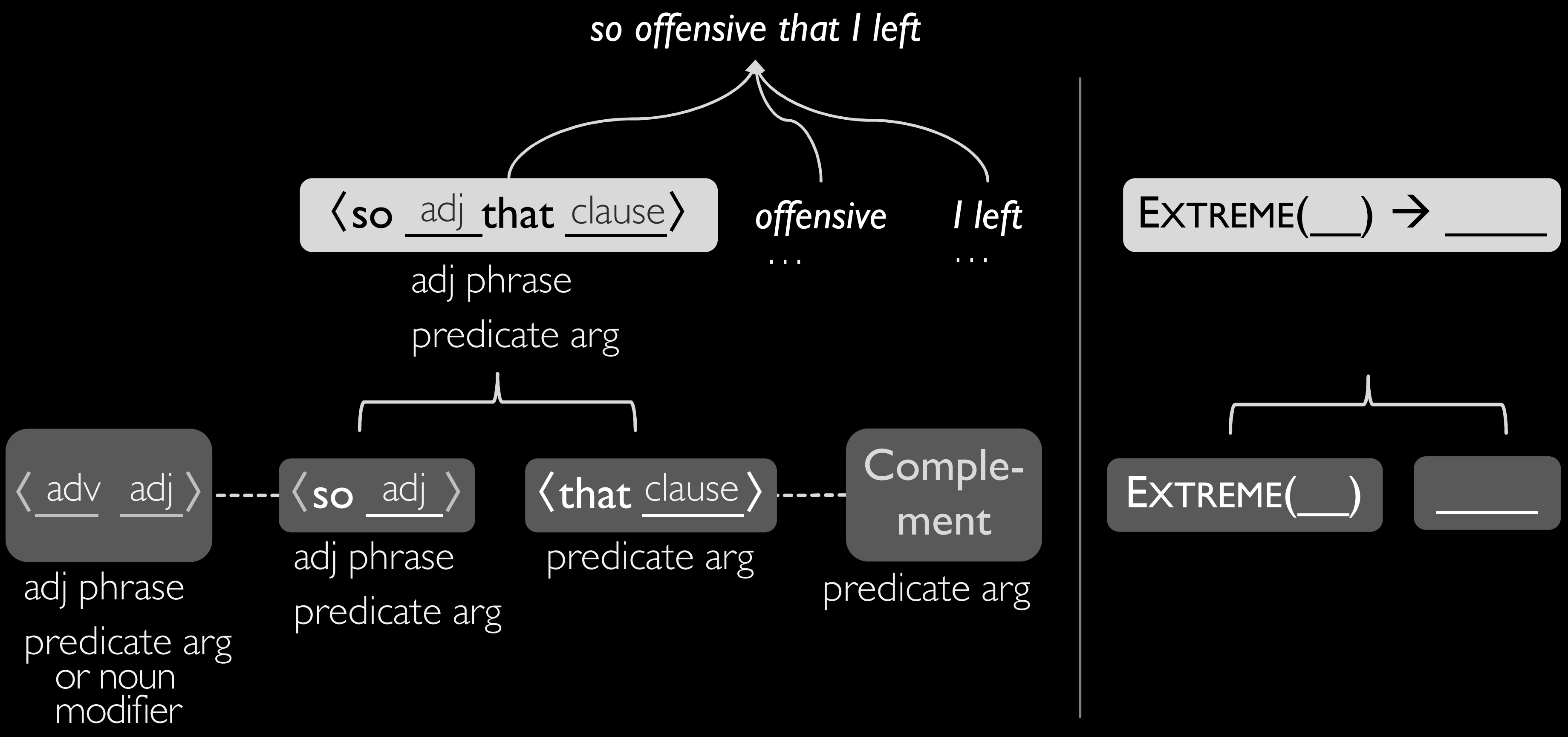
Full CxG theory means "constructions all the way down":

so offensive that I left



(see Goldberg, 2006)

Full CxG theory means
"constructions all the way down":



so offensive that I left

⟨**so** adj **that** clause⟩

adj phrase

predicate arg

offensive

. . .

I left

. . .

EXTREME(__) → _____

⟨ adv adj ⟩

adj phrase

predicate arg or noun modifier

⟨**so** adj ⟩

adj phrase

predicate arg

⟨**that** clause⟩

predicate arg

Comple- ment

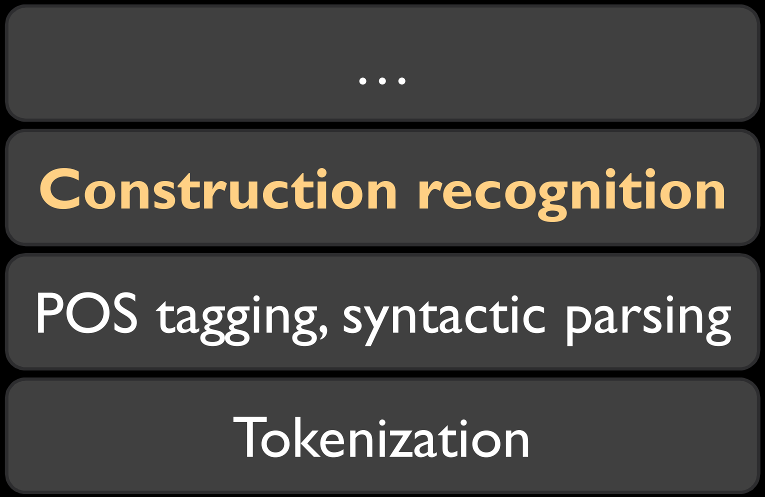
predicate arg

EXTREME(__)

_____

(see Goldberg, 2006)

The “constructions on top” approach reaps the low-hanging fruit from applying CxG to NLP.



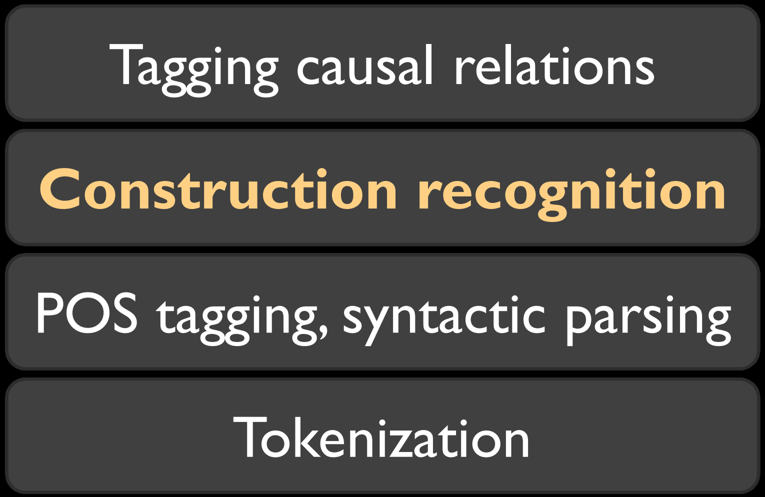
…

Construction recognition

POS tagging, syntactic parsing

Tokenization

The “constructions on top” approach reaps the low-hanging fruit from applying CxG to NLP.



Tagging causal relations

Construction recognition

POS tagging, syntactic parsing

Tokenization

“Constructions on top” borrows two key insights of CxG.

1. Words, multi-word expressions, and grammar are all on equal footing as “learned pairings of form and function.”

2. Constructions pair patterns of surface forms directly with meanings.

(see Goldberg, 2013)

Thesis statement:
Using the "constructions on top" approach to applying CxG, we can:

Thesis statement:

Using the "constructions on top" approach to applying CxG, we can:

- Improve shallow semantic parsing coverage using richer, more flexible linguistic representations.

Thesis statement:
Using the "constructions on top" approach to applying CxG, we can:

- Improve shallow semantic parsing coverage using richer, more flexible linguistic representations.

- Design annotation guidelines & annotate a corpus using these representations.

Thesis statement:
Using the "constructions on top" approach to applying CxG, we can:

- Improve shallow semantic parsing coverage using richer, more flexible linguistic representations.
- Design annotation guidelines & annotate a corpus using these representations.
- Build automated machine learning taggers for constructional realizations of semantic relations.

Today's talk:

1. The BECAUSE annotation scheme & corpus of causal language

2. Causeway-L/Causeway-S: two pattern-based taggers for causal constructions

3. DeepCx: a neural, transition-based tagger for causal constructions

Today's talk:

1. The BECAUSE annotation scheme & corpus of causal language

2. Causeway-L/Causeway-S: two pattern-based taggers for causal constructions

3. DeepCx: a neural, transition-based tagger for causal constructions

Previous projects have struggled to annotate real-world causality.

SemEval 2007 Task 4

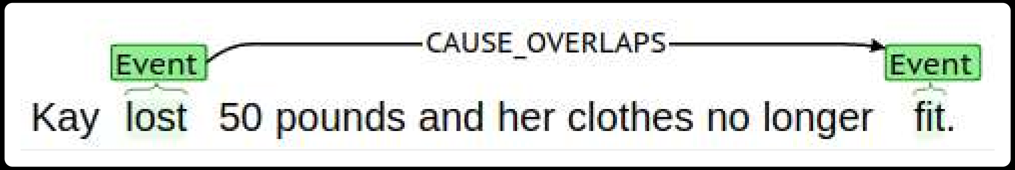
(Girju et al., 2007)

“A person infected with a <e1>flu</e1> <e2>virus</e2> strain develops antibodies against it.”

```
Cause-Effect(e2, e1) = "true"
```

CaTeRS

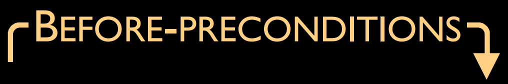
(Mostafazadeh et al., 2016)



Kay lost 50 pounds and her clothes no longer fit.

Richer Event Descriptions

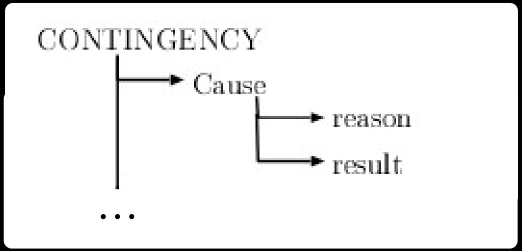
(O’Gorman et al., 2016; Croft et al., 2016)



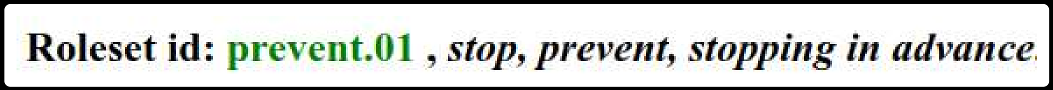
We’ve **allocated** a budget to **equip** the barrier with electronic detention equipment.

Existing shallow semantic parsing schemes include some elements of causal language.

Penn Discourse Treebank
(Prasad et al., 2008)

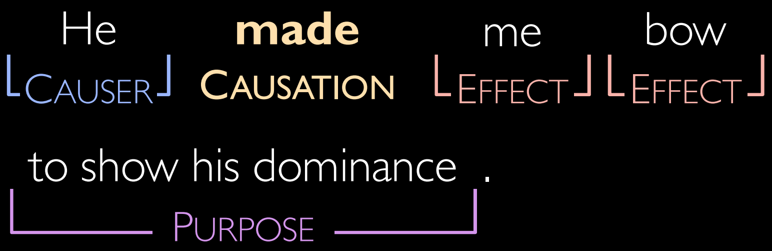


PropBank
(Palmer et al., 2005)



Roleset id: prevent.01 , ***stop, prevent, stopping in advance***

FrameNet
(Fillmore & Baker, 2010;
Ruppenhofer et al., 2016)



He **made** me bow to show his dominance .

Causal language:
a clause or phrase in which
one event, state, action, or entity
is **explicitly presented**
as promoting or hindering
another

(Dunietz et al., 2015, 2017)

Connective: fixed constructional cue indicating a causal relationship

John trapped the fox **because**
it was threatening his chickens.

John **prevented** the fox
from eating his chickens
by building a fence.

Ice cream consumption **causes** drowning.

Connective: fixed constructional cue indicating a causal relationship

John trapped the fox **because**
it was threatening his chickens.

John **prevented** the fox
from eating his chickens
by building a fence.

Ice cream consumption **causes** drowning. ………… Not “truly” causal

Cause: presented as producing effect
Effect: presented as outcome

John trapped the fox because
it was threatening his chickens.

John prevented the fox
from eating his chickens
by building a fence.

Ice cream consumption causes drowning.

Connectives can be arbitrarily complex.

For markets to work,
banks must not expect bailouts.

This opens the way for broader regulation.

We distinguish three types of causation.

The system failed **because of** a loose screw. → CONSEQUENCE

Mary left **because** John was coming. → MOTIVATION

Mary left **in order to** avoid John. → PURPOSE

Latest annotation scheme shows very good inter-annotator agreement.

	Agreement
Connective spans (F_1)	0.77
Causation types (κ)	0.70
Cause spans (% exact match | same connective)	0.89
Effect spans (% exact match | same connective)	0.86

2 trained annotators
260 sentences
98 instances of causal language

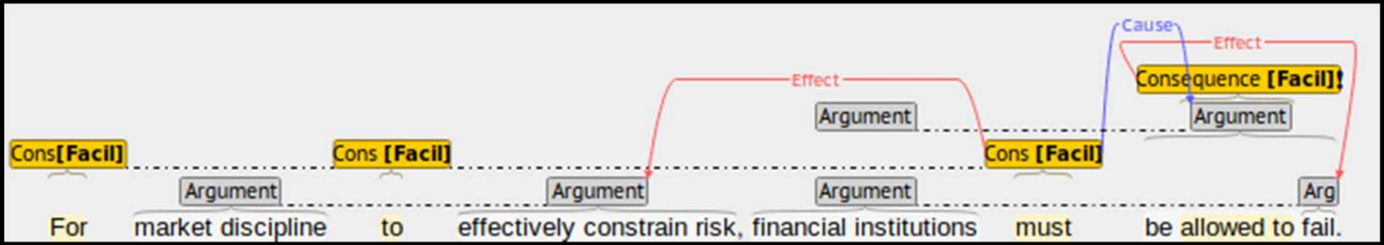
We have annotated a small corpus with this scheme.

	Documents	Sentences	Causal
New York Times Washington section (Sandhaus, 2014)	59	1924	717
Penn TreeBank WSJ	47	1542	534
2014 NLP Unshared Task in PoliInformatics (Smith et al., 2014)	3	772	324
Manually Annotated Sub-Corpus (Ide et al., 2010)	12	629	228
Total	121	4790	1803

BECAUSE = **B**ank of **E**ffects and **Cau**ses **S**tated **E**xplicitly

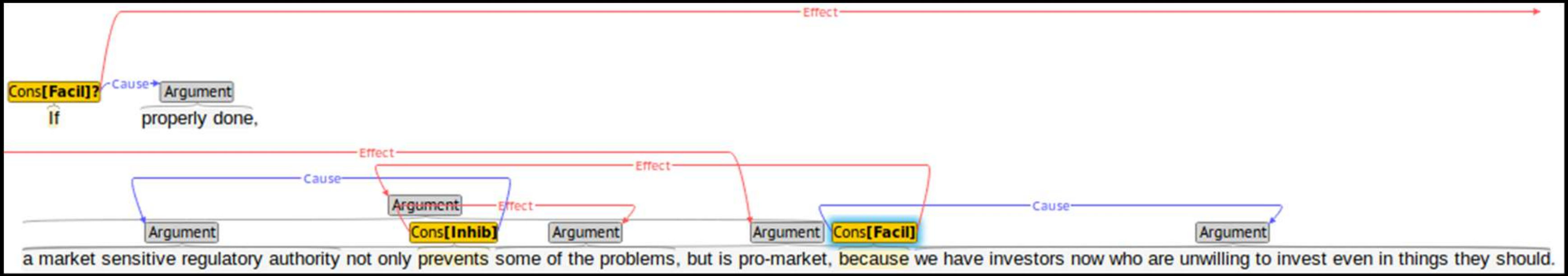
Actual corpus examples can get quite complex.

"**For** market discipline **to** effectively constrain risk,
financial institutions **must** be **allowed to** fail."



For market discipline to effectively constrain risk, financial institutions must be allowed to fail.

"**If** properly done, a market sensitive regulatory authority
not only **prevents** some of the problems, but is pro-market,
because we have investors now who are unwilling to invest even in things they should."



If properly done,

a market sensitive regulatory authority not only prevents some of the problems, but is pro-market, because we have investors now who are unwilling to invest even in things they should.

Average causal sentence length: 30 words

Today's talk:

1. The BECAUSE annotation scheme & corpus of causal language

2. Causeway-L/Causeway-S: two pattern-based taggers for causal constructions

3. DeepCx: a neural, transition-based tagger for causal constructions

Today's talk:

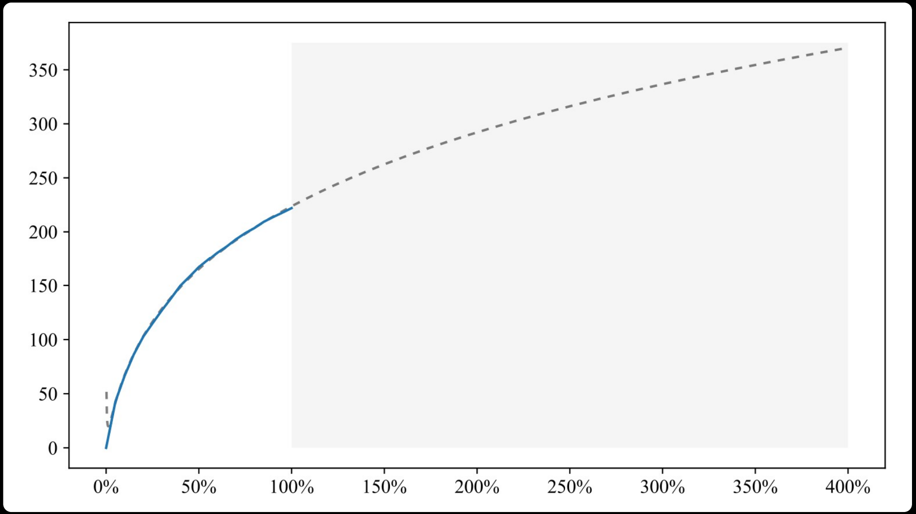
1. The BECAUSE annotation scheme & corpus of causal language

2. Causeway-L/Causeway-S: two pattern-based taggers for causal constructions

3. DeepCx: a neural, transition-based tagger for causal constructions

The computational task is challenging.

Long tail of causal connectives



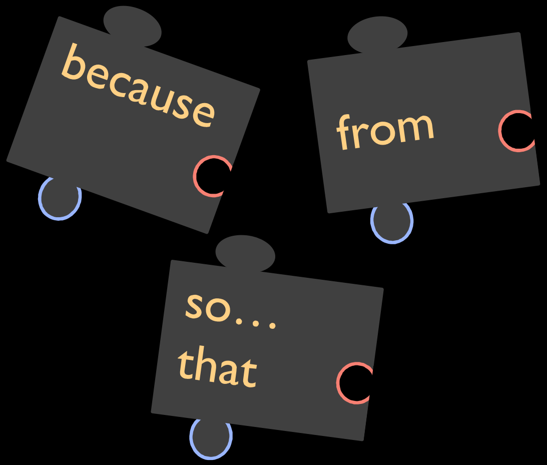
Requires sense disambiguation of connectives

e.g., "necessary **for** us **to** succeed" vs. "hard **for** me **to** do"

Complex output structure

Combinatorial connective possibilities

1. Pattern-based connective discovery



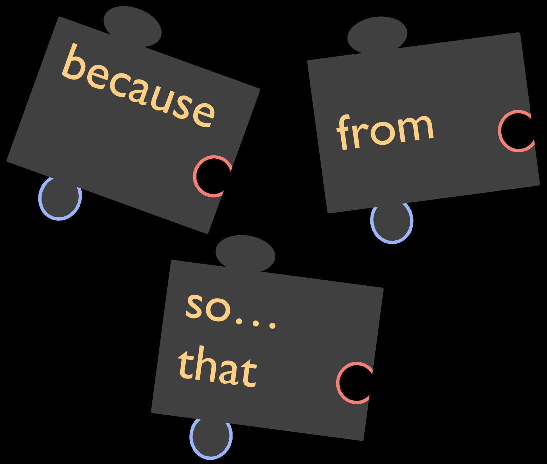
I nearly died **from** worry.

You could have called me **from** your hotel.

(Dunietz et al., 2017)

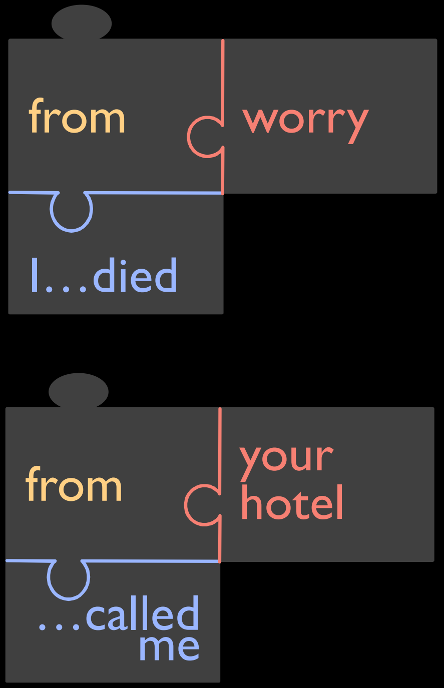
1. Pattern-based connective discovery

2. Argument identification



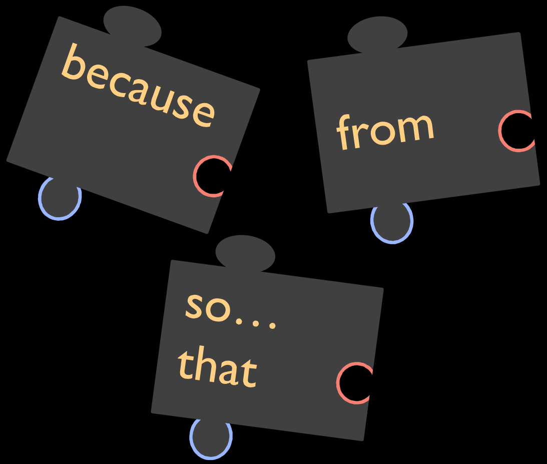
I nearly died **from** worry.

You could have called me **from** your hotel.



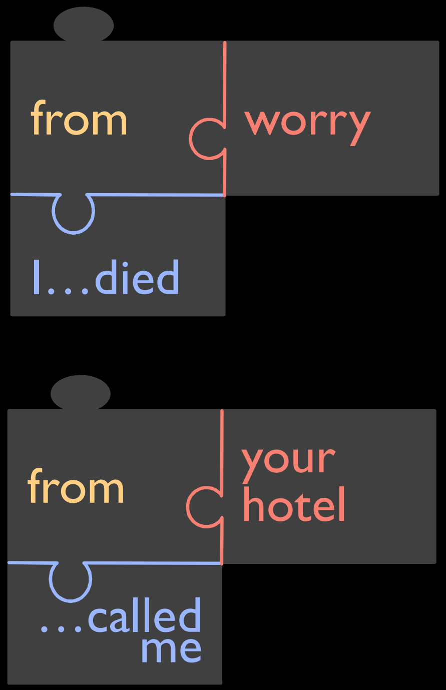
(Dunietz et al., 2017)

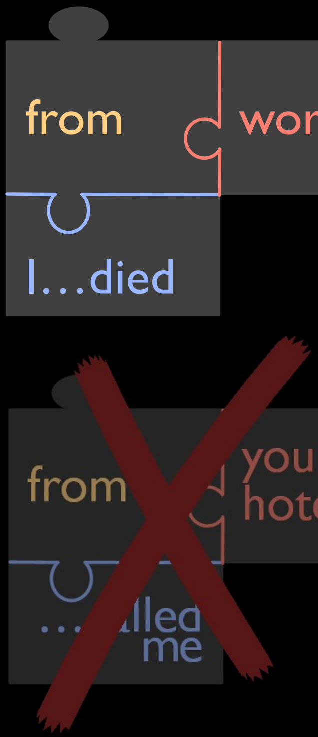
1. Pattern-based connective discovery → 2. Argument identification → 3. Statistical classifier to filter results



I nearly died **from** worry.

You could have called me **from** your hotel.



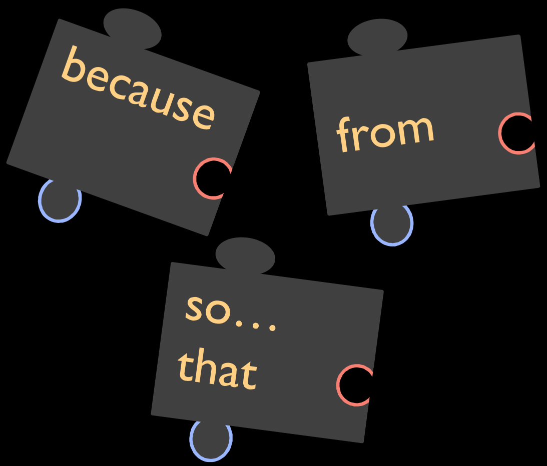


(Dunietz et al., 2017)

1. Pattern-based connective discovery

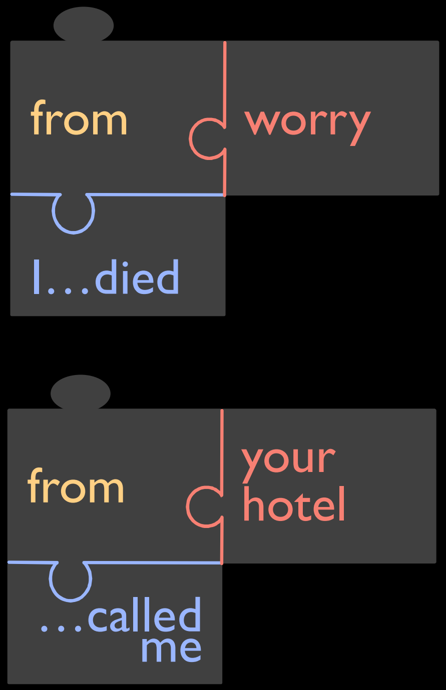
2. Argument identification

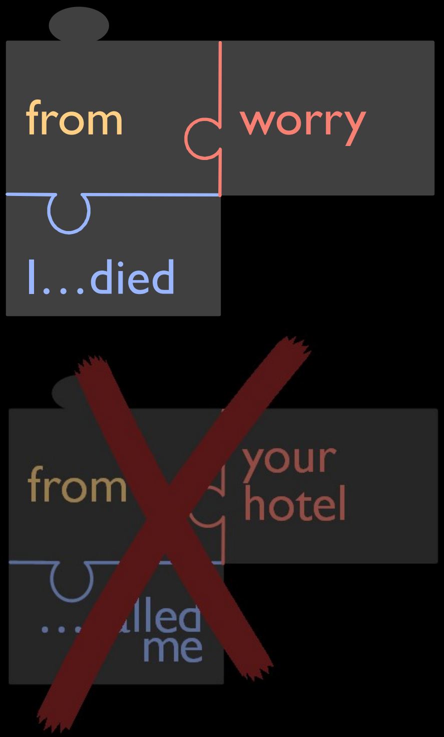
3. Statistical classifier to filter results



I nearly died **from** worry.

