PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

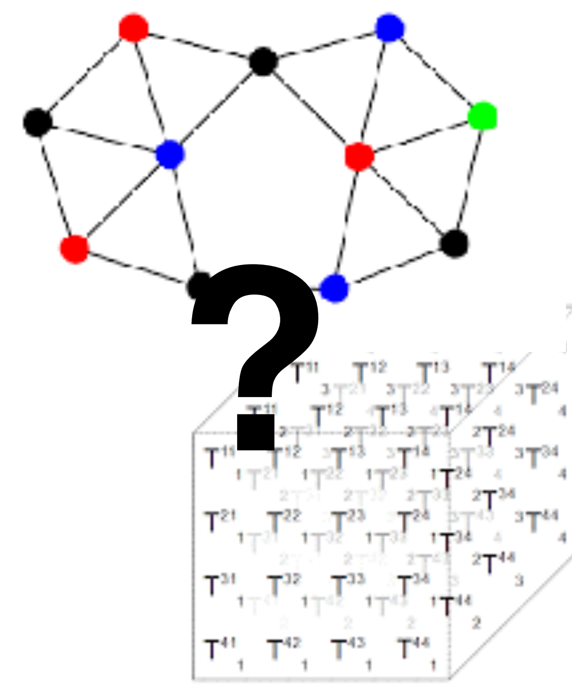
Semantic Role Labeling with Labeled Span Graph Networks

Luheng He
Google A.I.
Oct. 2018

Work done at the University of Washington

Predicting Rich Semantic Structure with Simple Models

On January 13, 2018, a false ballistic missile alert was issued via the Emergency Alert System and Commercial Mobile Alert System over television, radio, and cellphones in the U.S. state of Hawaii. The alert stated that there was an incoming ballistic missile threat to Hawaii, advised residents to seek shelter, and concluded "This is not a drill". The message was sent at 8:07 a.m. local time.



Input Document

Semantic Structure

Task: Semantic Role Labeling (SRL)

On January 13, 2018, a false ballistic missile alert was issued via the Emergency Alert System and Commercial Mobile Alert System over television, radio, and cellphones in the U.S. state of Hawaii.

“Who did what to whom, when and where”

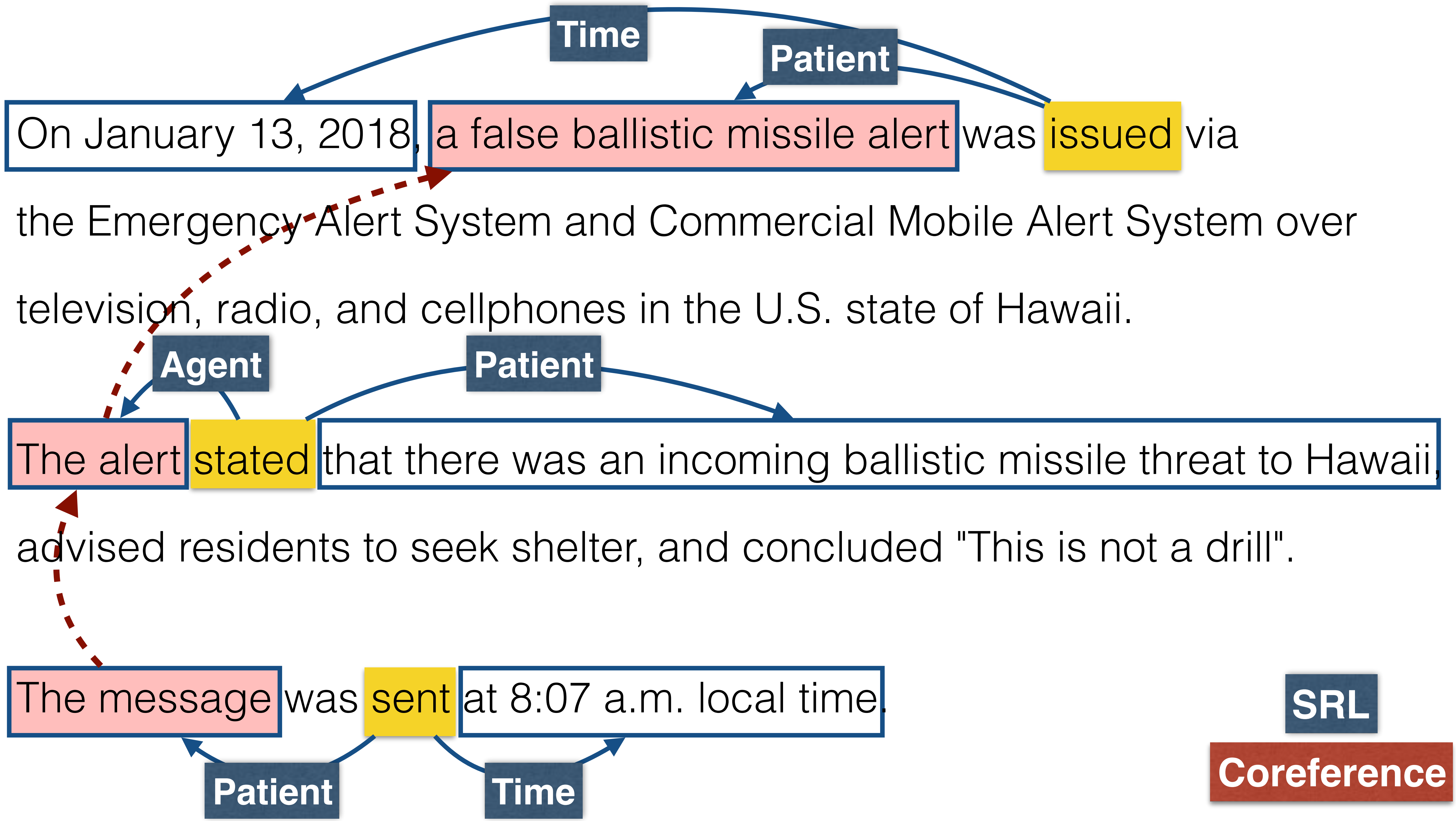
The alert ... ng ballistic missile threat to Hawaii, advised r ... ncluded "This is not a drill".