You could have called me **from** your hotel.

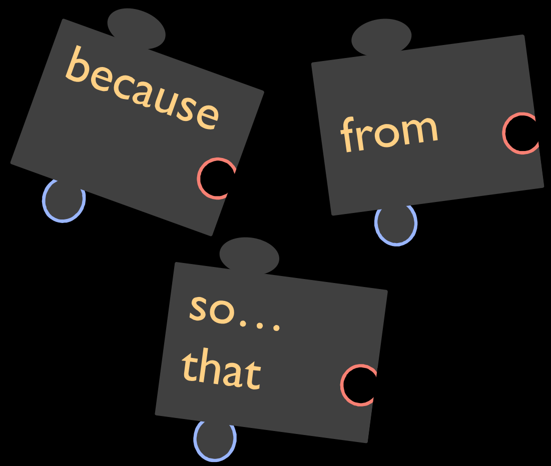




4. Remove duplicate connectives

(Dunietz et al., 2017)

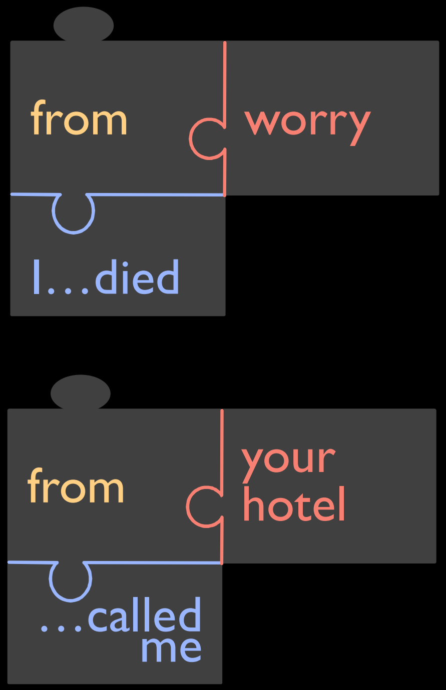
1. Pattern-based connective discovery
(tentative)



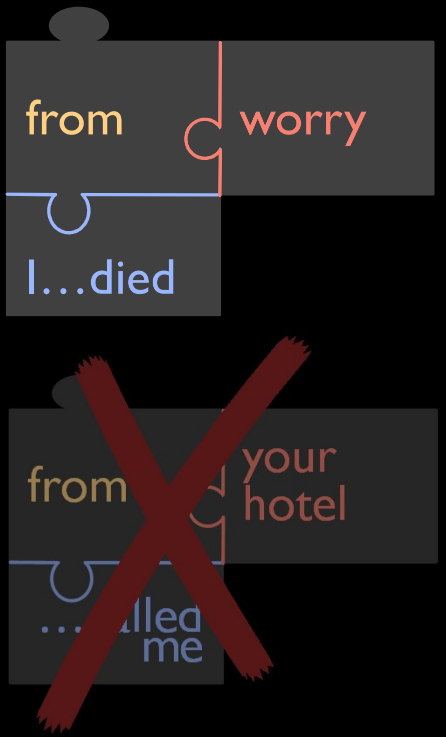
I nearly died **from** worry.

You could have called me **from** your hotel.

2. Argument identification
(tentative)



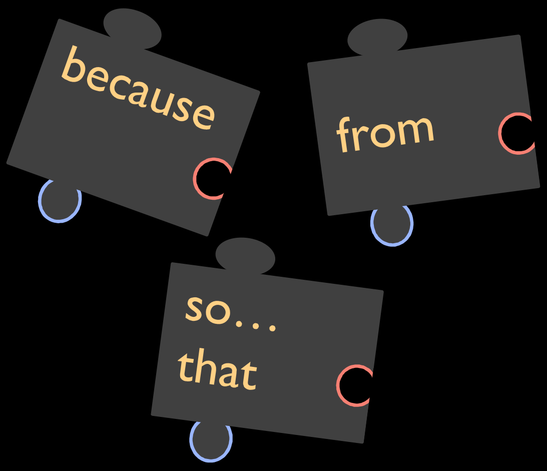
3. Statistical classifier to filter results



4. Remove duplicate connectives

(Dunietz et al., 2017)

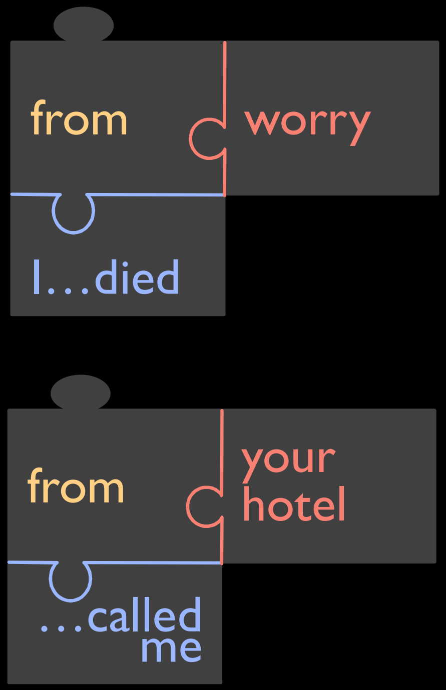
1. **Pattern-based connective discovery**
(tentative)



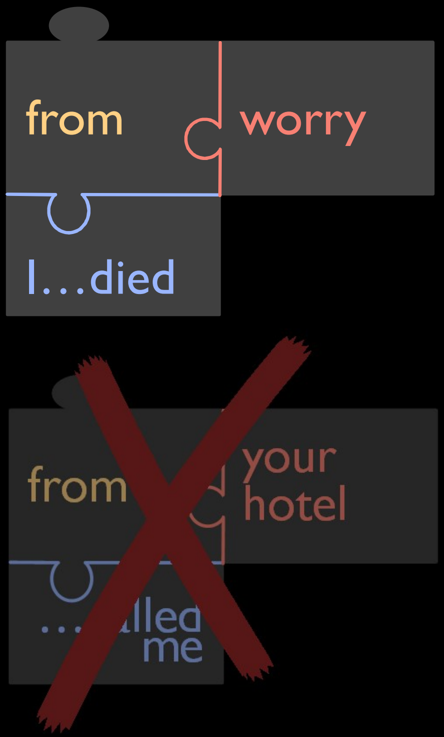
I nearly died **from** worry.

You could have called me **from** your hotel.

2. **Argument identification**
(tentative)



3. **Statistical classifier to filter results**



Causeway-S: Syntactic patterns + head expansion rules

Causeway-L: Lexical patterns + CRF sequence labeler

4. **Remove duplicate connectives**

(Dunietz et al., 2017)

2. Causeway-S/Causeway-L:
   two pattern-based taggers
   for causal constructions

   i. Causeway-S: Syntax-based pipeline

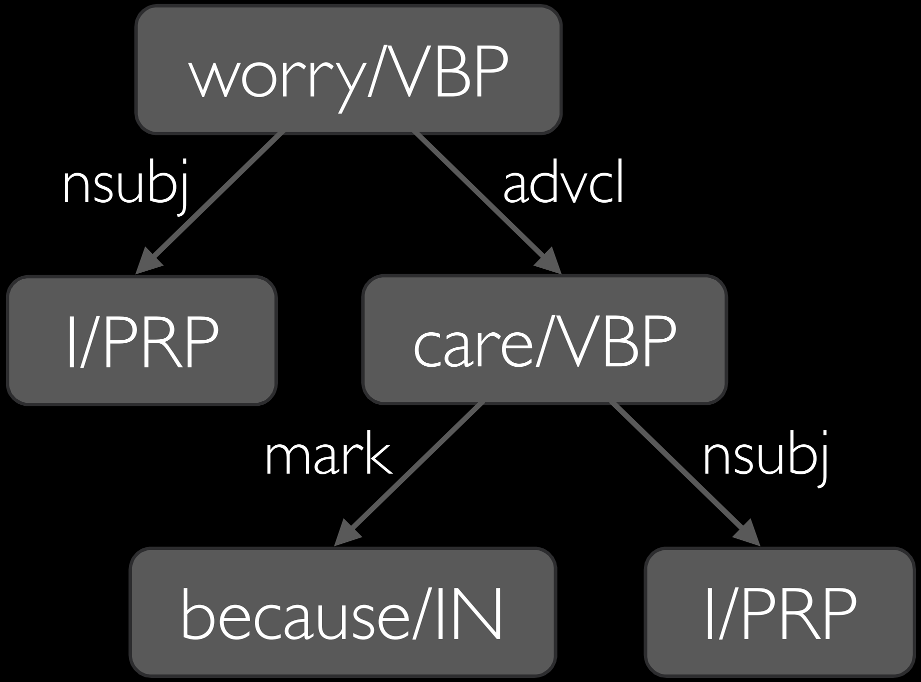
   ii. Causeway-L: Lexical pattern-based pipeline

2. Causeway-S/Causeway-L:
   two pattern-based taggers
   for causal constructions

   i. Causeway-S: Syntax-based pipeline

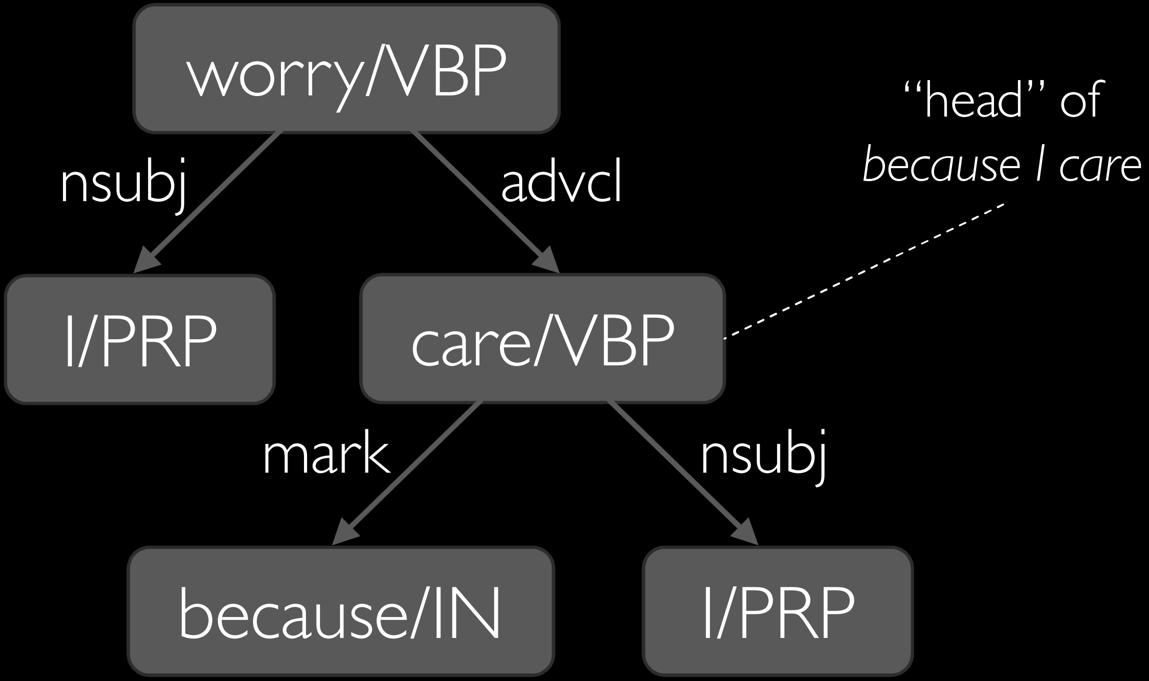
   ii. Causeway-L: Lexical pattern-based pipeline

Syntax-based connective discovery: **each construction is treated as a partially-fixed parse tree fragment.**



I worry because I care.

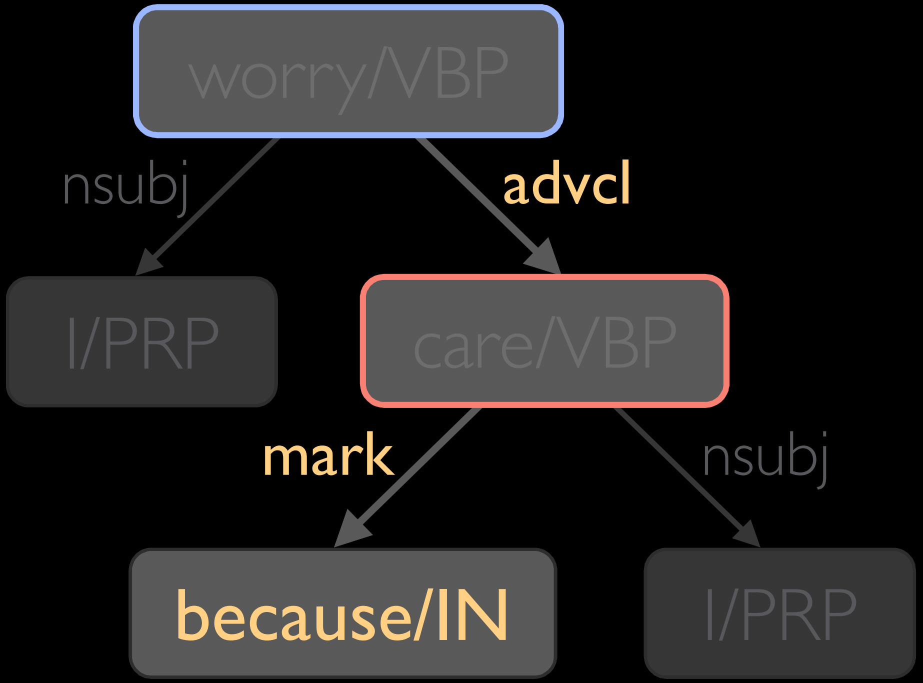
Syntax-based connective discovery: **each construction is treated as a partially-fixed parse tree fragment.**



"head" of *because I care*

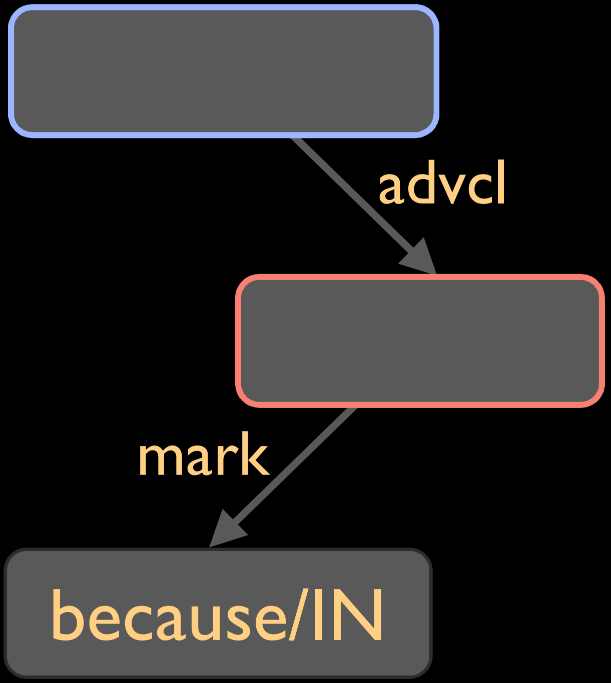
I worry because I care.

Syntax-based connective discovery: **each construction is treated as a partially-fixed parse tree fragment.**



I worry because I care.

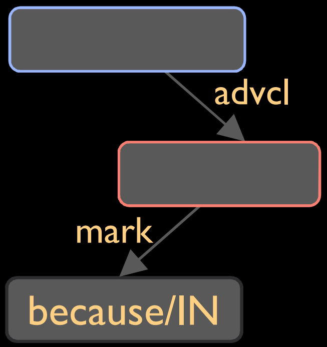
Syntax-based connective discovery: **each construction is treated as a partially-fixed parse tree fragment.**



Syntax-based connective discovery:

TRegex patterns are extracted in training, and matched at test time.

Training: I worry **because** I care.

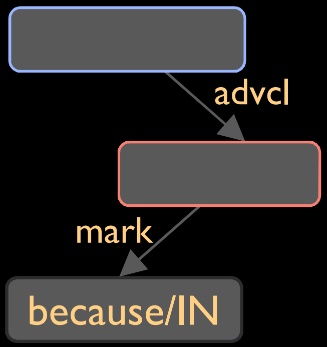


```
(/^because_[0-9]+$/
   <2 /^IN.*/ <1 mark
   > (/.*_[0-9]+/
      <1 advcl
      > (/.*_[0-9]+/)))
```

Syntax-based connective discovery:
TRegex patterns are extracted in training, and matched at test time.

Training:

I worry **because** I care.



```
(/^because_[0-9]+$/
   <2 /^IN.*/ <1 mark
   > (/.*_[0-9]+/
      <1 advcl
      > (/.*_[0-9]+/)))
```

Test:

I worry because I love you.

+

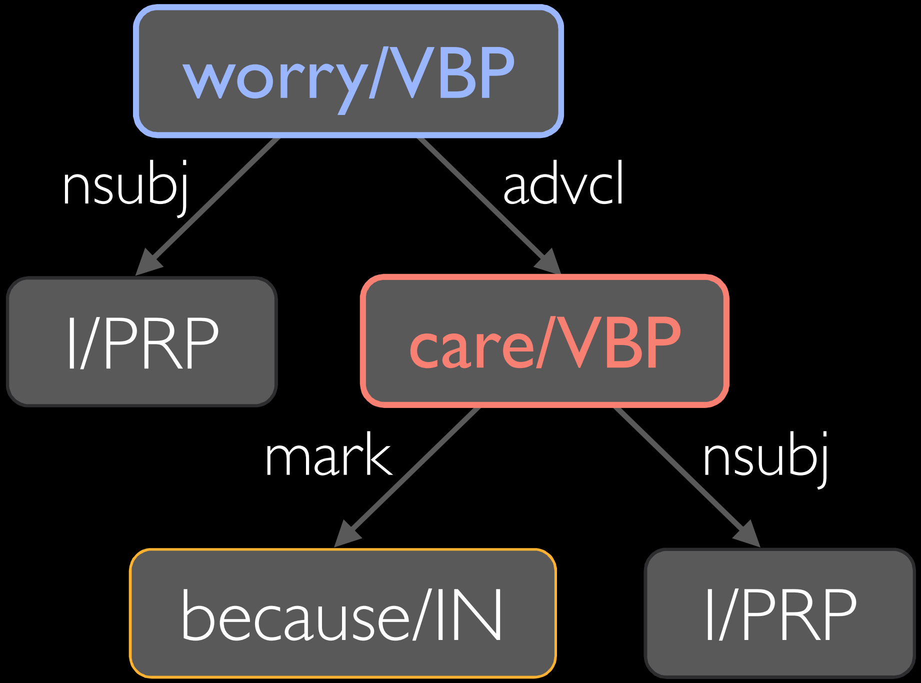
```
(/^because_[0-9]+$/
   <2 /^IN.*/ <1 mark
   > (/.*_[0-9]+/
      <1 advcl
      > (/.*_[0-9]+/)))
```

TRegex[1]

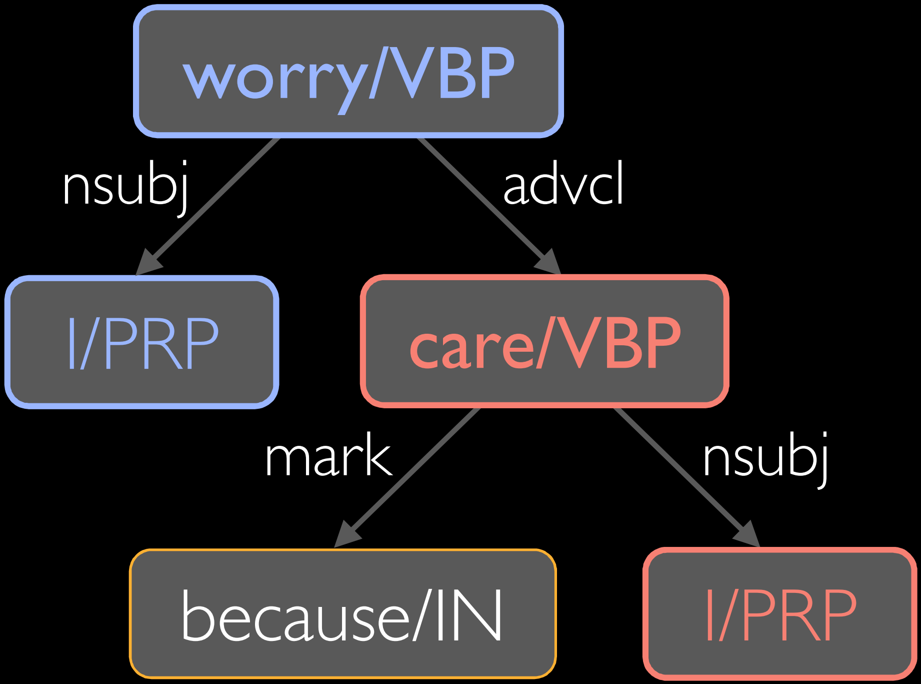
I worry **because** I love you.

[1] Levy and Andrew, 2006

Syntax-based argument ID: **Argument heads are expanded to include most dependents.**



Syntax-based argument ID:
Argument heads are expanded to include most dependents.



1. Causeway-S/Causeway-L: two simple systems for tagging causal constructions
    i. Causeway-S: Syntax-based pipeline
    ii. Causeway-L: Lexical pattern-based pipeline

1. Causeway-S/Causeway-L: two simple systems for tagging causal constructions
    1. Causeway-S: Syntax-based pipeline
    2. Causeway-L: Lexical pattern-based pipeline

Lexical pattern-based connective discovery: **constructions are matched by regular expressions over word lemmas.**

Training:

I worry **because**
I care.

```
(ˆ | )([ \ S]+ )+?(because/IN)
([ \ S]+ )+?
```

Test:

I worry because I love you.

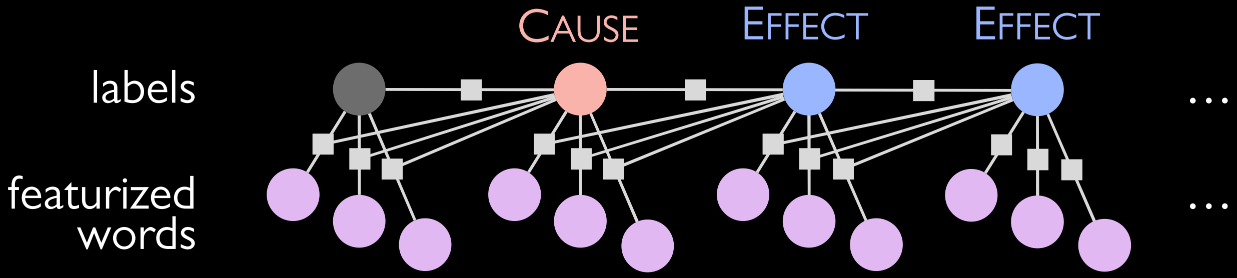
+

```
(ˆ | )([ \ S]+ )+?(because/IN)
([ \ S]+ )+?
```

regex

I worry **because**
I love you.

Lexical pattern-based argument ID: **Arguments are labeled by a conditional random field.**

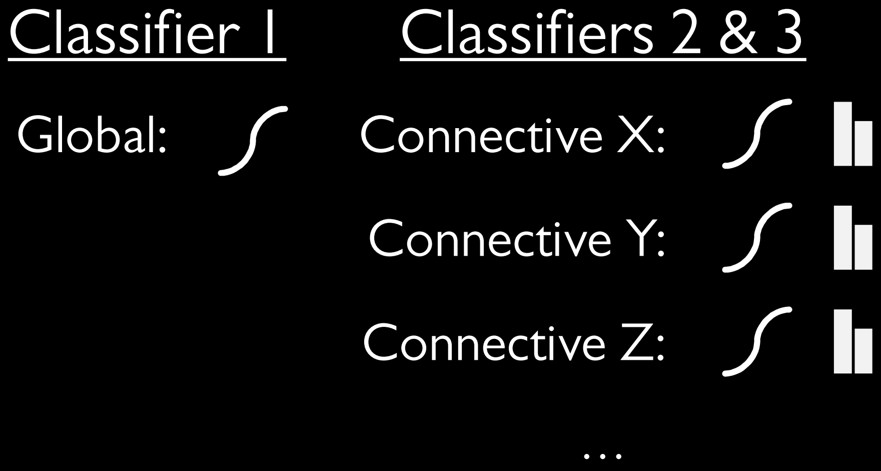


$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\prod_{t=1}^{T}\exp\left\{\sum_{k=1}^{K}\theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right\}$$

Features include information about:

- Word
- Connective
- Relationship between word & connective

Both approaches use a soft vote of three classifiers as a filter.



Classifier 1

Global:

Classifiers 2 & 3

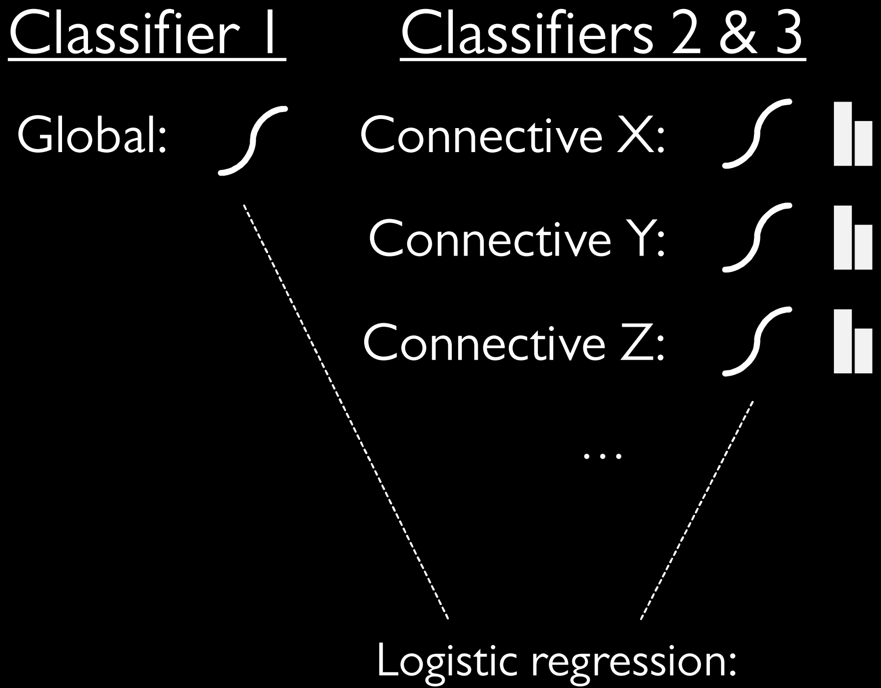
Connective X:

Connective Y:

Connective Z:

…

Both approaches use a soft vote of three classifiers as a filter.



<u>Classifier 1</u>

Global:

<u>Classifiers 2 & 3</u>

Connective X:

Connective Y:

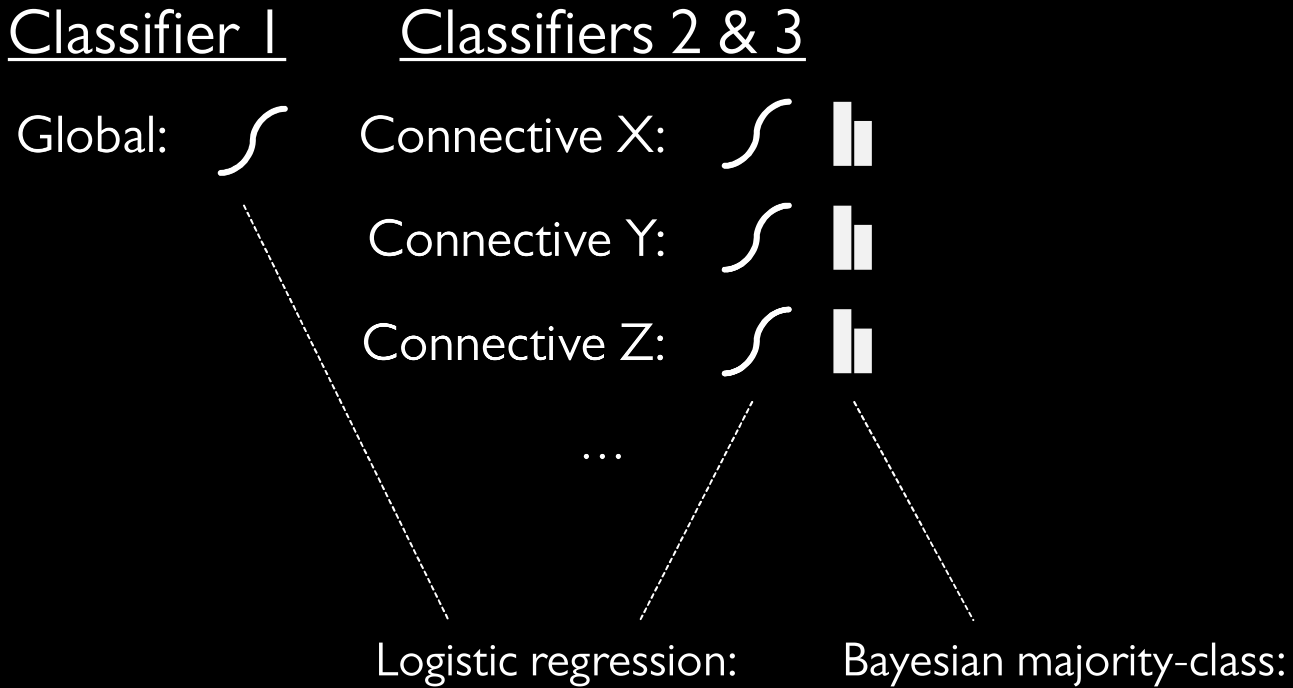
Connective Z:

…

Logistic regression:

$$p(y = \text{true}|\mathbf{x}) = \frac{1}{1 + \exp\{-\theta_0 + \boldsymbol{\theta}^\top \mathbf{x}\}}$$

Both approaches use a soft vote of three classifiers as a filter.



Classifier 1

Global:

Classifiers 2 & 3

Connective X:

Connective Y:

Connective Z:

…

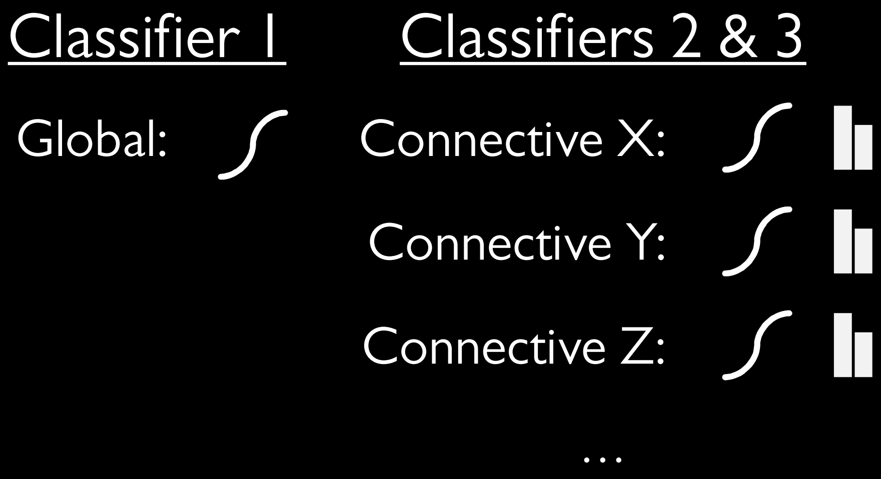
Logistic regression:

Bayesian majority-class:

$$p(y = \text{true}|\mathbf{x}) = \frac{1}{1 + \exp\{-\theta_0 + \boldsymbol{\theta}^\top \mathbf{x}\}}$$

$$p(y = \text{true}|\mathbf{x}) = \frac{\#\{\text{pattern is causal}\}}{\#\{\text{pattern appears in corpus}\}}$$

Both approaches use a soft vote of three classifiers as a filter.



Classifier 1

Global:

Classifiers 2 & 3

Connective X:

Connective Y:

Connective Z:

…

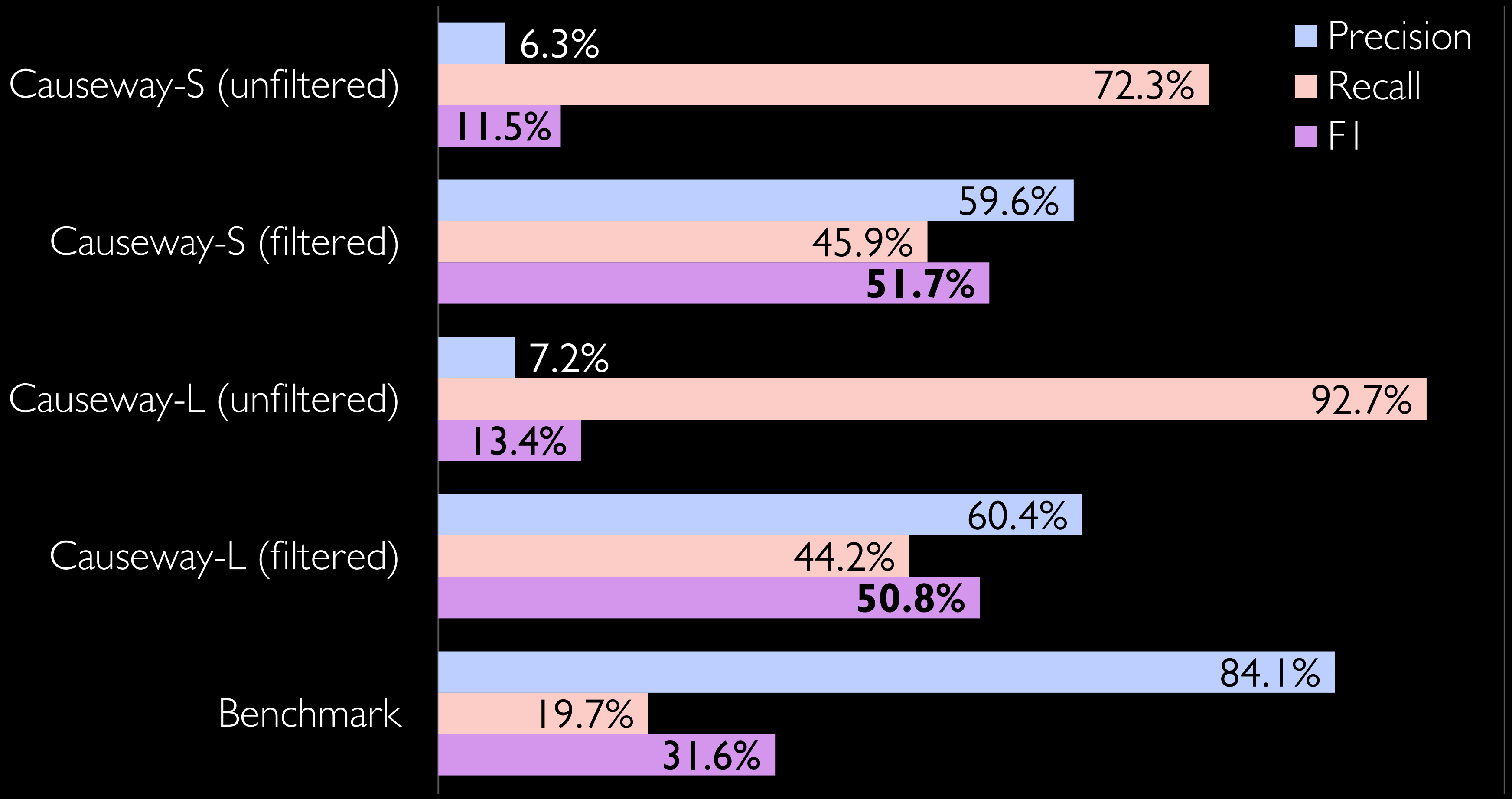
Example classifier features
(*c*=cause head, e = effect head):

- POS tags of *c* and e
- Number of words between *c* and e
- Domination relationship between *c* and e
- Matching connective pattern
- Pair of tense/aspect/modality modifier sets of *c* and e
- WordNet hypernyms

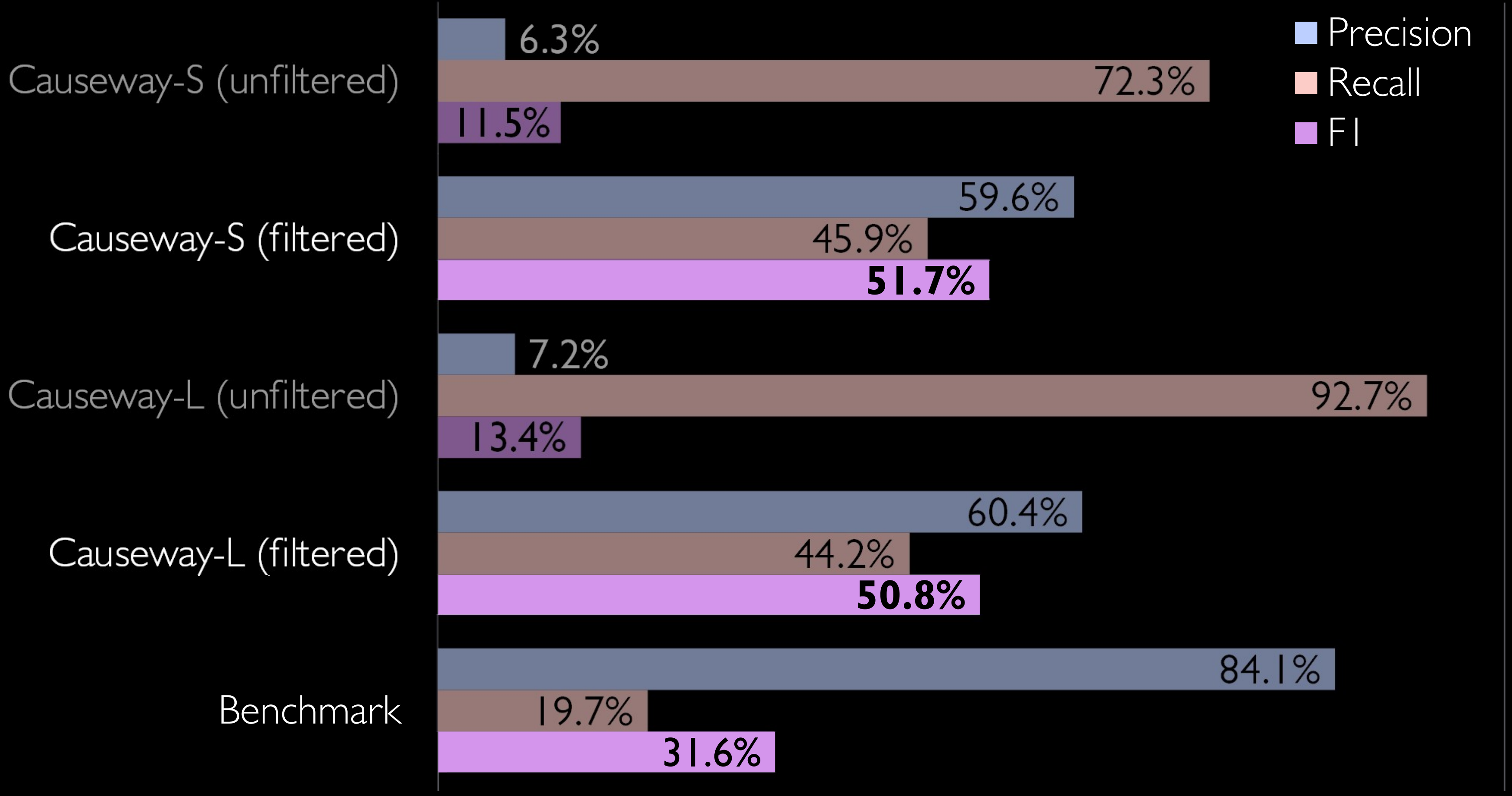
Our benchmark is a dependency path memorization heuristic.

Connective	Parse paths to possible cause/effect heads	Causal / Not causal
prevent from	nsubj, advcl	**27/ 4**
prevent from	nsubj, advmod	**0 / 8**
because of	case, case → nmod	**14 / 1**
…	…	…

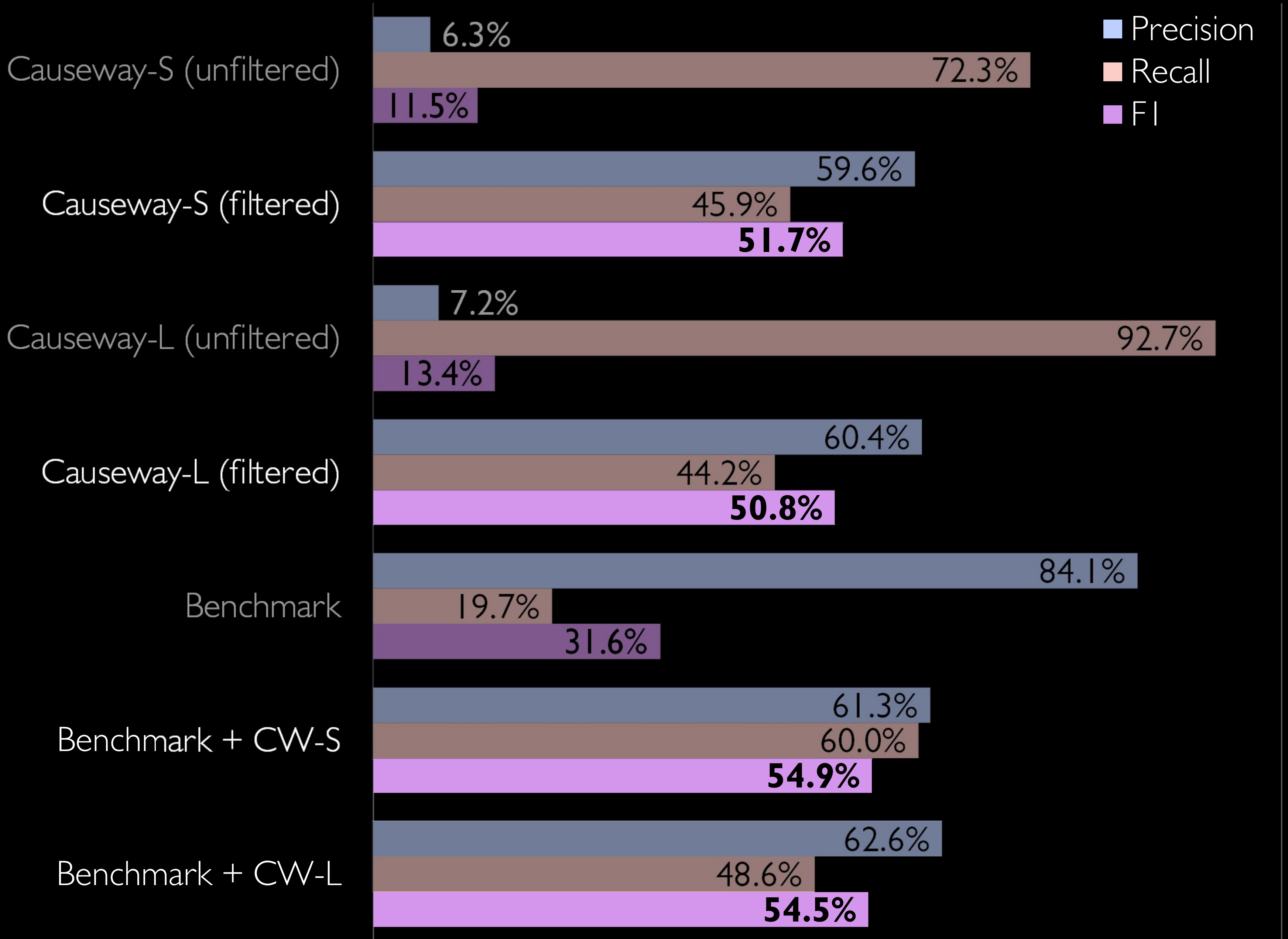
Connective discovery



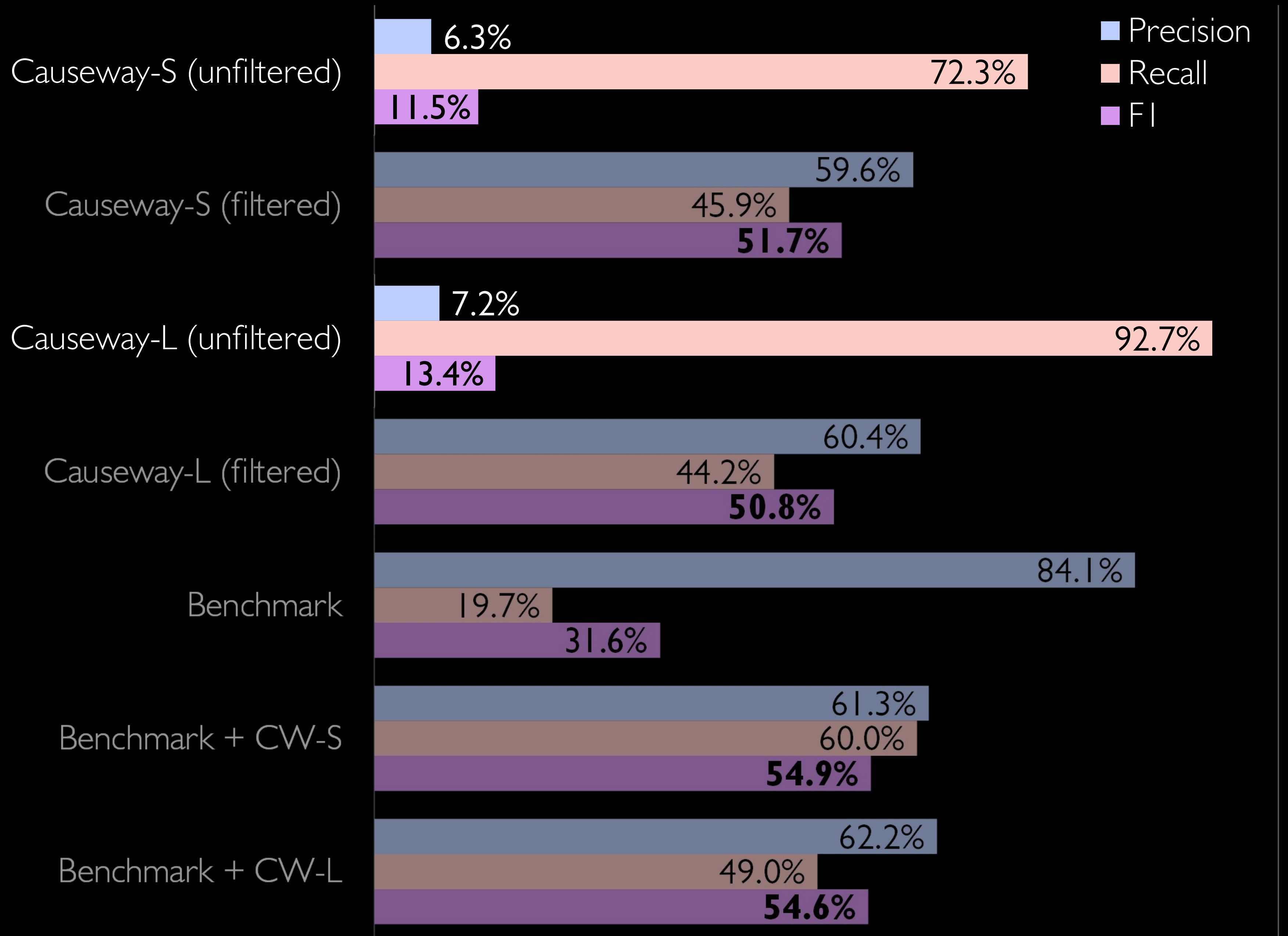
Connective discovery:
Causeway outperforms the benchmark by ~20 points.



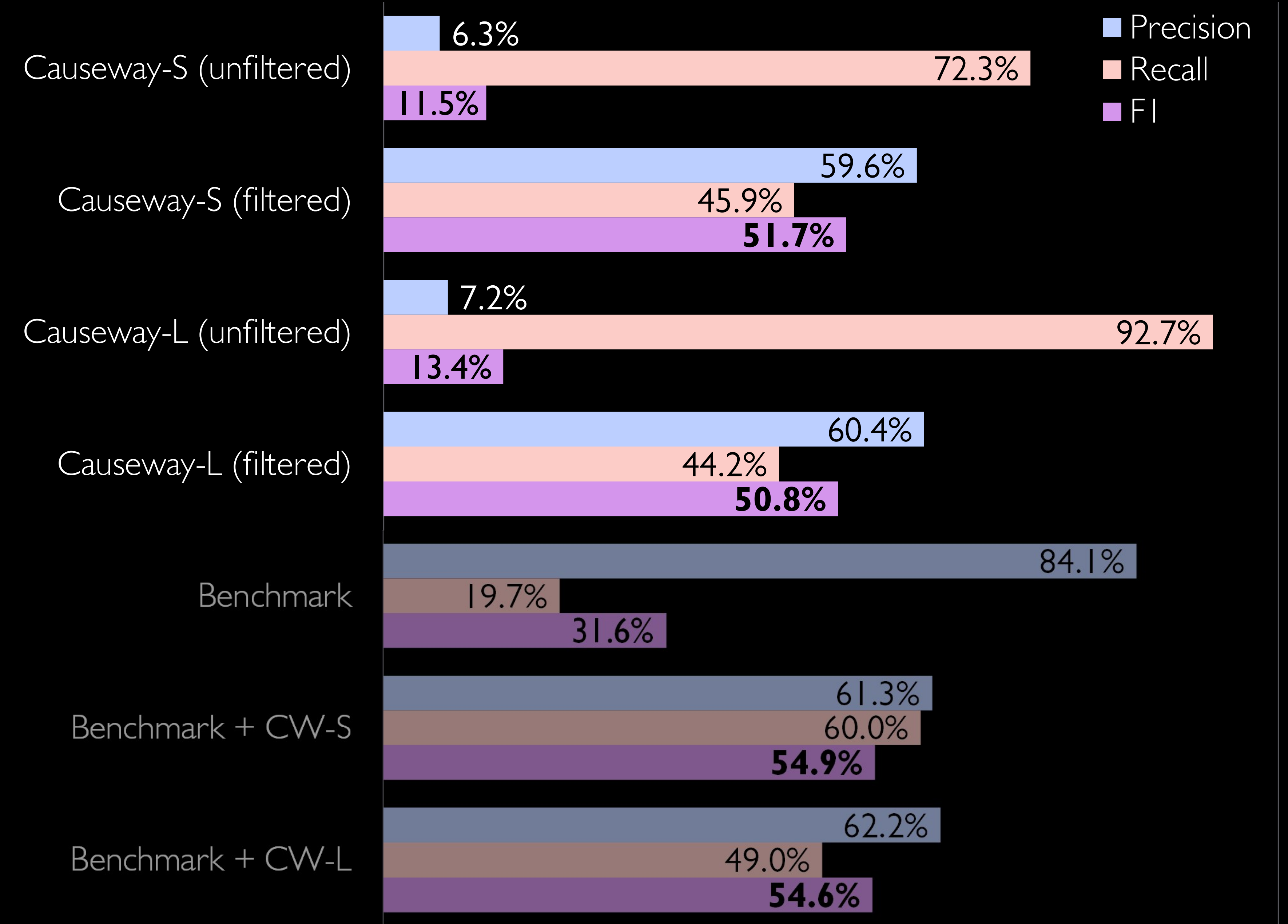
Performance improves even more when Causeway is combined with the benchmark.