The message was **sent** at 8:07 a.m. local time.

Patient **Time** **SRL**

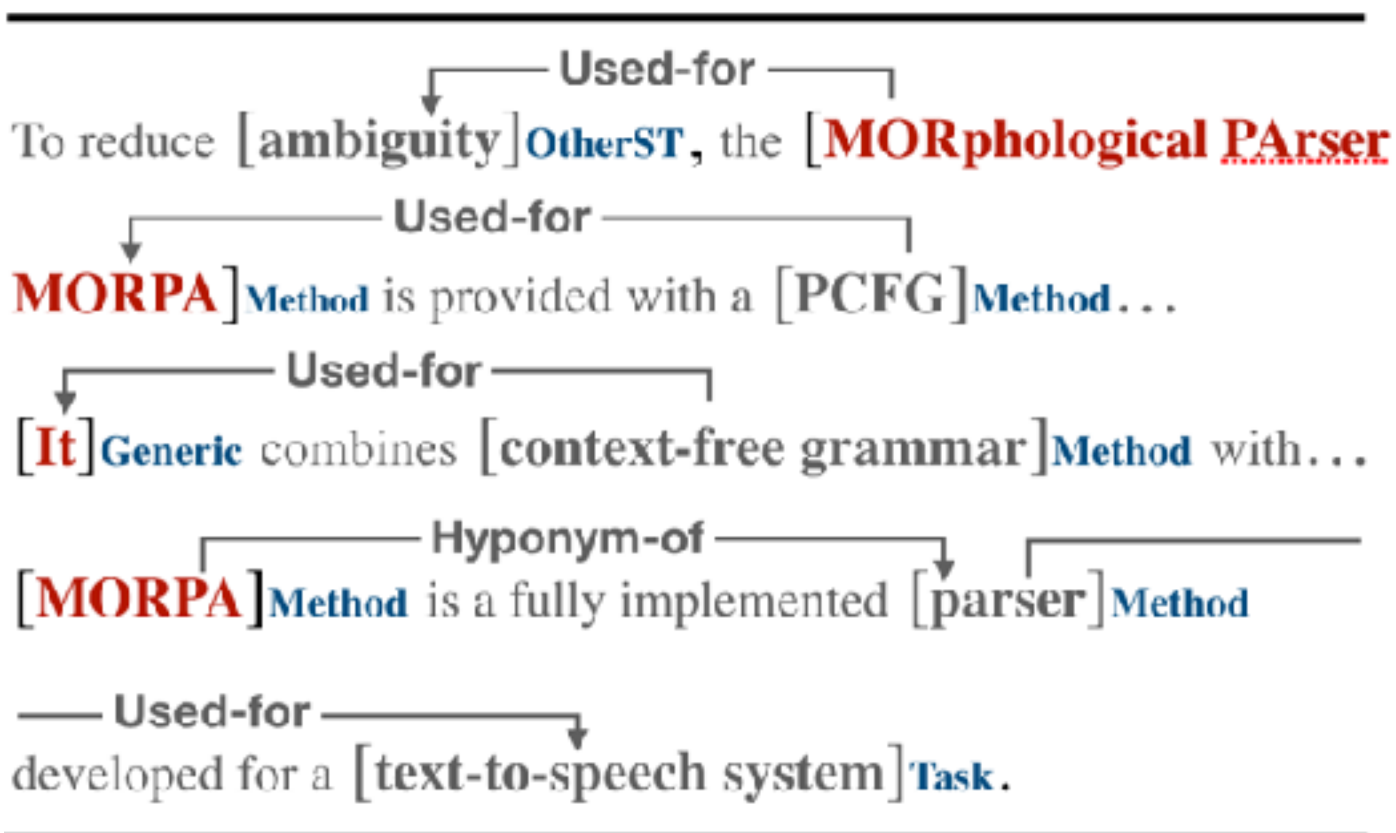
From Wikipedia: 2018 Hawaii false missile alert. Only part of the structures are visualized.