The first stage gets high recall & low precision

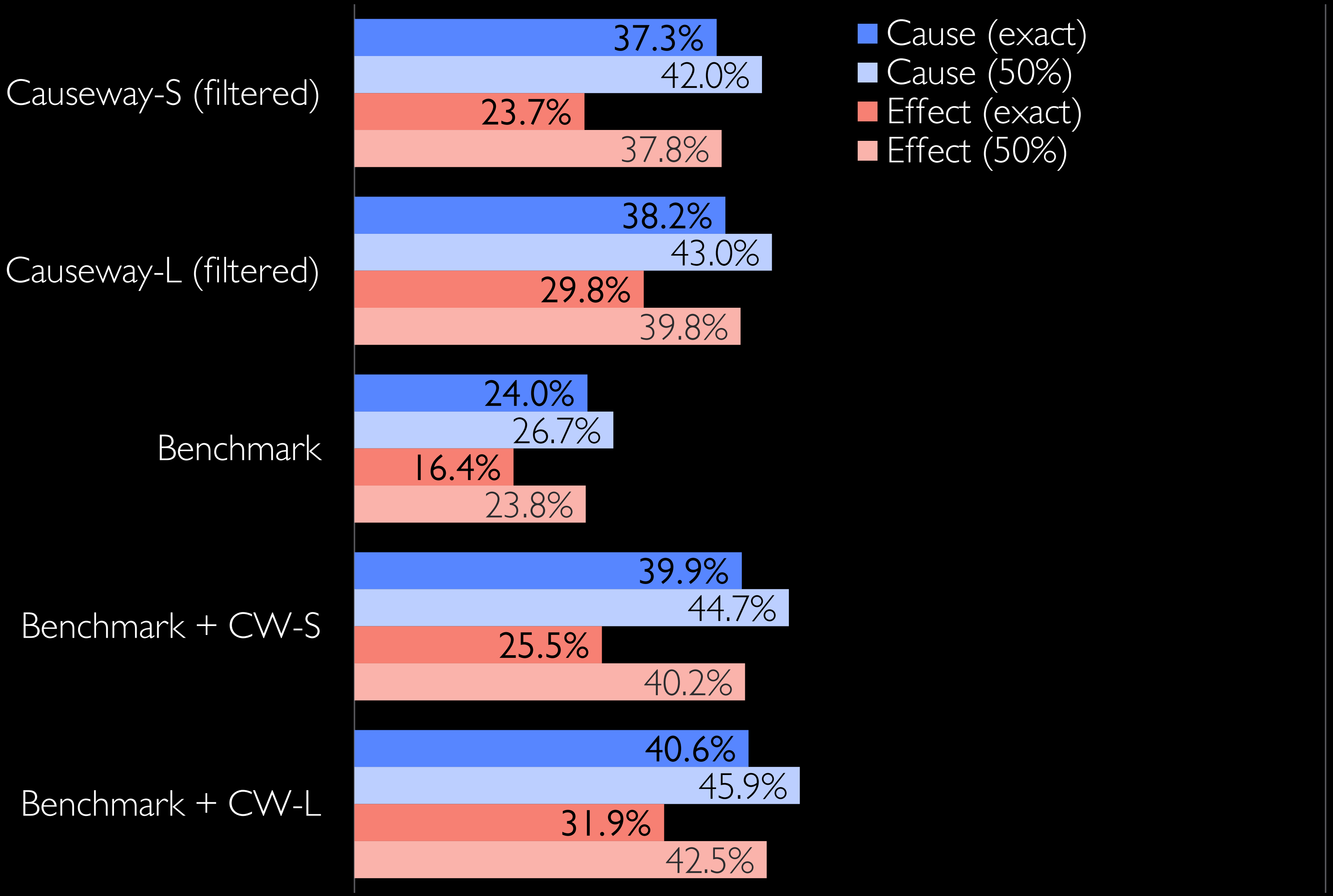


The first stage gets high recall & low precision, but the filters balance them out for a better F_1.



Argument identification is passable given connective discovery, though effects are harder than causes.

Argument identification is passable given connective discovery, though effects are harder than causes.



Today's talk:

1. The BECAUSE annotation scheme & corpus of causal language

2. Causeway-L/Causeway-S: two pattern-based taggers for causal constructions

3. DeepCx: a neural, transition-based tagger for causal constructions

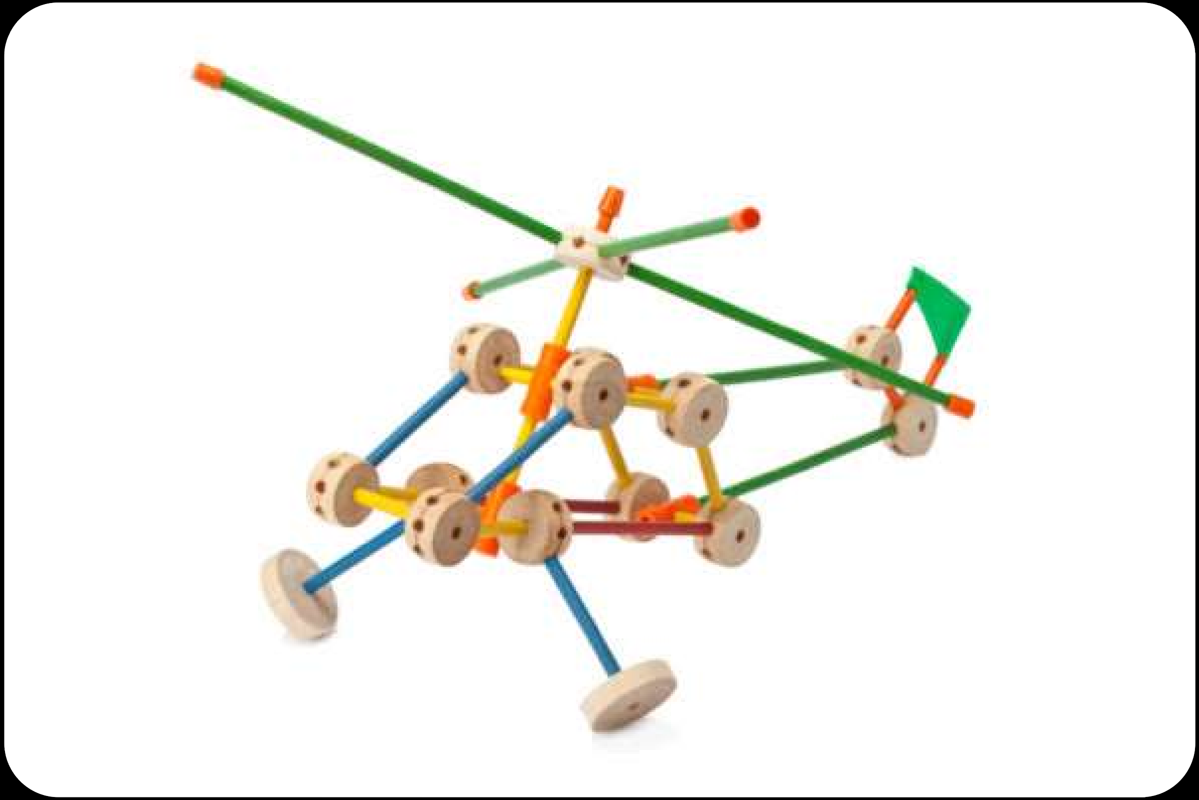
Today's talk:

1. The BECAUSE annotation scheme & corpus of causal language

2. Causeway-L/Causeway-S: two pattern-based taggers for causal constructions

3. DeepCx: a neural, transition-based tagger for causal constructions

Transition-based tagging builds a complex output structure using a sequence of simple operations.

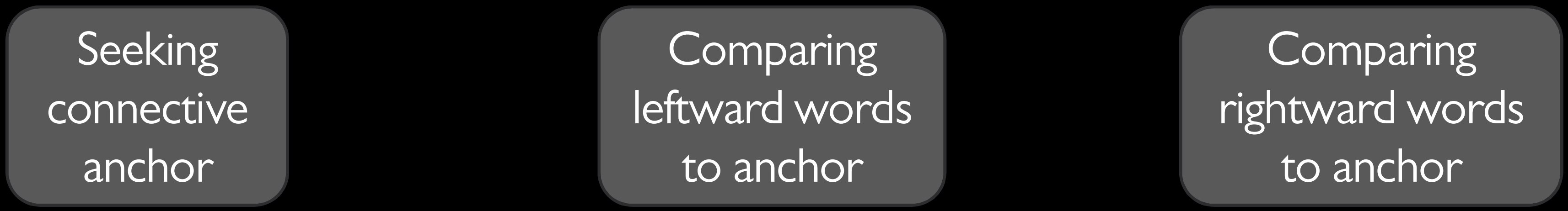


The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)



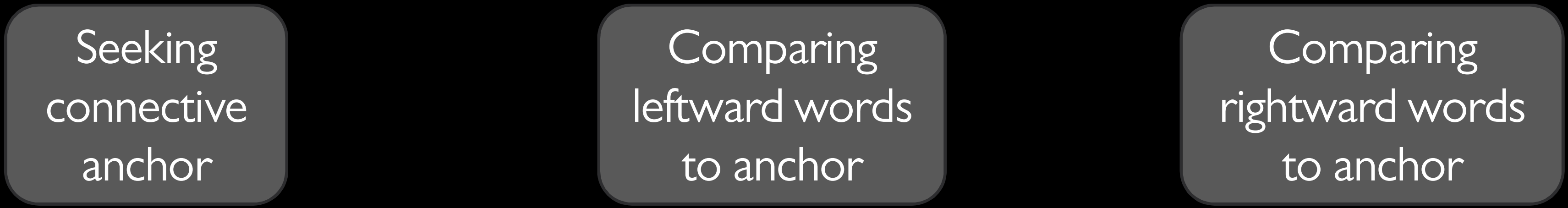
Seeking connective anchor

Comparing leftward words to anchor

Comparing rightward words to anchor

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)



Seeking connective anchor

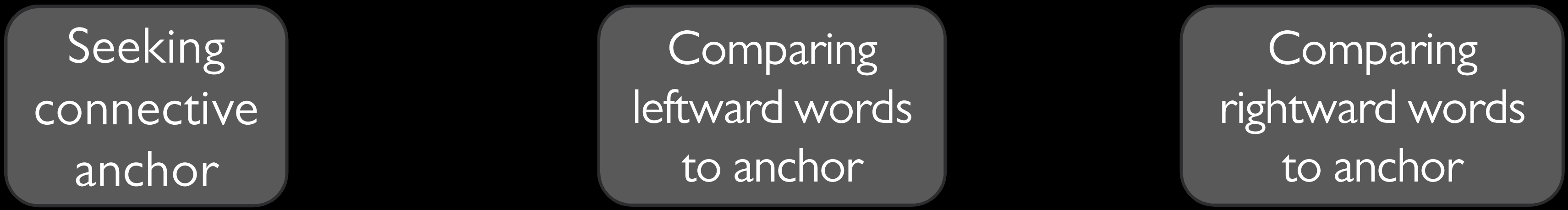
Comparing leftward words to anchor

Comparing rightward words to anchor

Well , they moved because of the schools

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)



Seeking connective anchor

Comparing leftward words to anchor

Comparing rightward words to anchor

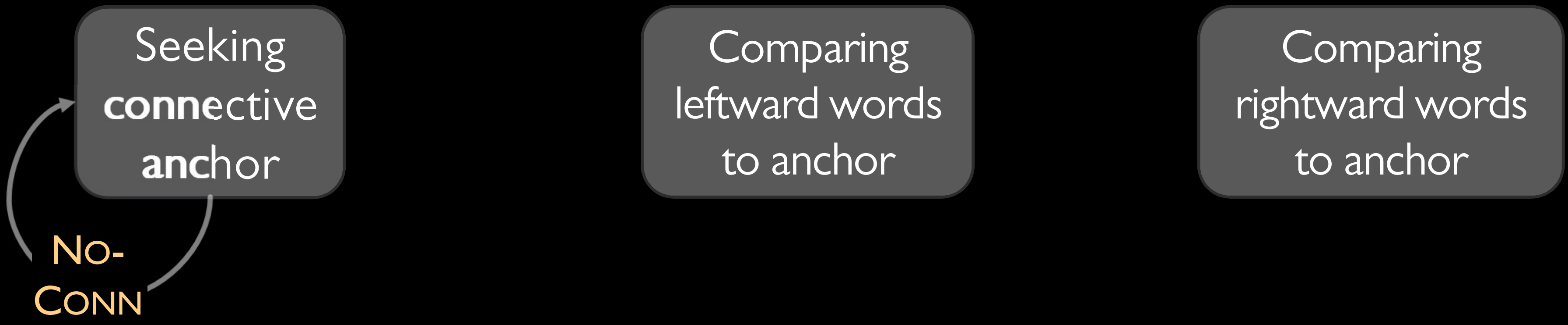
Tagger state

Well , they moved because of the schools

(Possible) connective anchor

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)



Seeking connective anchor

Comparing leftward words to anchor

Comparing rightward words to anchor

No-Conn

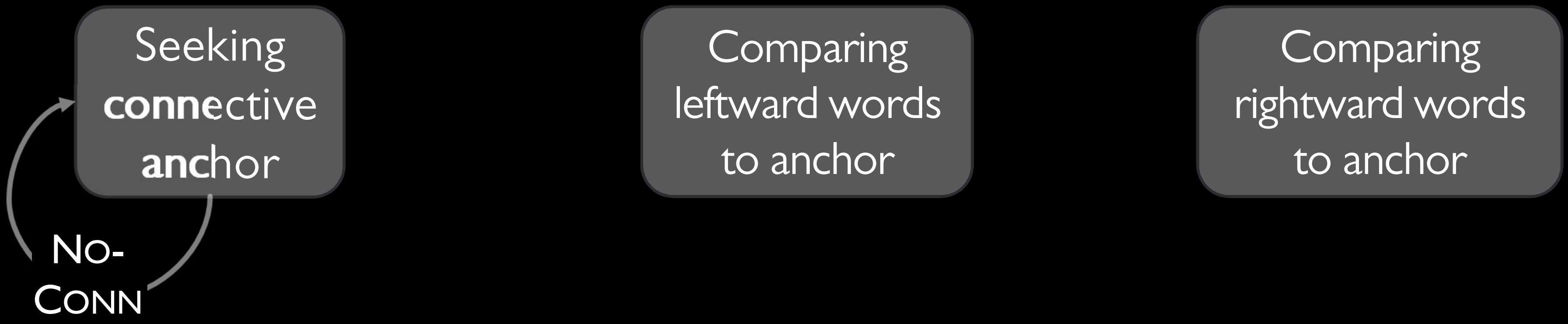
Tagger state

Well , they moved because of the schools

(Possible) connective anchor

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


Seeking connective anchor
No-Conn
Comparing leftward words to anchor
Comparing rightward words to anchor

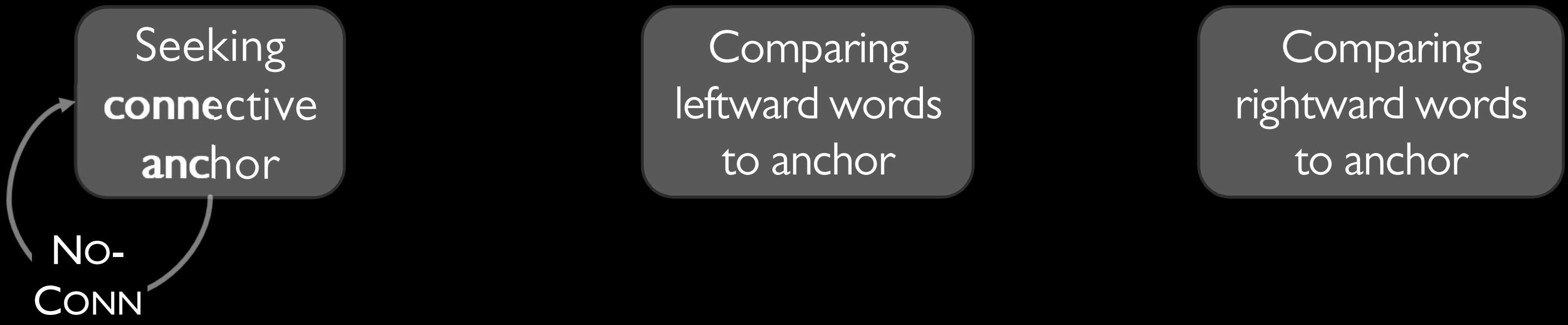
Tagger state

Well , they moved because of the schools

(Possible) connective anchor

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


Seeking connective anchor
Comparing leftward words to anchor
Comparing rightward words to anchor
No-Conn

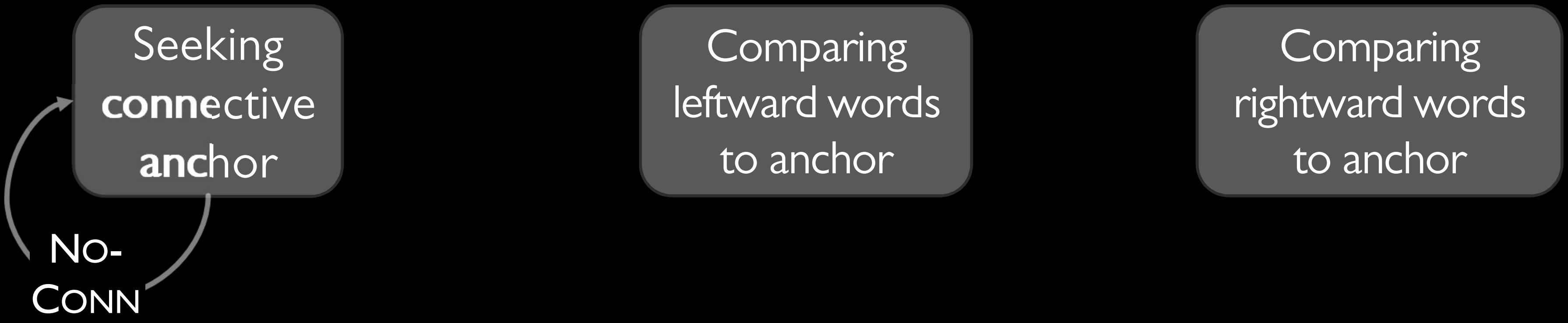
Tagger state

Well , they moved because of the schools

(Possible) connective anchor

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


Seeking connective anchor
Comparing leftward words to anchor
Comparing rightward words to anchor
No-Conn

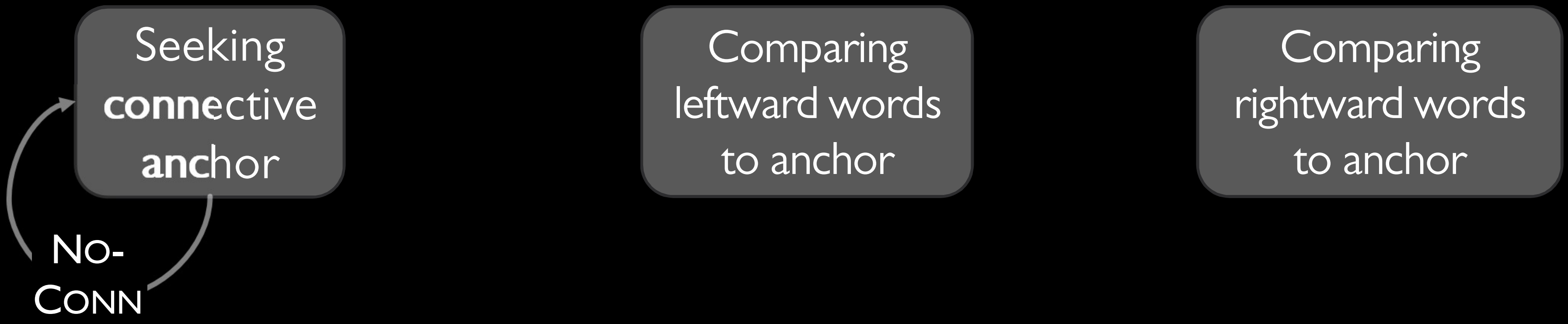
Tagger state

Well , they moved because of the schools

(Possible) connective anchor

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


Seeking connective anchor
Comparing leftward words to anchor
Comparing rightward words to anchor
No-Conn

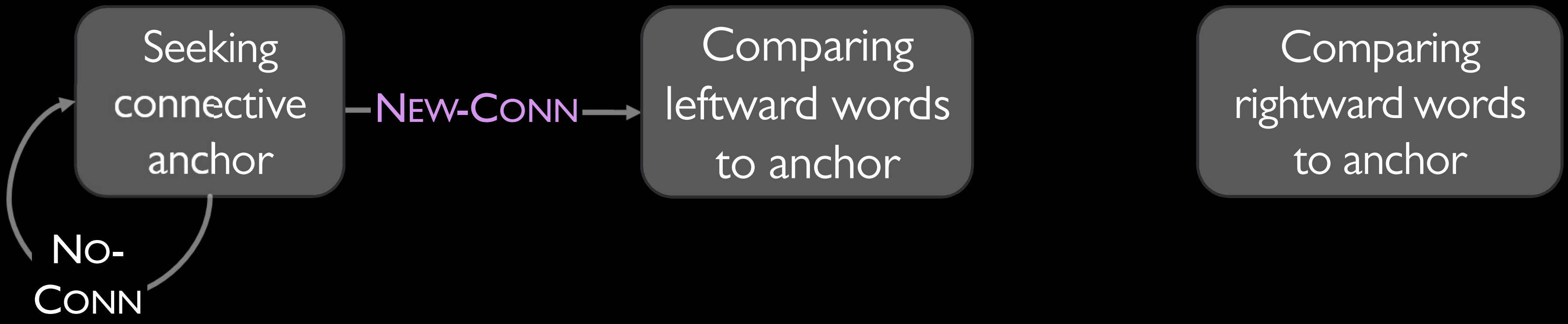
Tagger state

Well , they moved because of the schools

(Possible) connective anchor

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


Seeking connective anchor
New-Conn
Comparing leftward words to anchor
Comparing rightward words to anchor
No-Conn

Tagger state

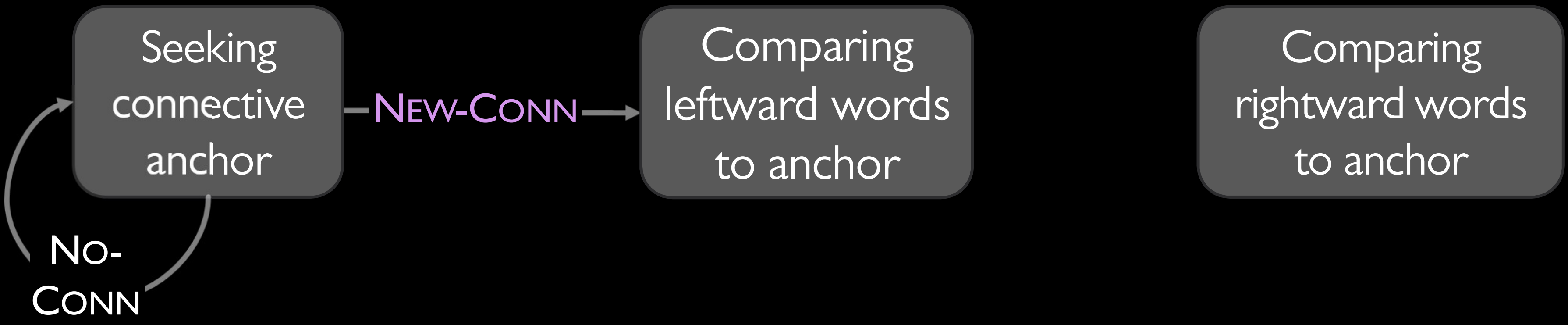
Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

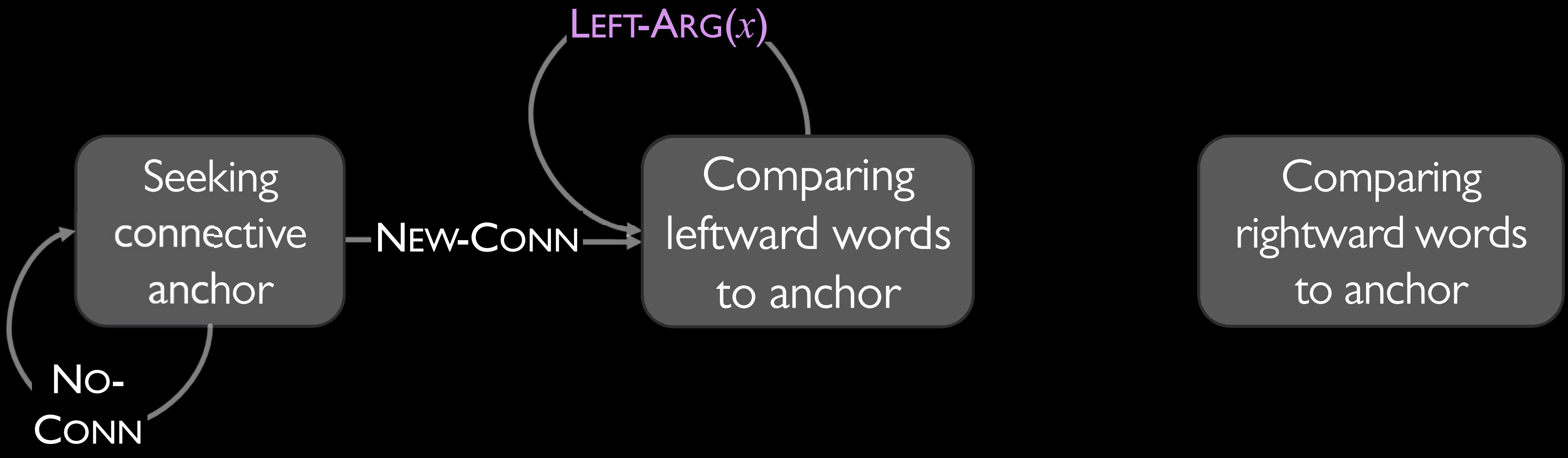
Partially-constructed instance

because

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


LEFT-ARG(x)
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

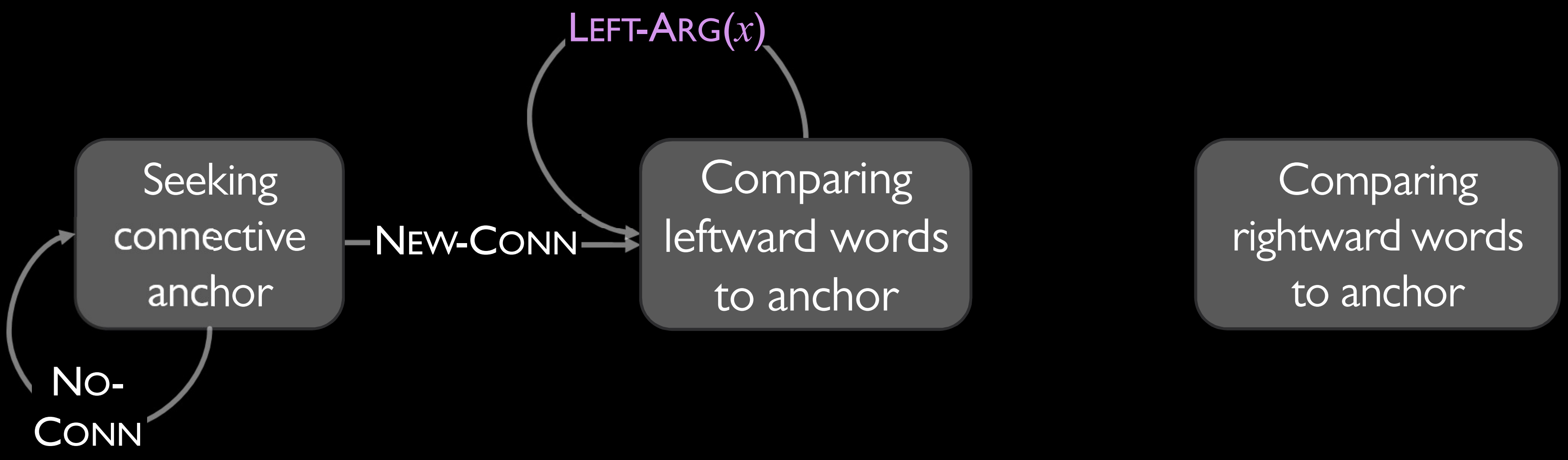
Partially-constructed instance

because

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


LEFT-ARG(x)
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

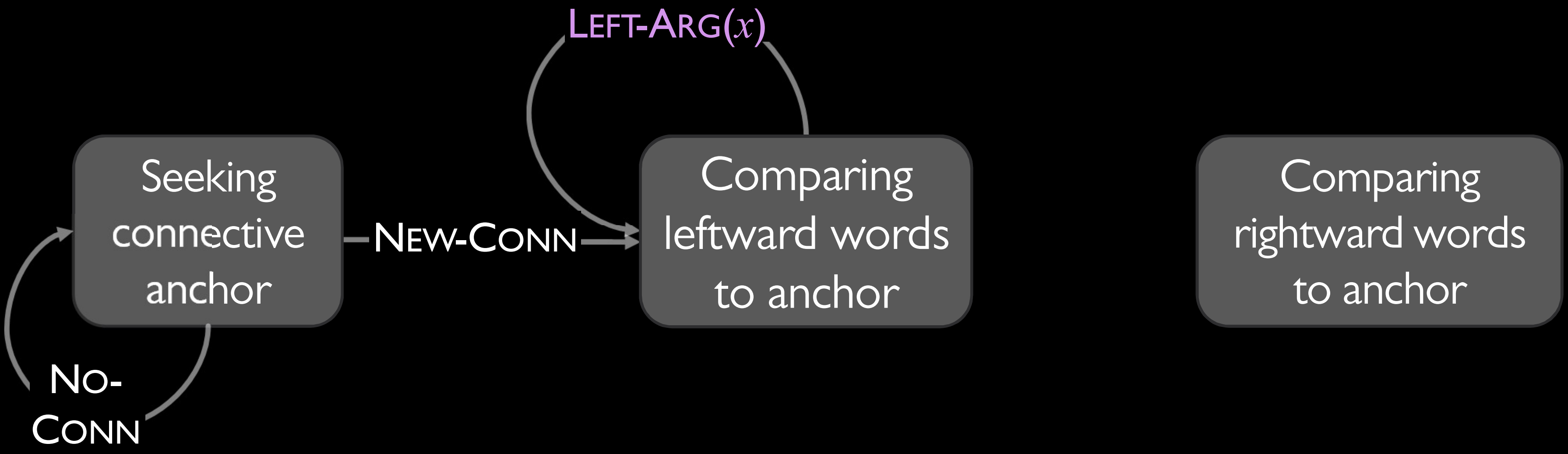
Partially-constructed instance

moved because

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


LEFT-ARG(x)
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

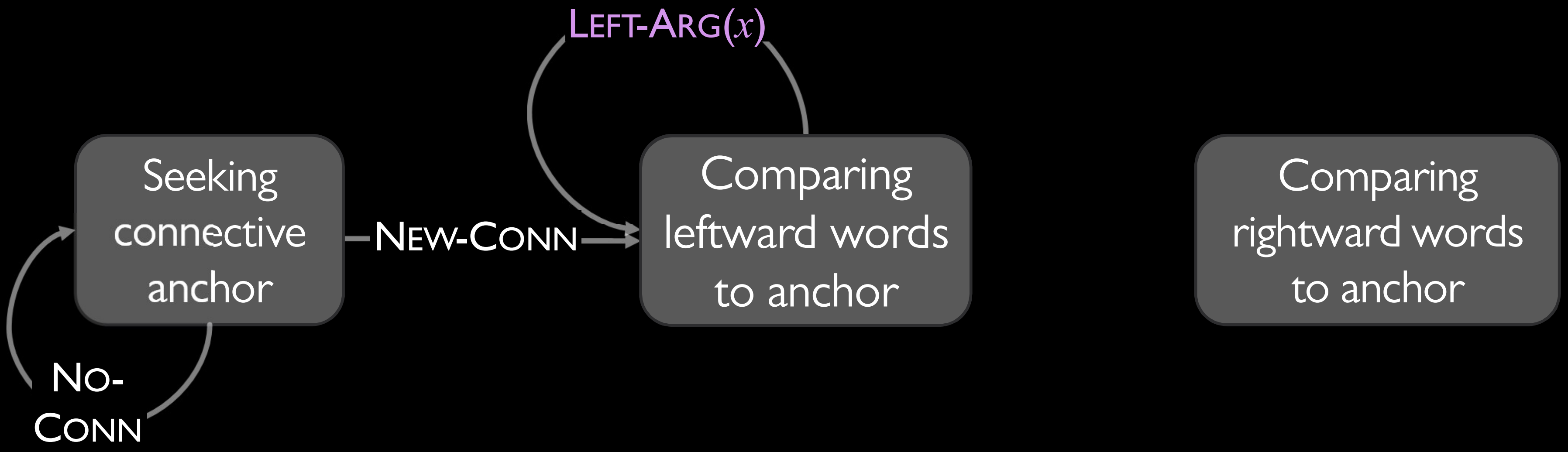
Partially-constructed instance

moved because

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


LEFT-ARG(x)
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

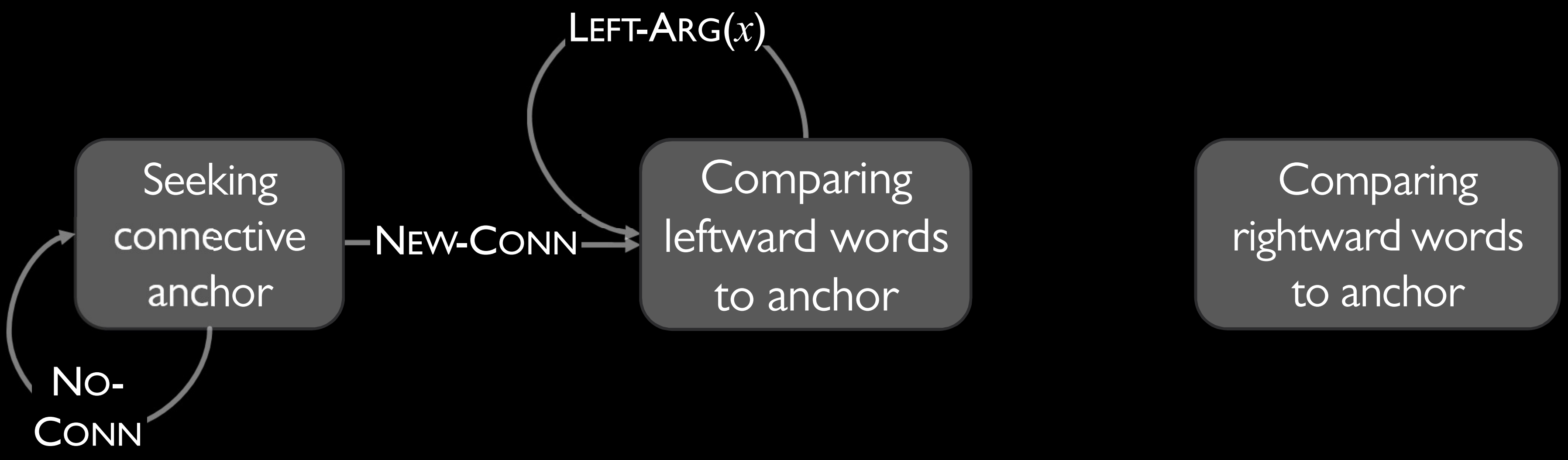
Partially-constructed instance

they moved because

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)



LEFT-ARG(x)

Seeking connective anchor

NEW-CONN

Comparing leftward words to anchor

Comparing rightward words to anchor

NO-CONN

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

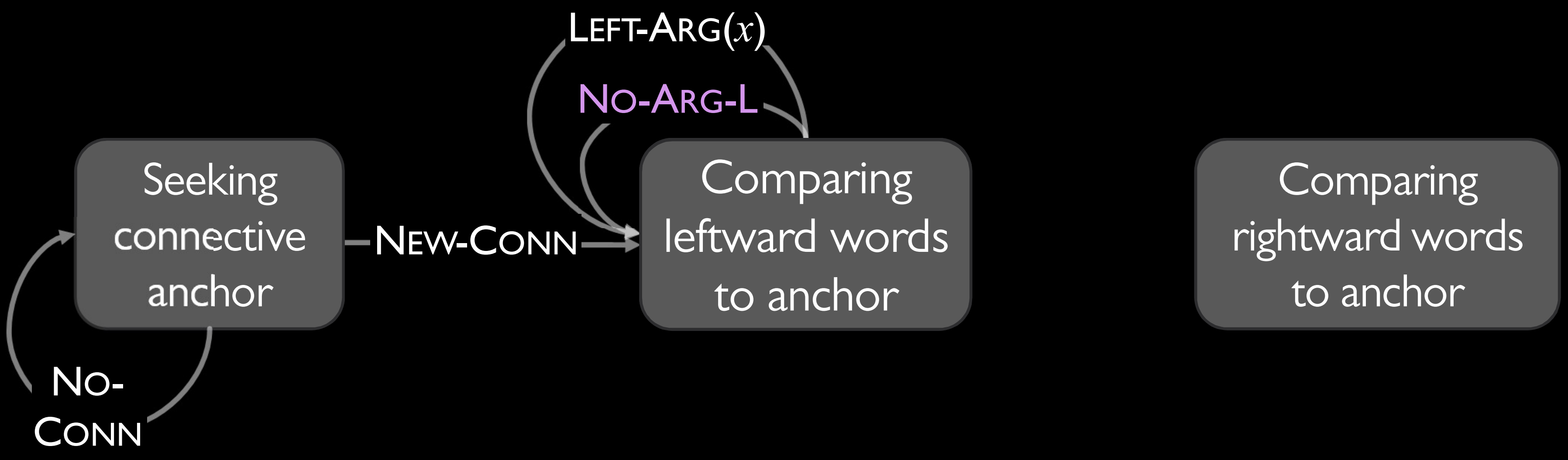
Partially-constructed instance

they moved because

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


LEFT-ARG(x)
NO-ARG-L
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

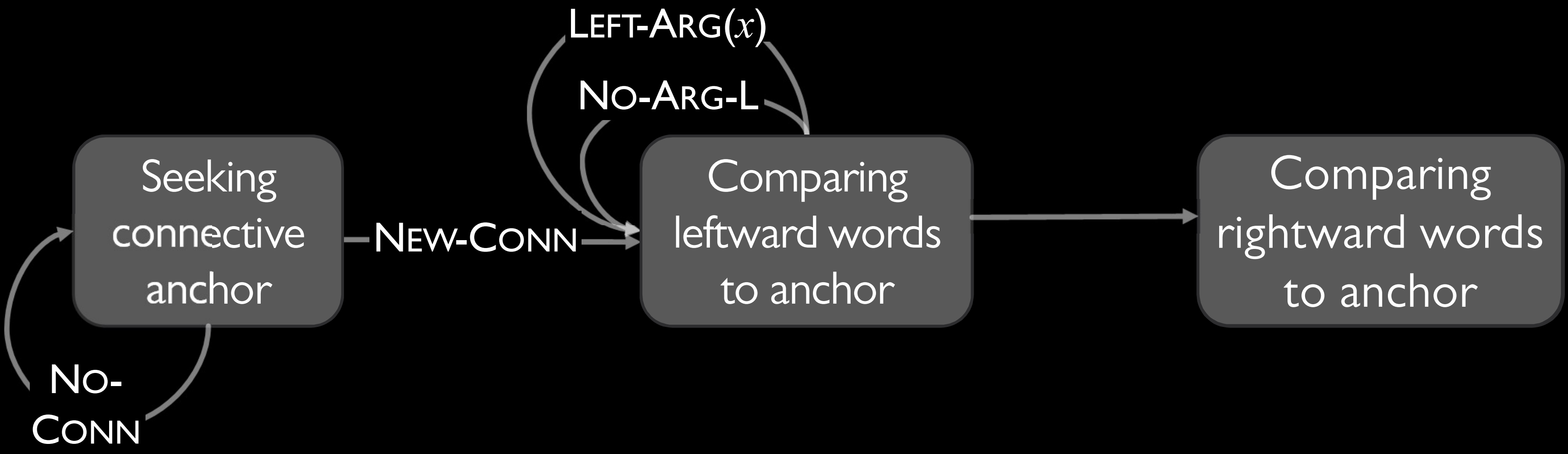
Partially-constructed instance

they moved because

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


LEFT-ARG(x)
NO-ARG-L
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

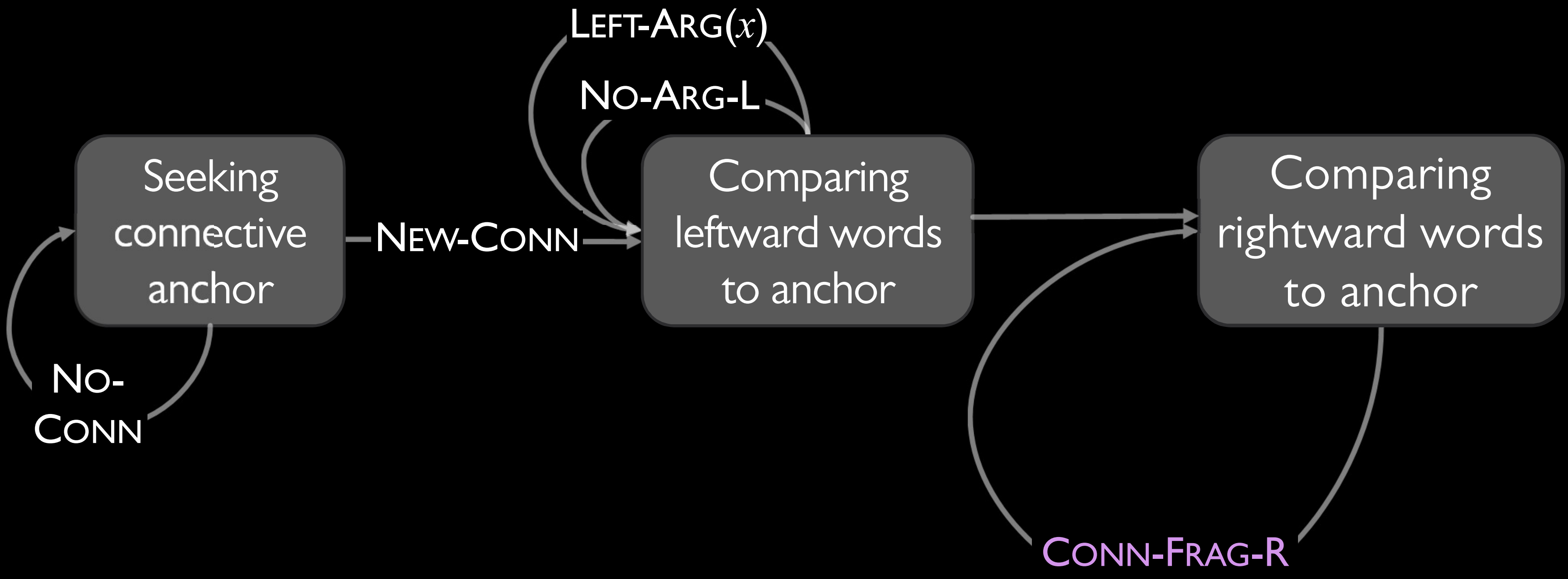
Partially-constructed instance

they moved because

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)



LEFT-ARG(x)
NO-ARG-L
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN
CONN-FRAG-R

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

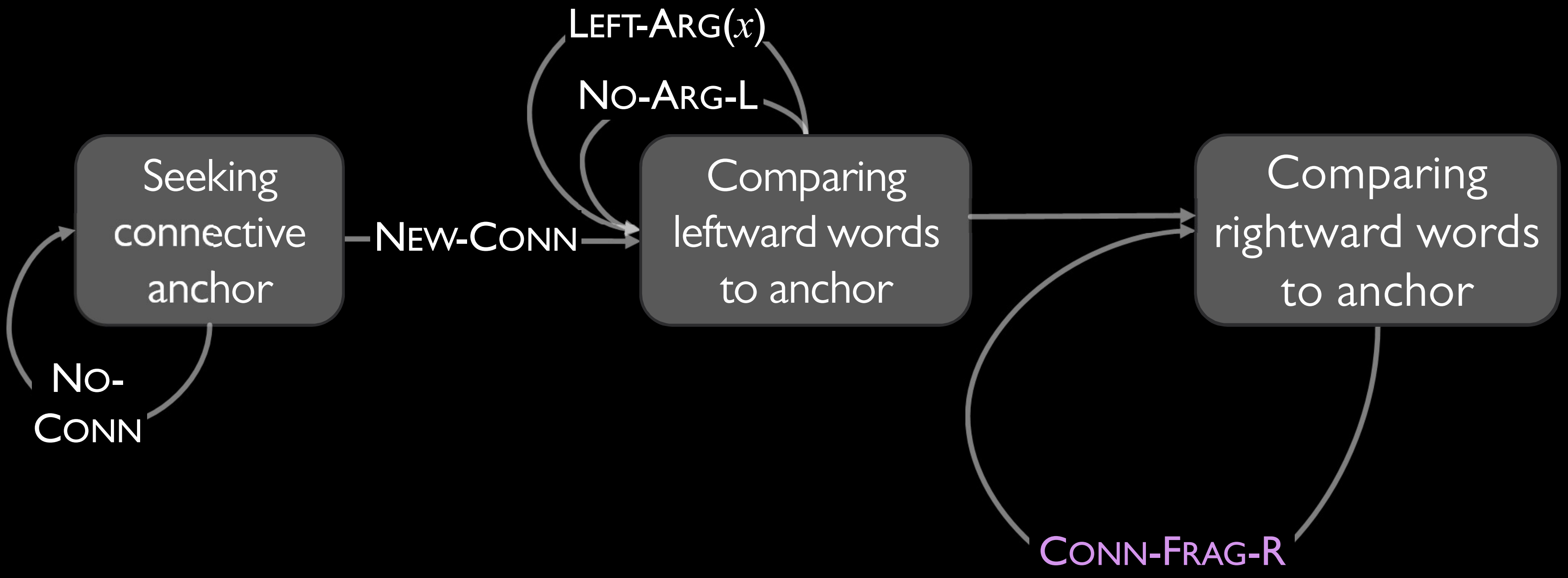
Partially-constructed instance

they moved because

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


LEFT-ARG(x)
NO-ARG-L
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN
CONN-FRAG-R

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

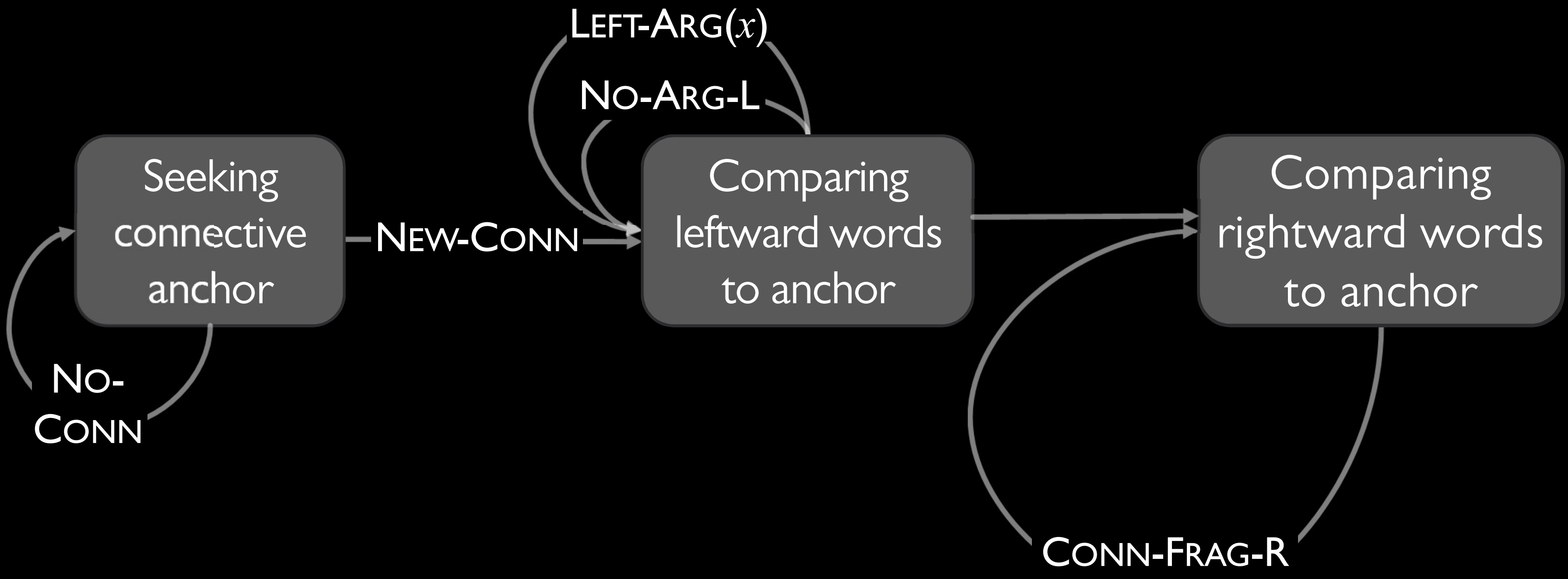
Partially-constructed instance

they moved because of

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


Left-Arg(x)
No-Arg-L
Seeking connective anchor
New-Conn
Comparing leftward words to anchor
Comparing rightward words to anchor
No-Conn
Conn-Frag-R

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

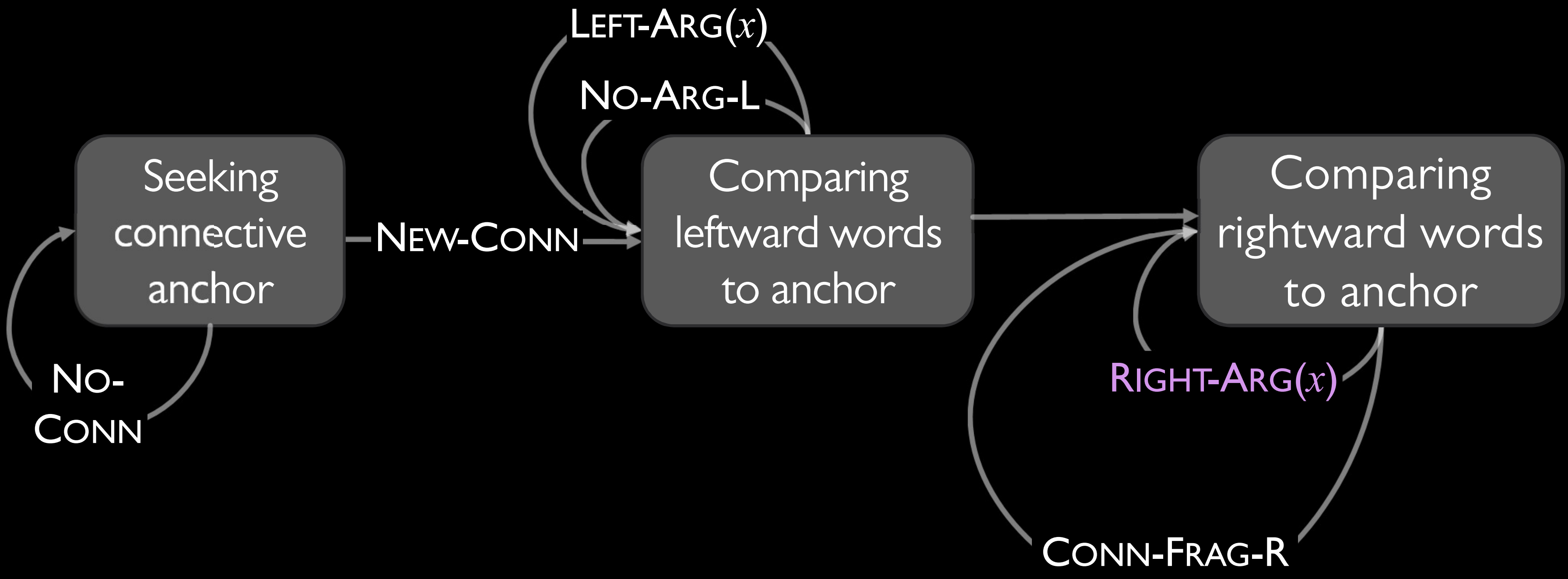
Partially-constructed instance

they moved because of

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)



Seeking connective anchor

No-Conn

New-Conn

Comparing leftward words to anchor

Left-Arg(x)

No-Arg-L

Comparing rightward words to anchor

Right-Arg(x)

Conn-Frag-R

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

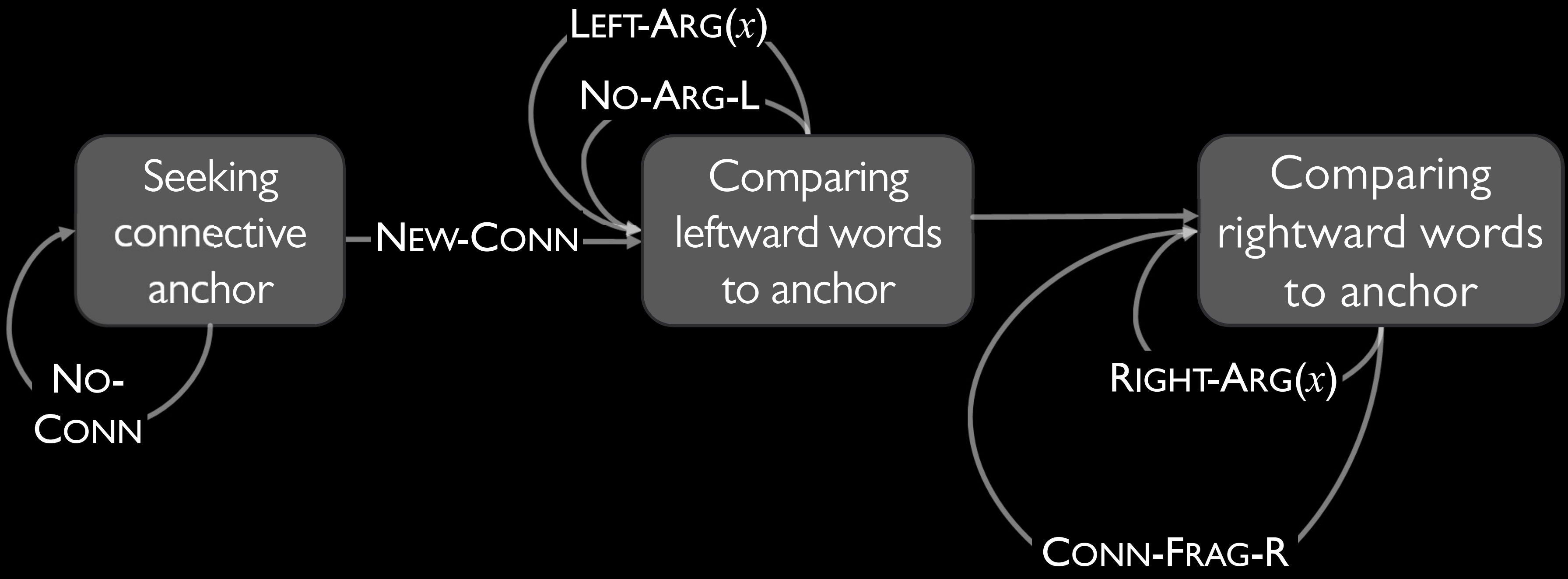
Next possible argument/fragment

Partially-constructed instance

they moved	because of	the
Effect	Connective	Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


LEFT-ARG(x)
NO-ARG-L
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN
RIGHT-ARG(x)
CONN-FRAG-R

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

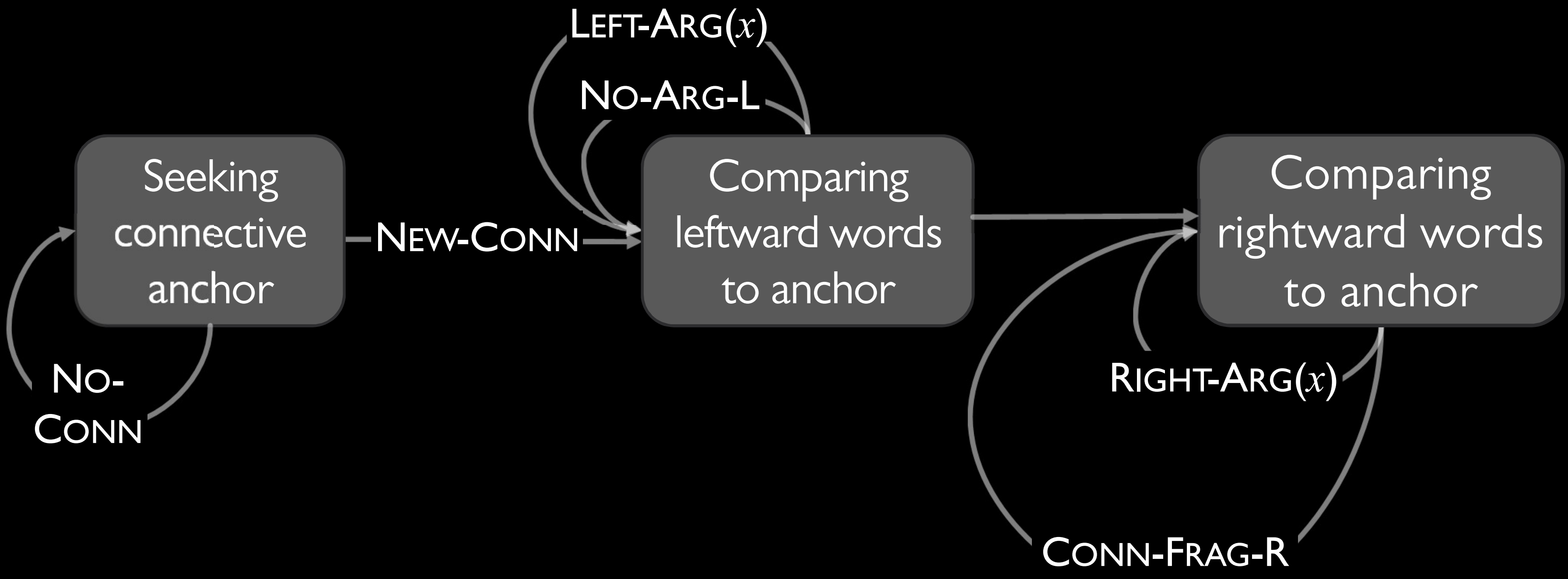
Partially-constructed instance

they moved because of the

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)



Seeking connective anchor

Comparing leftward words to anchor

Comparing rightward words to anchor

New-Conn

No-Conn

Left-Arg(x)

No-Arg-L

Right-Arg(x)

Conn-Frag-R

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

Next possible argument/fragment

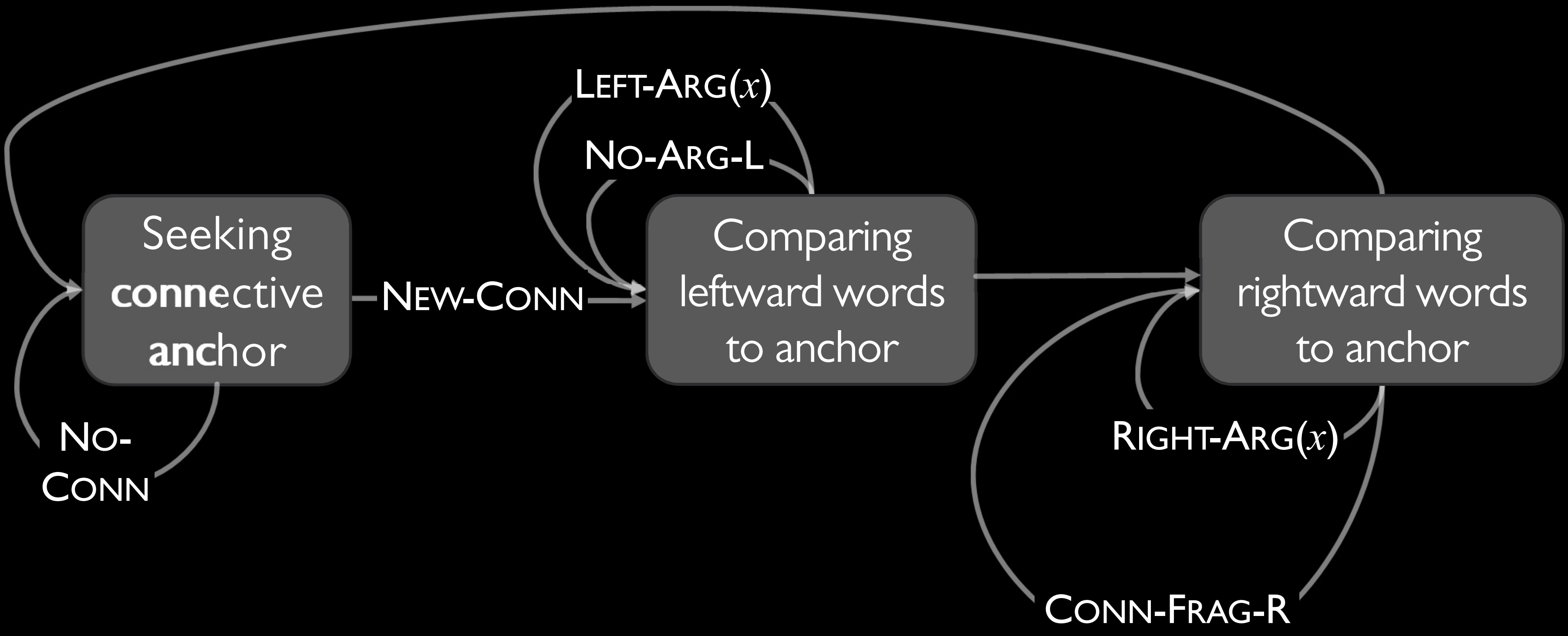
Partially-constructed instance

they moved because of the schools

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


LEFT-ARG(x)
NO-ARG-L
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN
RIGHT-ARG(x)
CONN-FRAG-R

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

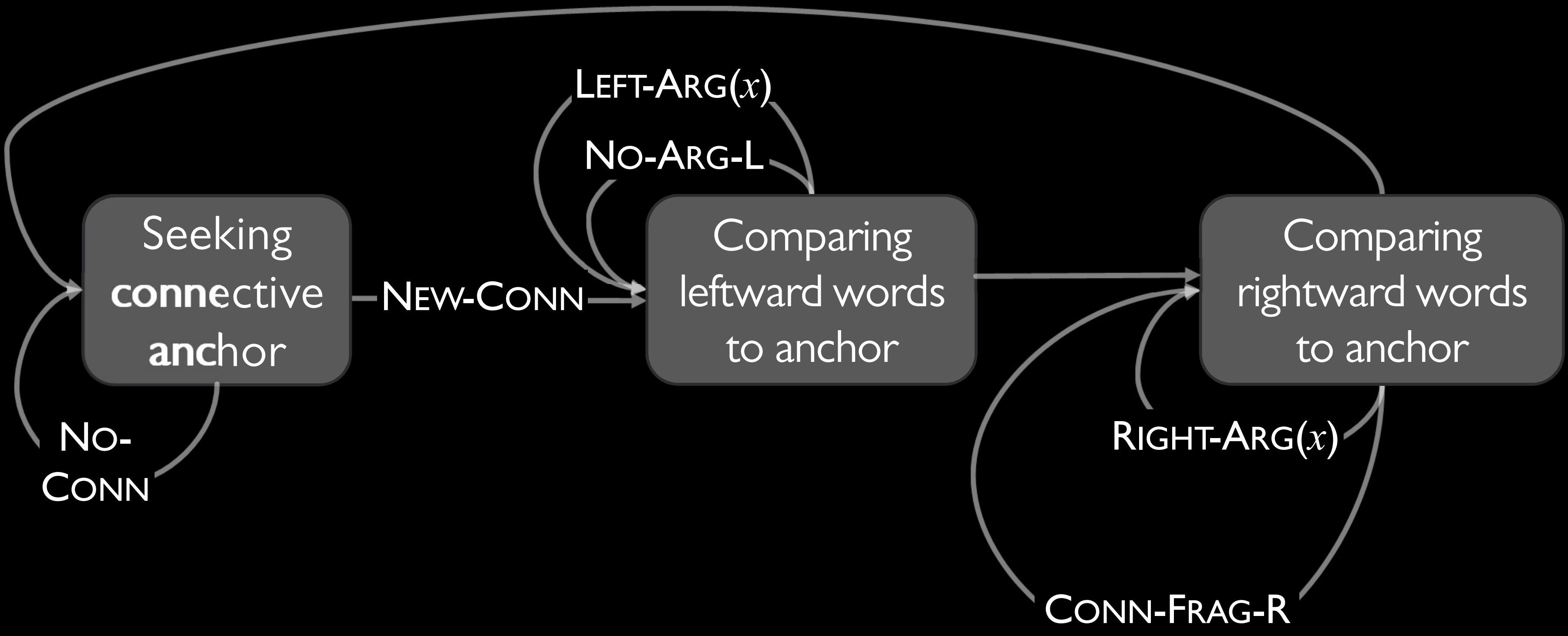
Next possible argument/fragment

Partially-constructed instance

they moved	because of	the schools
Effect	Connective	Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


Seeking connective anchor
Comparing leftward words to anchor
Comparing rightward words to anchor
NEW-CONN
NO-CONN
LEFT-ARG(x)
NO-ARG-L
RIGHT-ARG(x)
CONN-FRAG-R

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

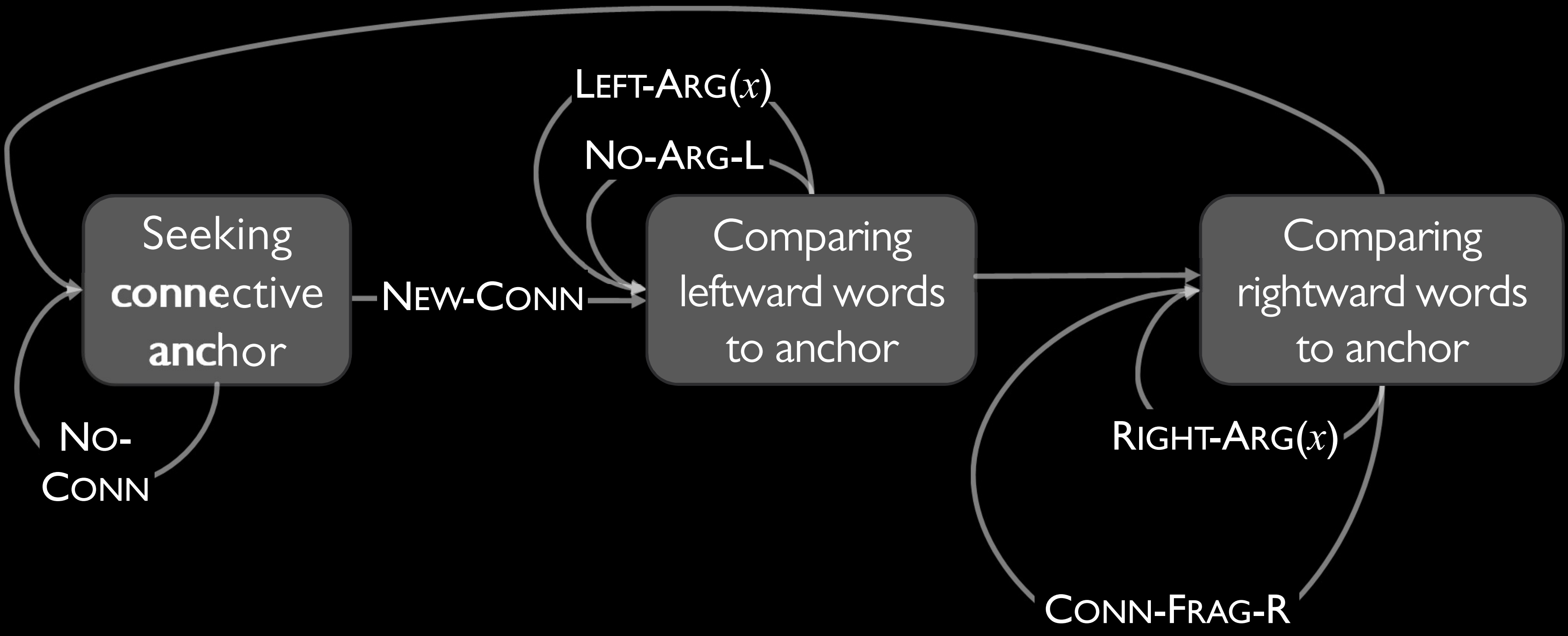
Next possible argument/fragment

Partially-constructed instance

they moved	because of	the schools
Effect	Connective	Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


LEFT-ARG(x)
NO-ARG-L
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN
RIGHT-ARG(x)
CONN-FRAG-R

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

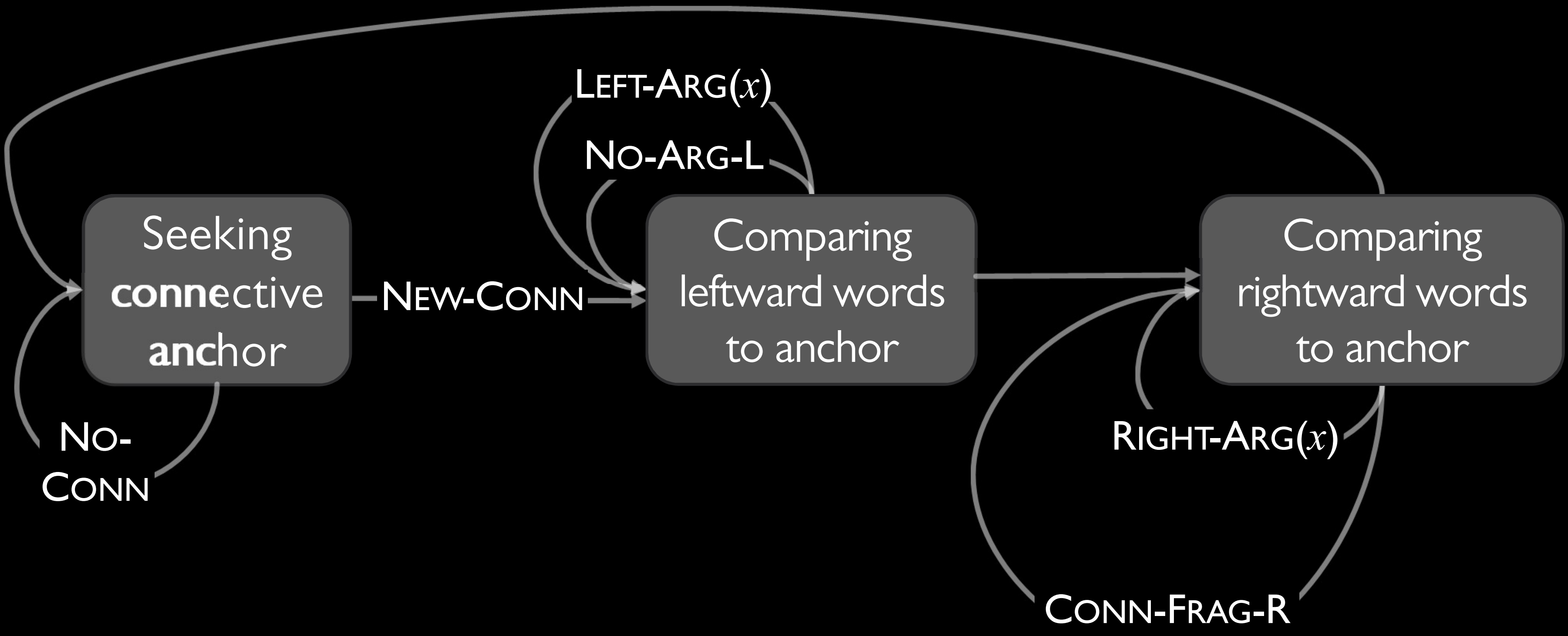
Next possible argument/fragment

Partially-constructed instance

they moved	because of	the schools
Effect	Connective	Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)


LEFT-ARG(x)
NO-ARG-L
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN
RIGHT-ARG(x)
CONN-FRAG-R

Tagger state

Well , they moved because of the schools

(Possible) connective anchor

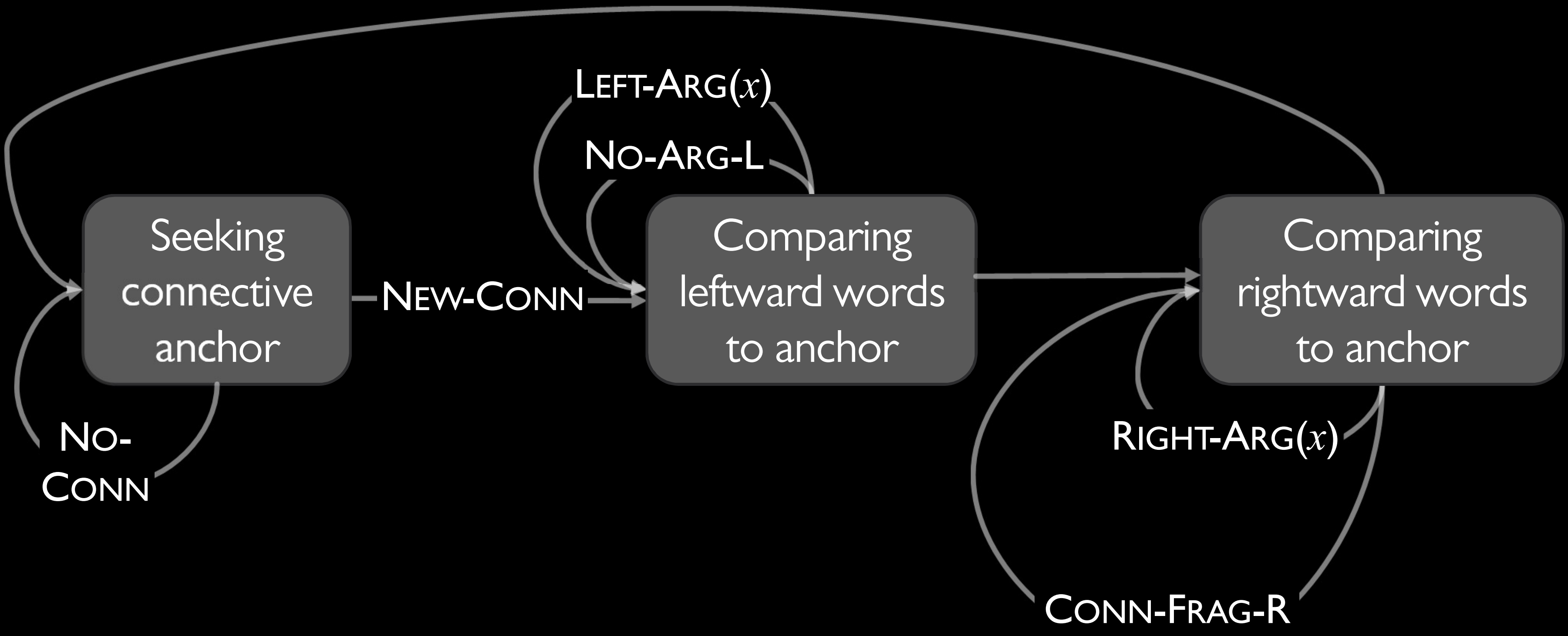
Next possible argument/fragment

Partially-constructed instance

they moved	because of	the schools
Effect	Connective	Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)



Seeking connective anchor

New-Conn

No-Conn

Comparing leftward words to anchor

Left-Arg(x)

No-Arg-L

Comparing rightward words to anchor

Right-Arg(x)

Conn-Frag-R

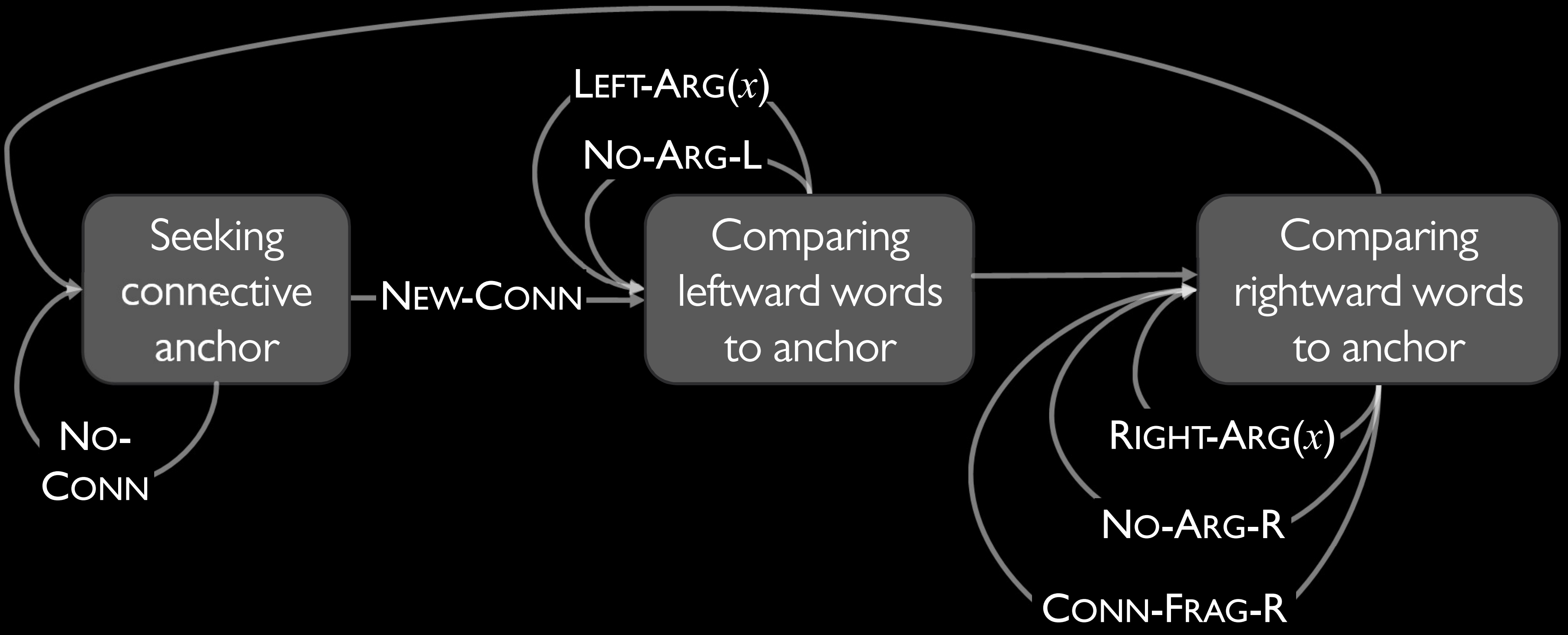
Fully-constructed instance

they moved because of the schools

Effect Connective Cause

The DeepCx transition scheme

(Heavily modified from Choi and Palmer, 2011)

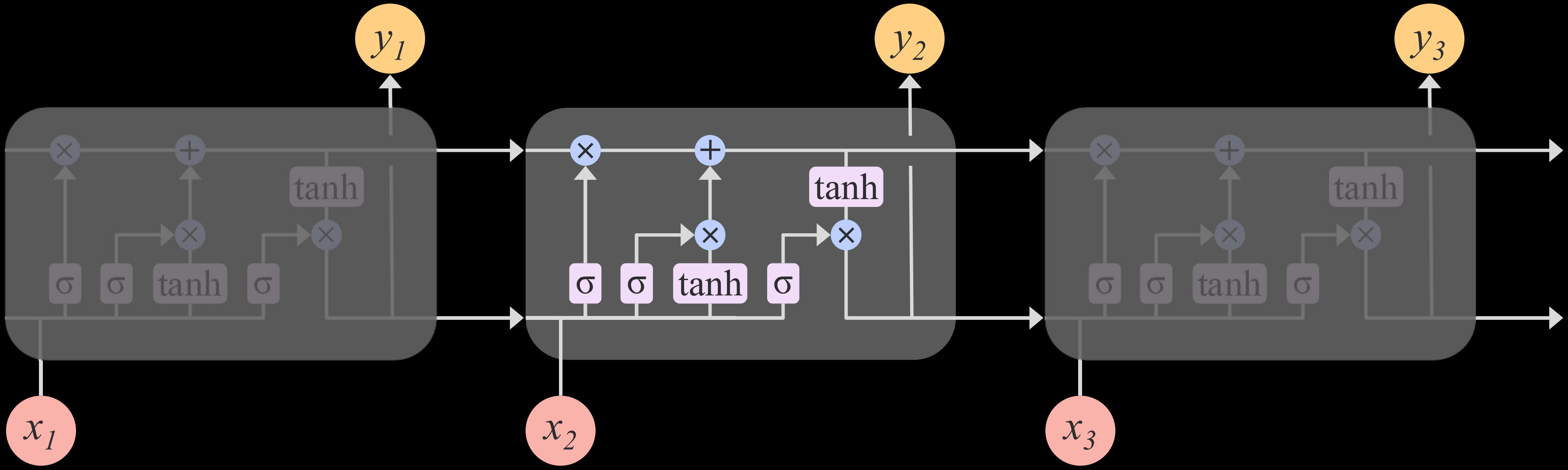

LEFT-ARG(x)
NO-ARG-L
Seeking connective anchor
NEW-CONN
Comparing leftward words to anchor
Comparing rightward words to anchor
NO-CONN
RIGHT-ARG(x)
NO-ARG-R
CONN-FRAG-R

Fully-constructed instance

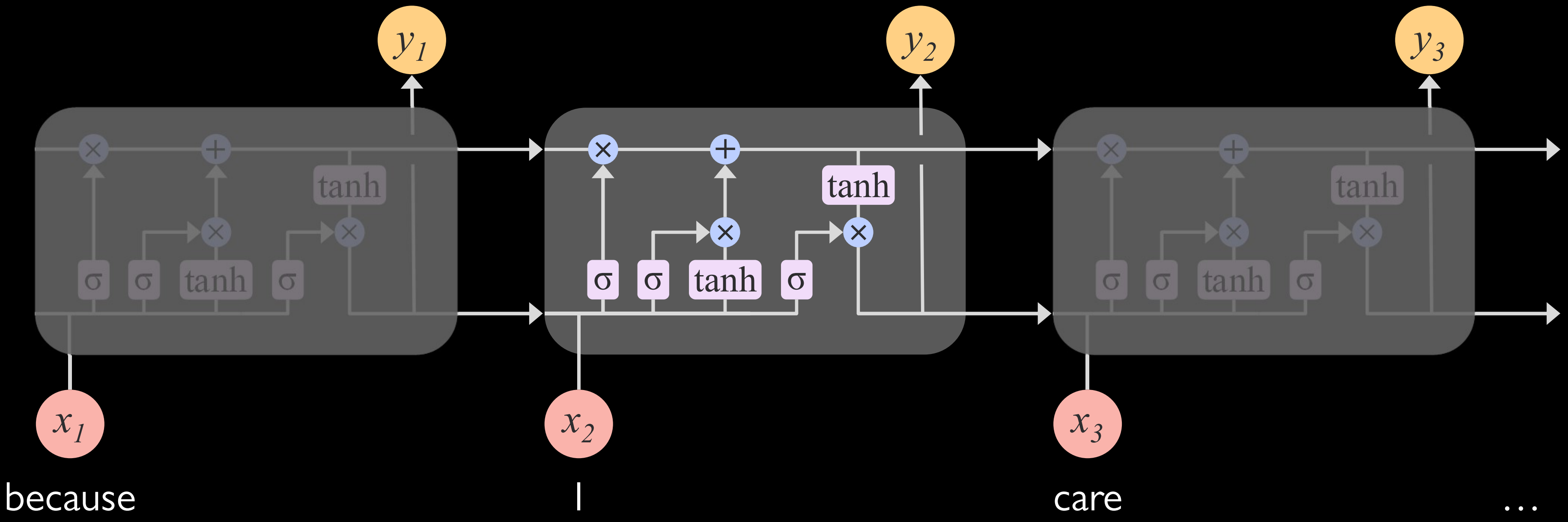
they moved because of the schools

Effect Connective Cause

DeepCx uses long short-term memory (LSTM) networks to embed sequences of words.

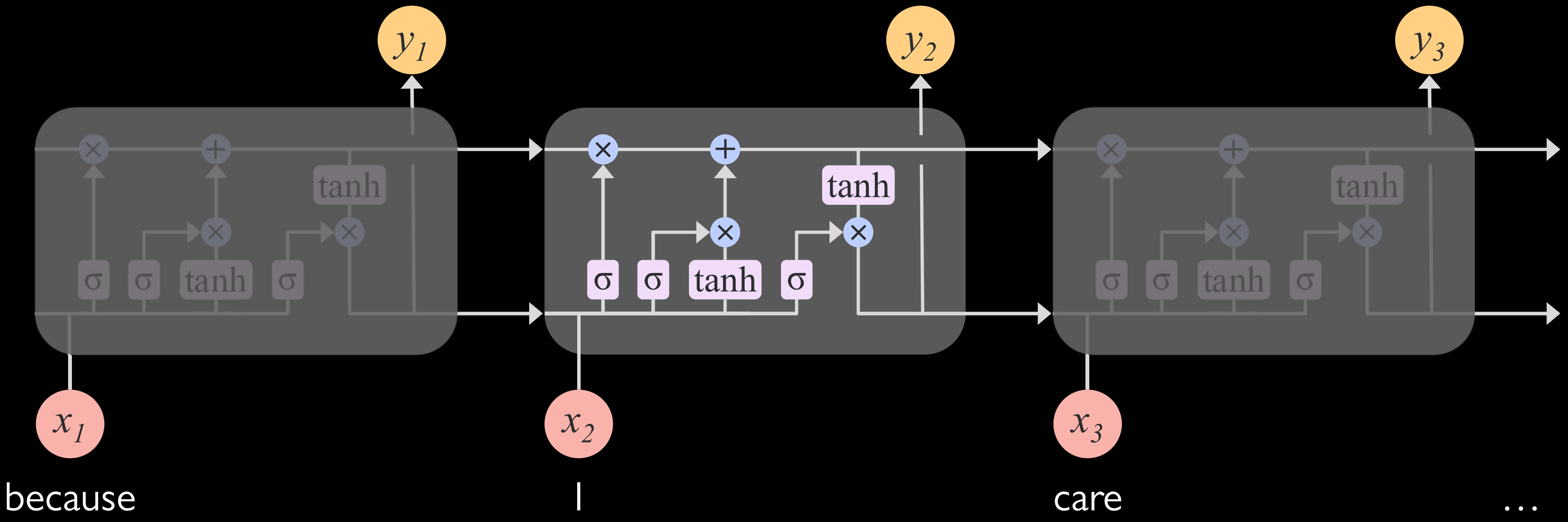


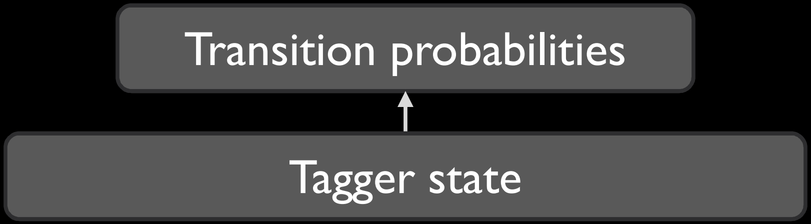
DeepCx uses long short-term memory (LSTM) networks to embed sequences of words.



DeepCx uses long short-term memory (LSTM) networks to embed sequences of words.

Summary of inputs so far

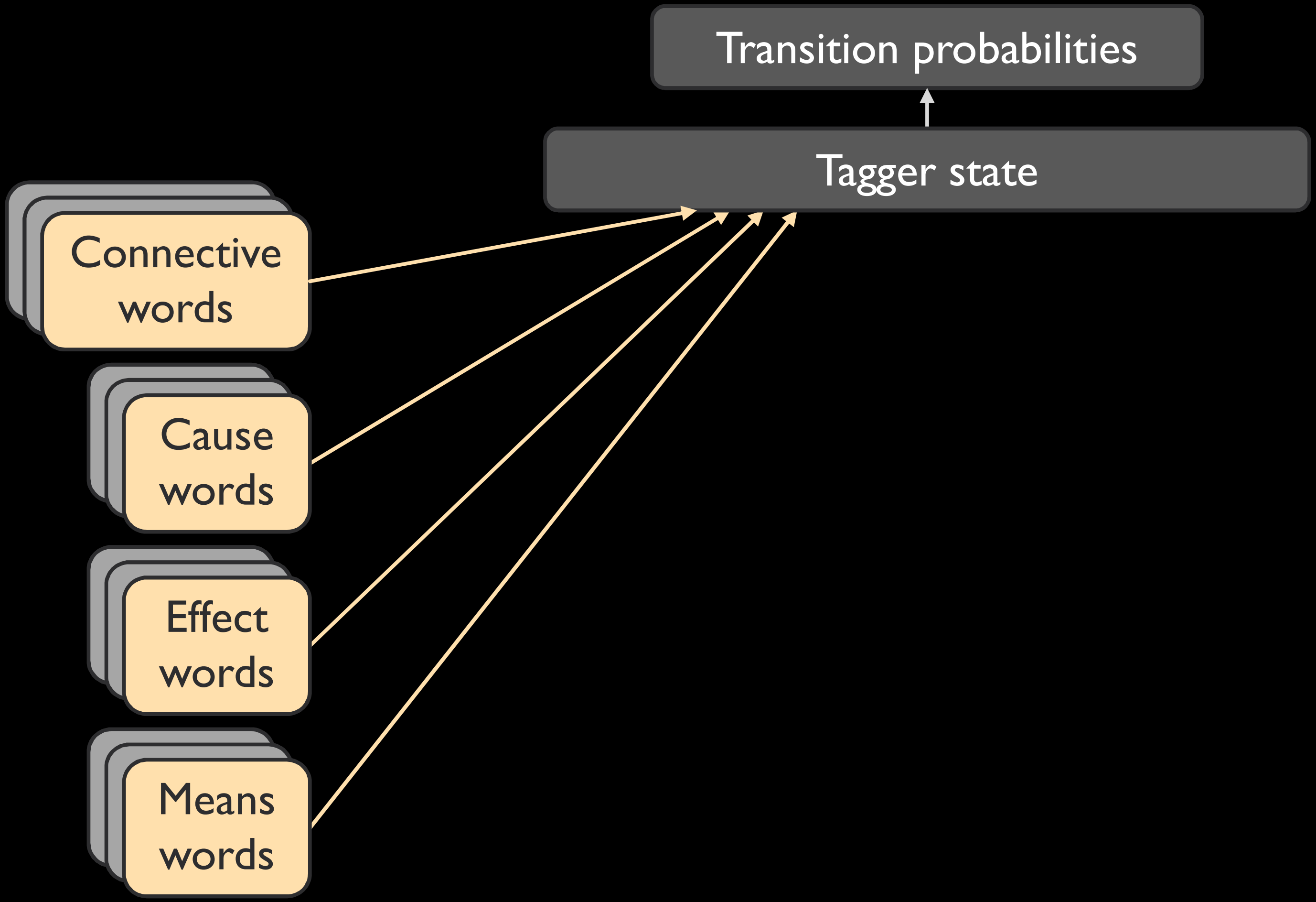




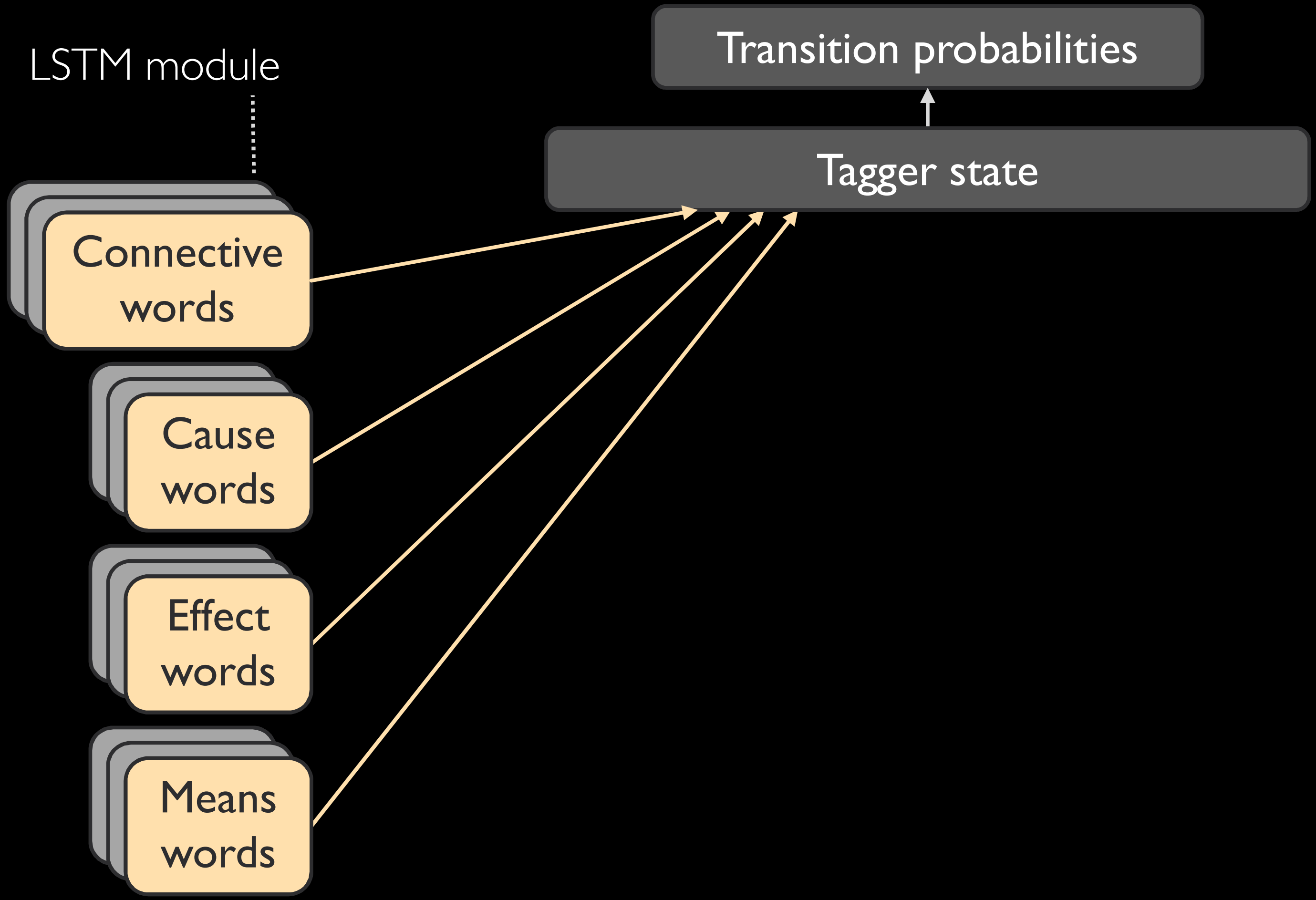
Transition probabilities

Tagger state

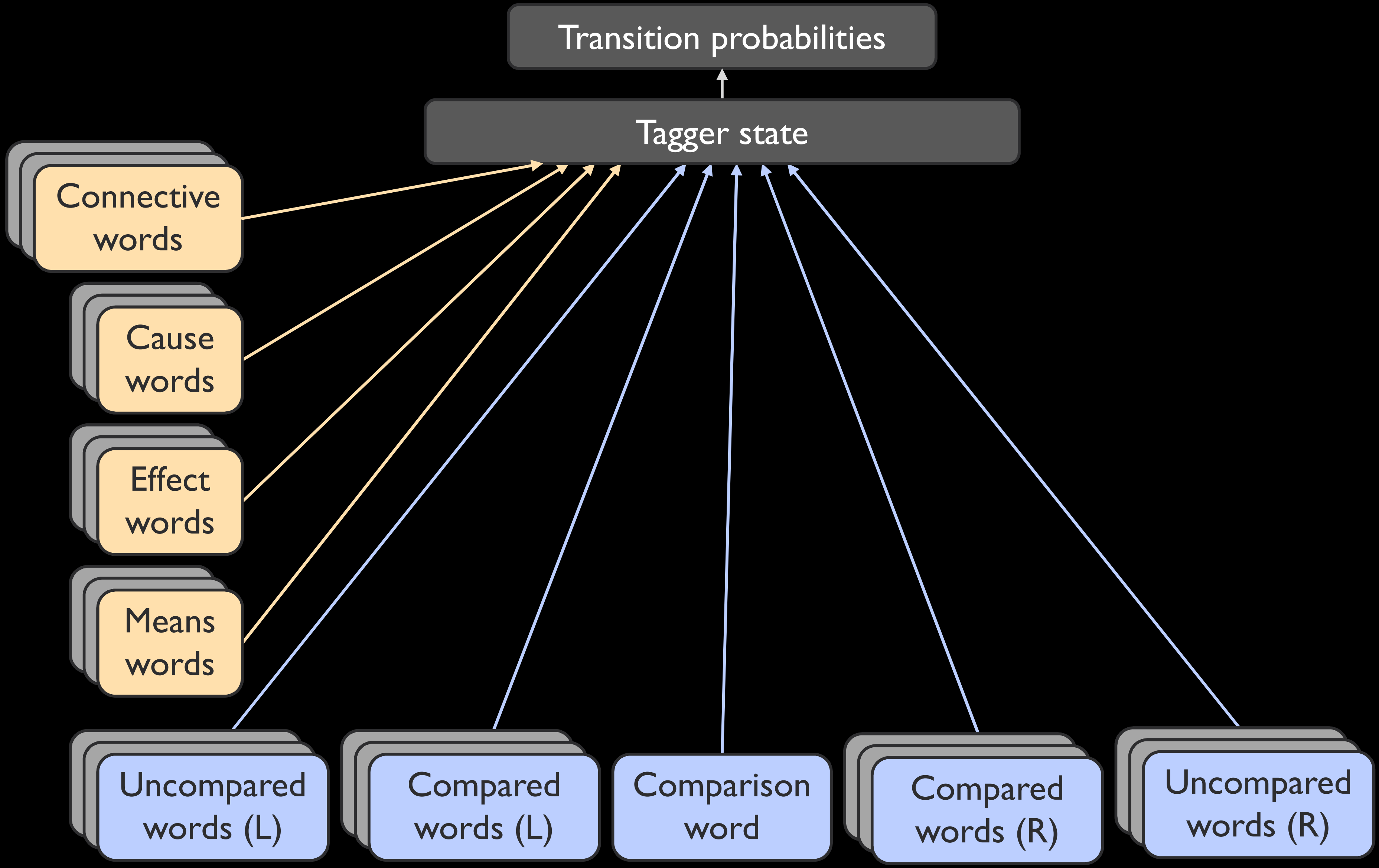
(Based on Dyer et al., 2015)



(Based on Dyer et al., 2015)



(Based on Dyer et al., 2015)



Transition probabilities

Tagger state

Connective words

Cause words

Effect words

Means words

Uncompared words (L)

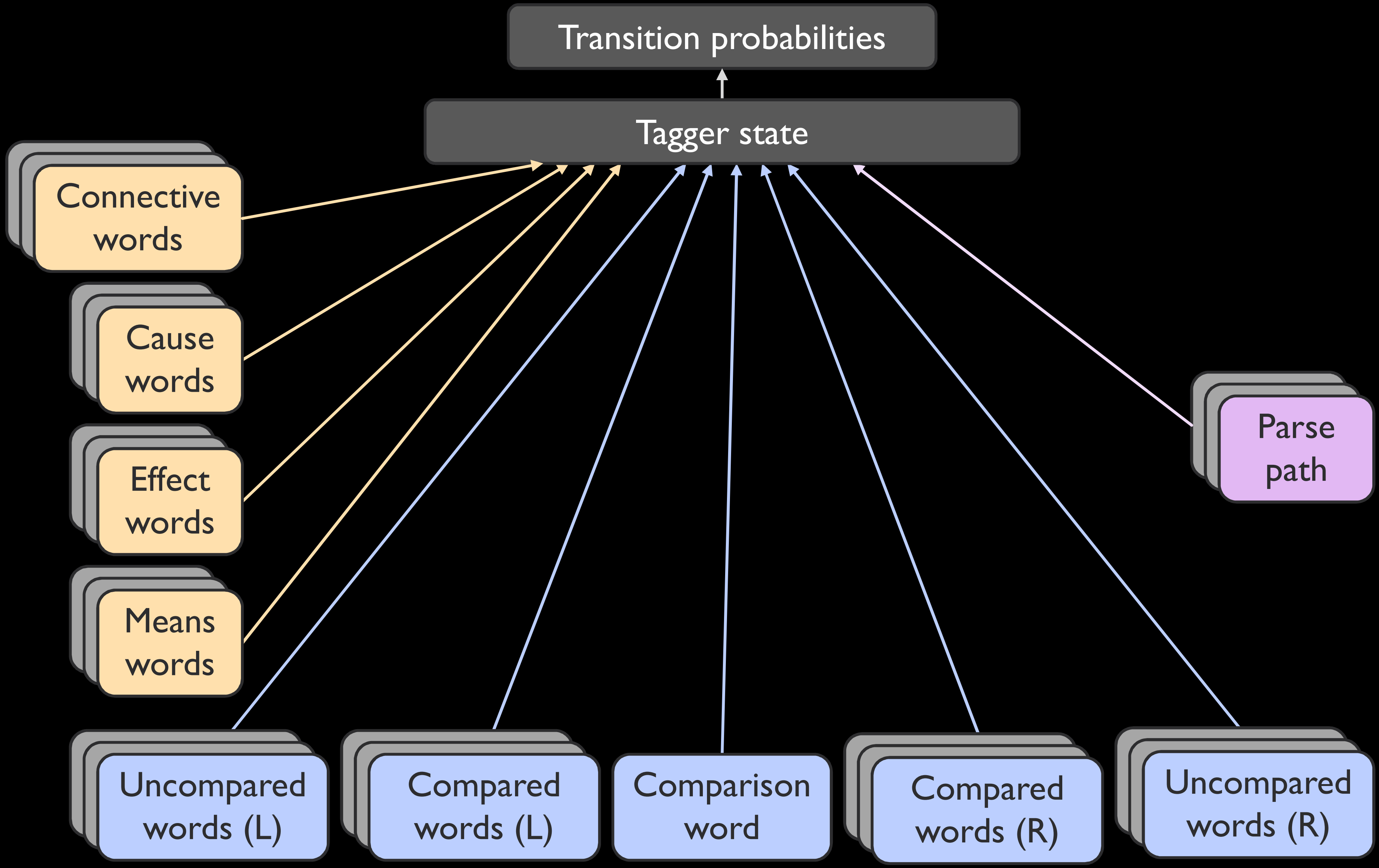
Compared words (L)

Comparison word

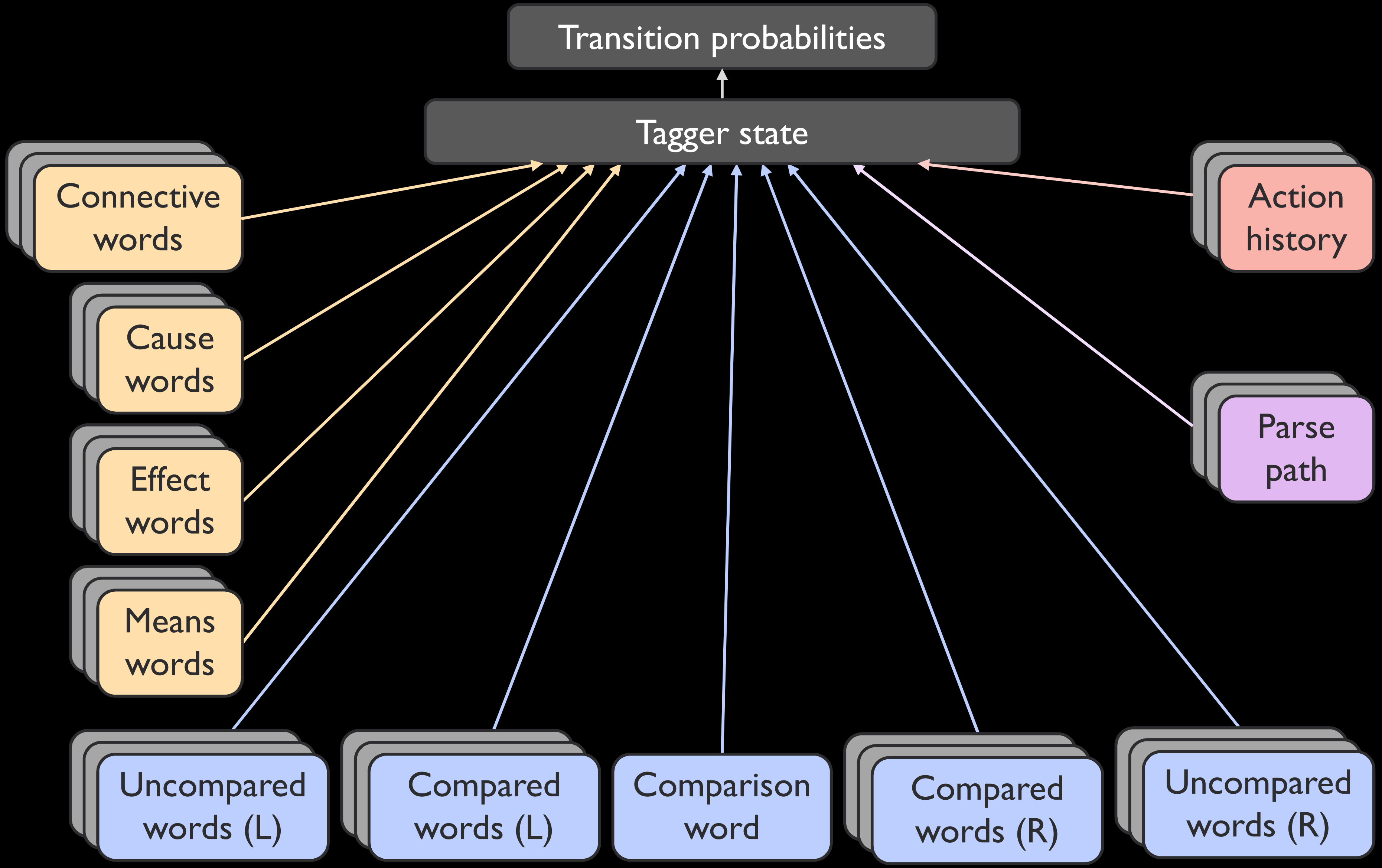
Compared words (R)

Uncompared words (R)

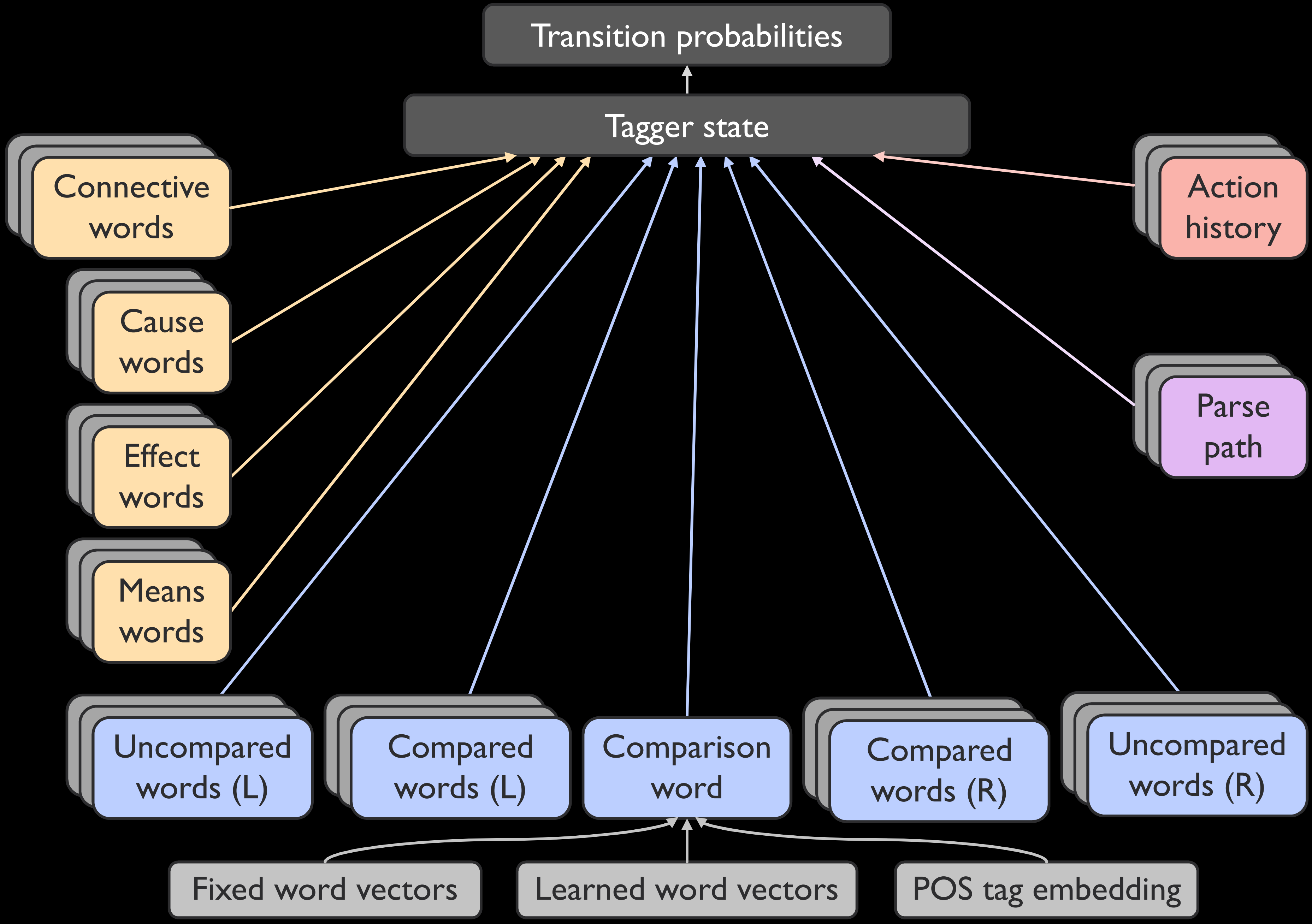
(Based on Dyer et al., 2015)



(Based on Dyer et al., 2015)



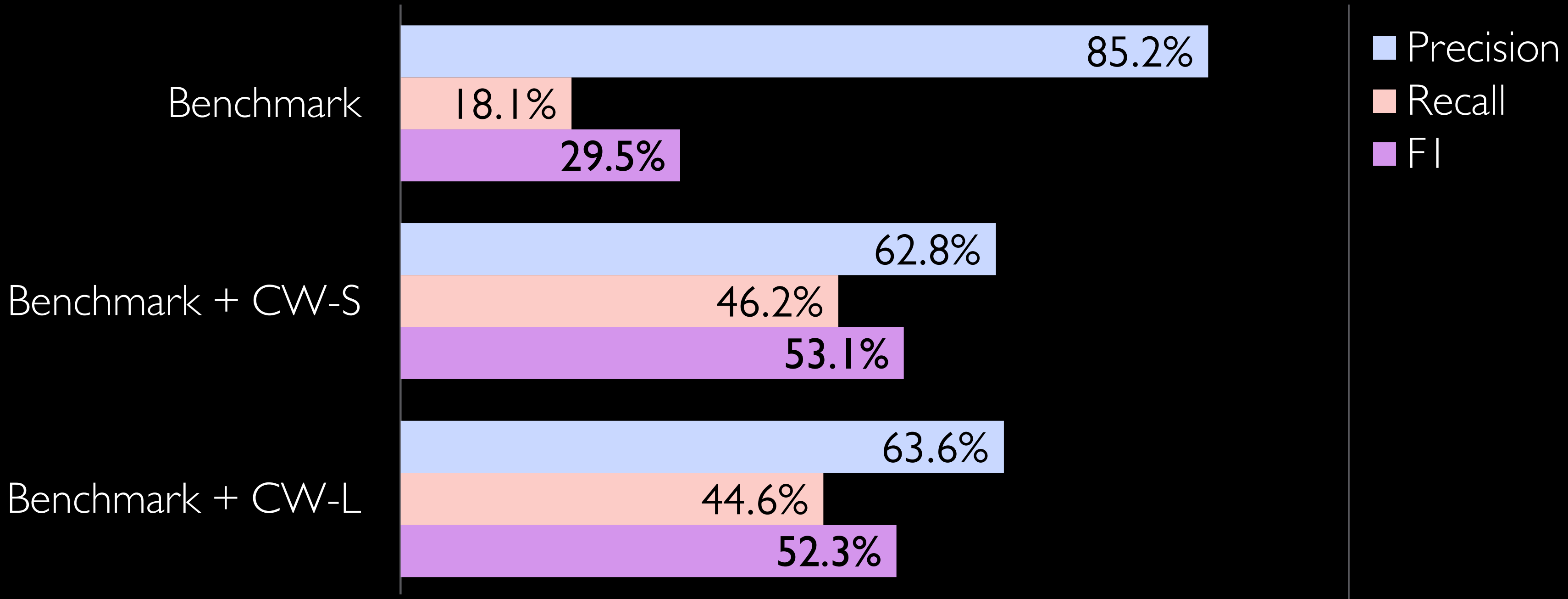
(Based on Dyer et al., 2015)



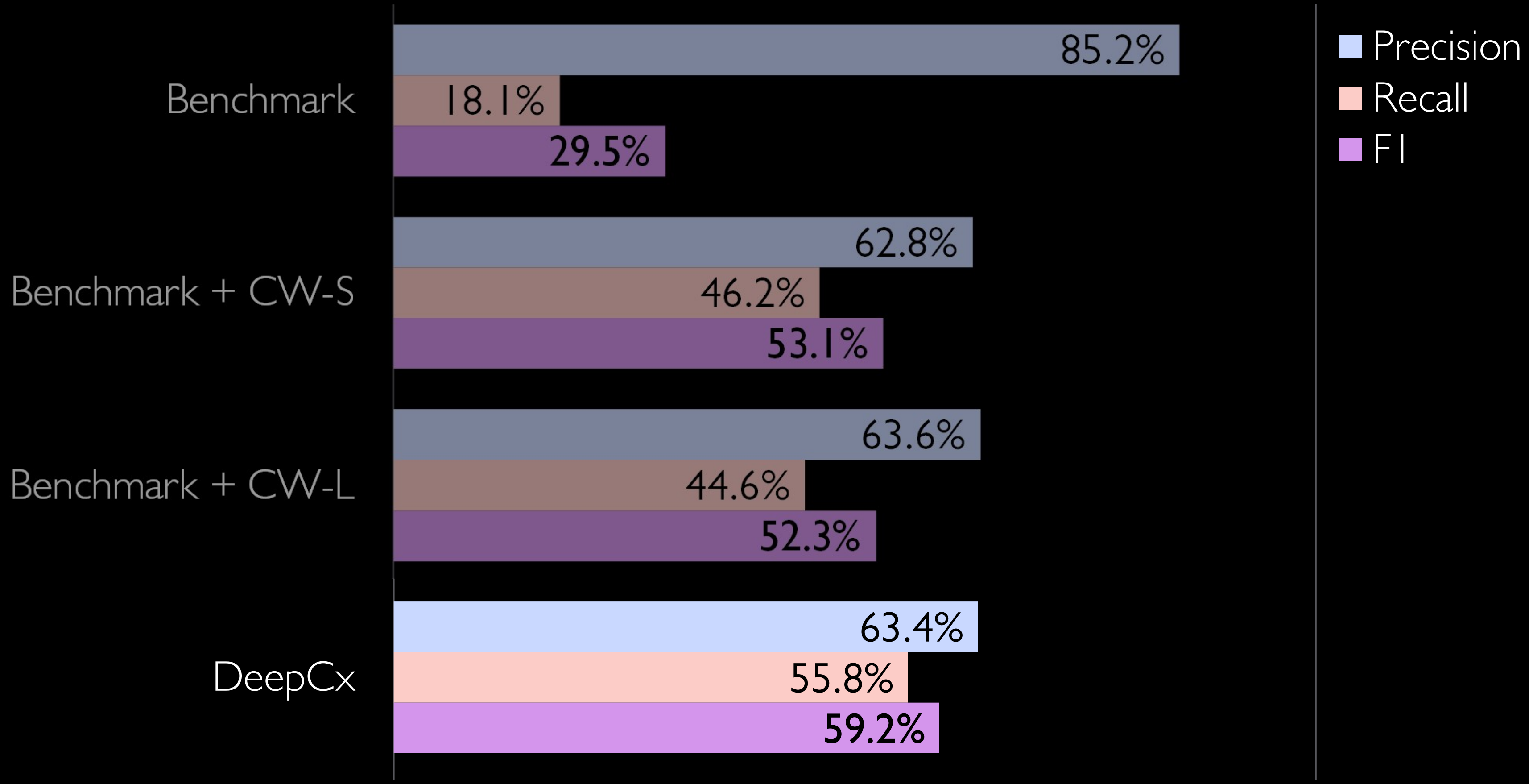
(Based on Dyer et al., 2015)

DeepCx significantly outperforms Causeway on connective discovery.

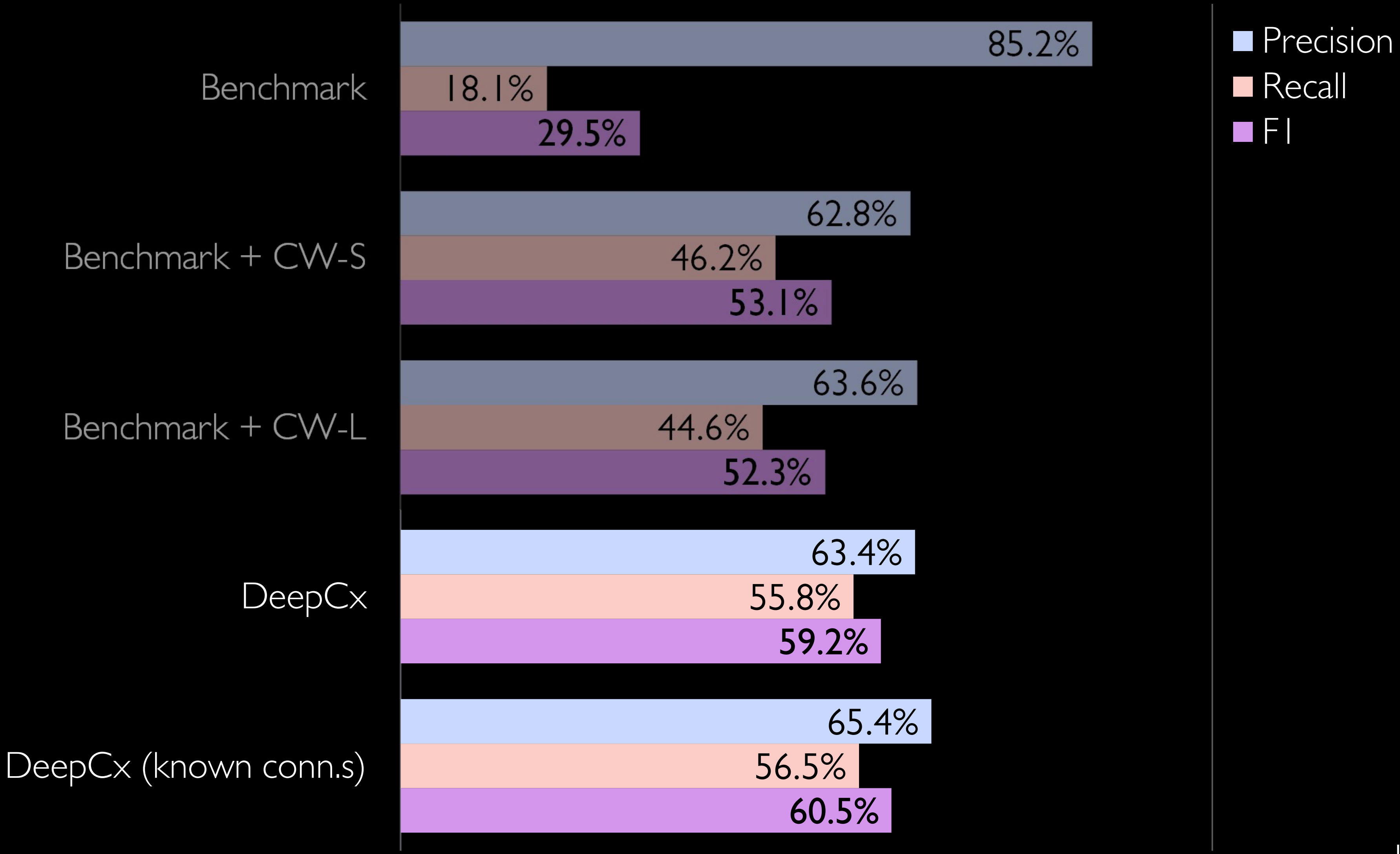
DeepCx significantly outperforms Causeway on connective discovery.



DeepCx significantly outperforms Causeway on connective discovery.

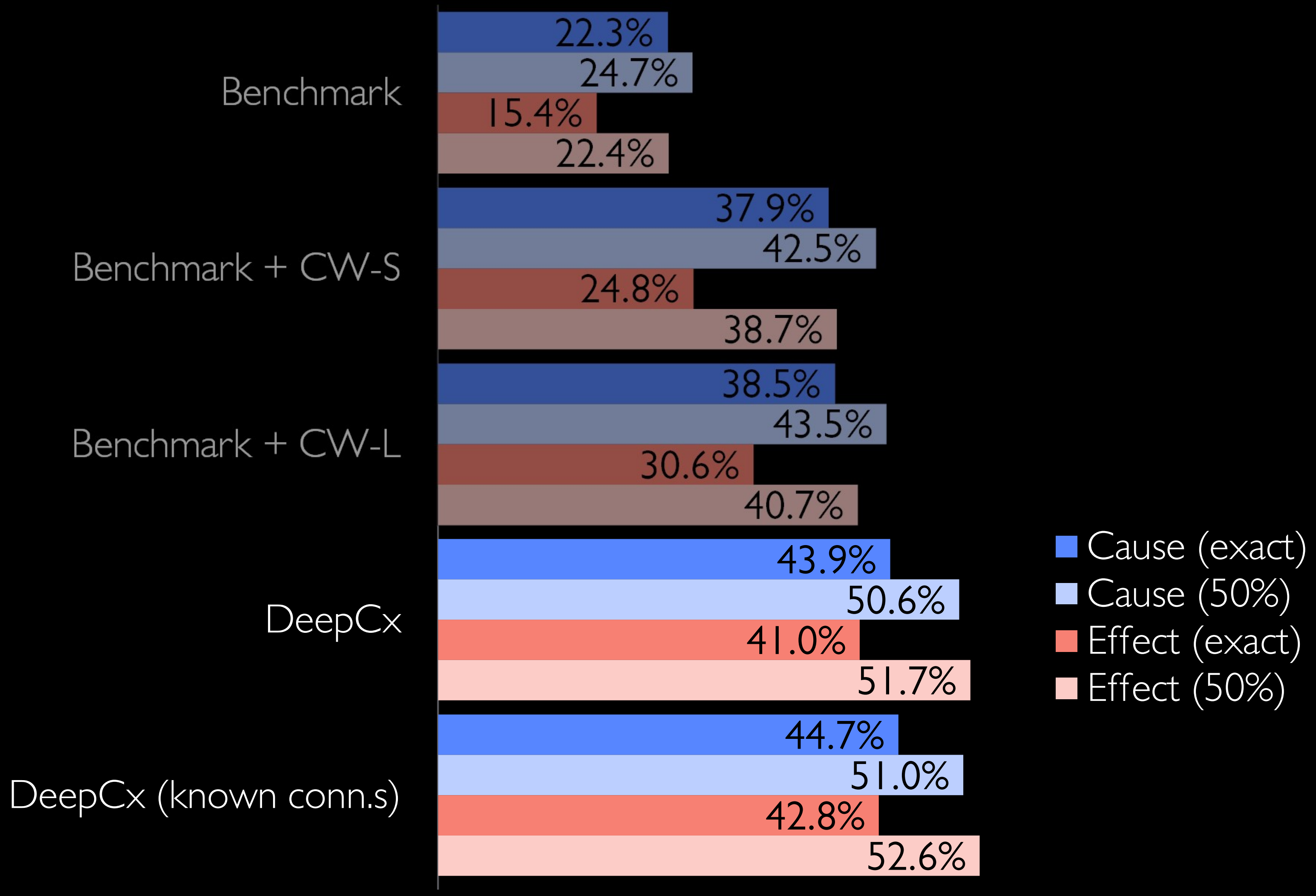


DeepCx significantly outperforms Causeway on connective discovery.

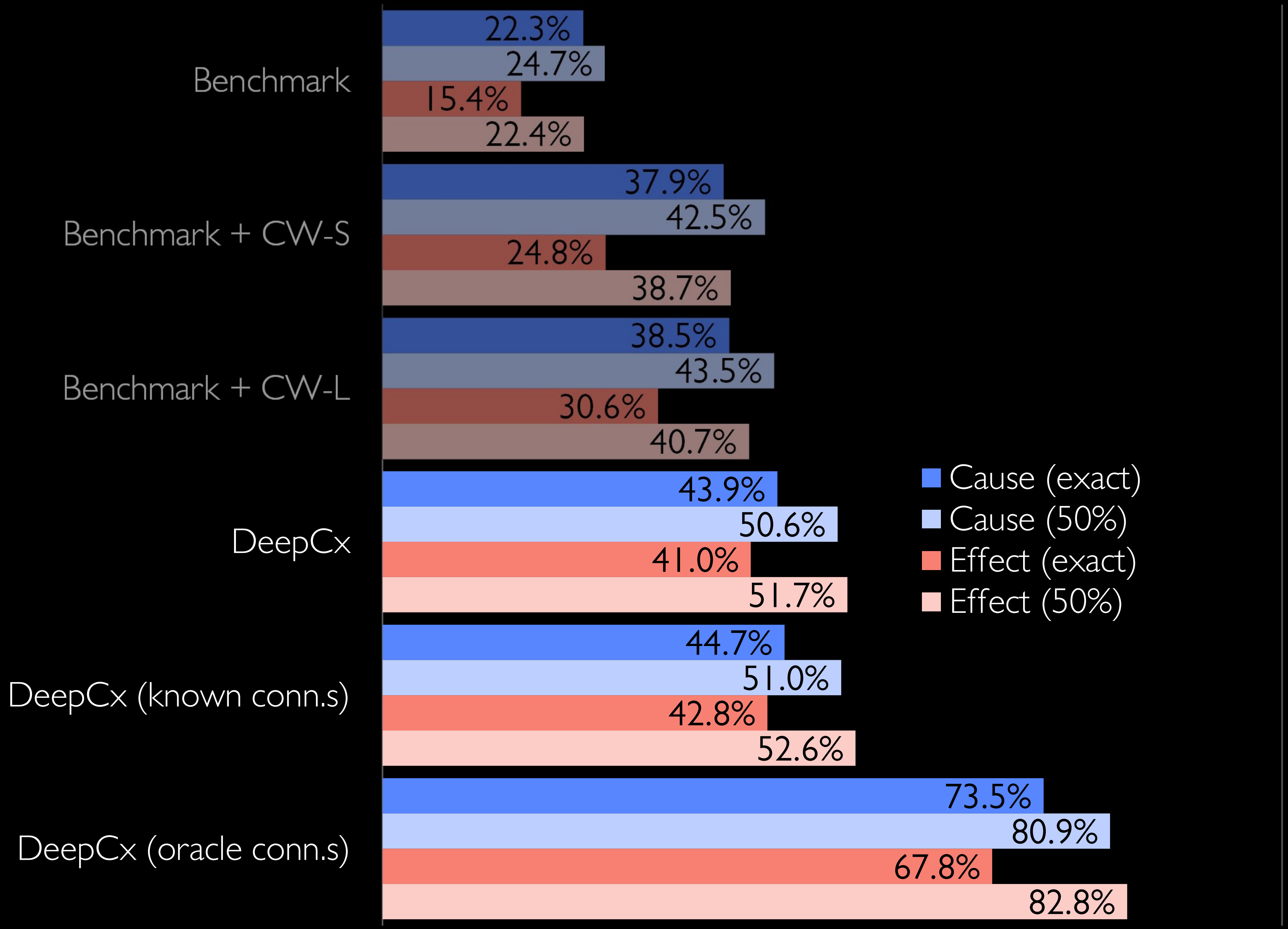


DeepCx also significantly outperforms Causeway on argument identification.

DeepCx also significantly outperforms Causeway on argument identification.



DeepCx also significantly outperforms Causeway on argument identification.



Today's talk:

1. The BECAUSE annotation scheme & corpus of causal language

2. Causeway-L/Causeway-S: two pattern-based taggers for causal constructions

3. DeepCx: a neural, transition-based tagger for causal constructions

Contributions

1. The "constructions on top" approach to operationalizing CxG

2. A COT-based approach to comprehensively annotating causal language

3. Pattern-based methods & architecture for tagging causal constructions

4. Transition scheme & DNN architecture for tagging complex constructions

Contributions

1. The "constructions on top" approach to operationalizing CxG
2. A COT-based approach to comprehensively annotating causal language ------→ BECAUSE [1]
3. Pattern-based methods & architecture for tagging causal constructions ------→ Causeway [2]
4. Transition scheme & DNN architecture for tagging complex constructions ------→ DeepCx [3]

[1] bit.ly/BECauSE

[2] bit.ly/CausewayTagger

[3] bit.ly/DeepCxTagger