Adding Coreference Resolution



From Wikipedia: 2018 Hawaii false missile alert. Only part of the structures are visualized.

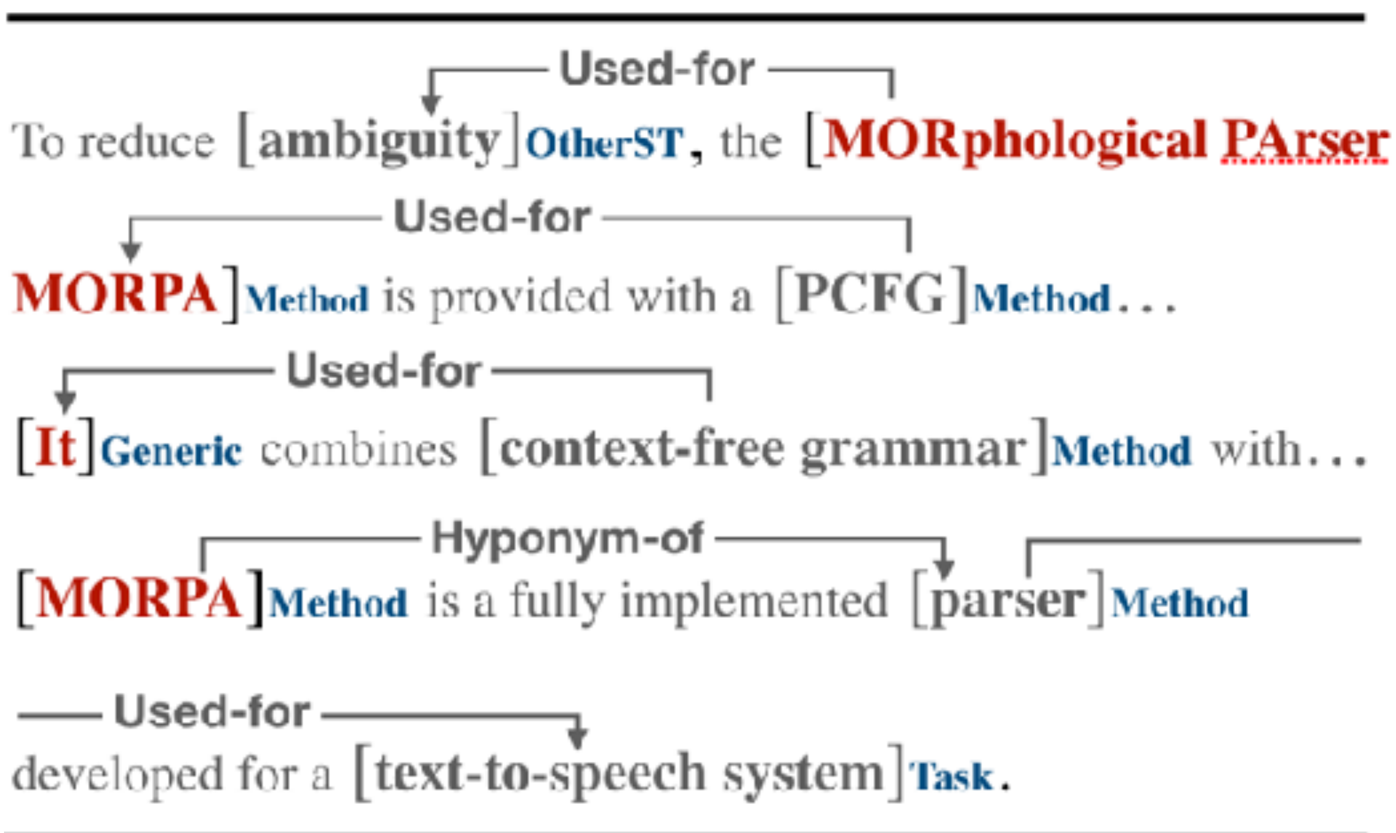
Another example: Relation Extraction on Scientific Documents

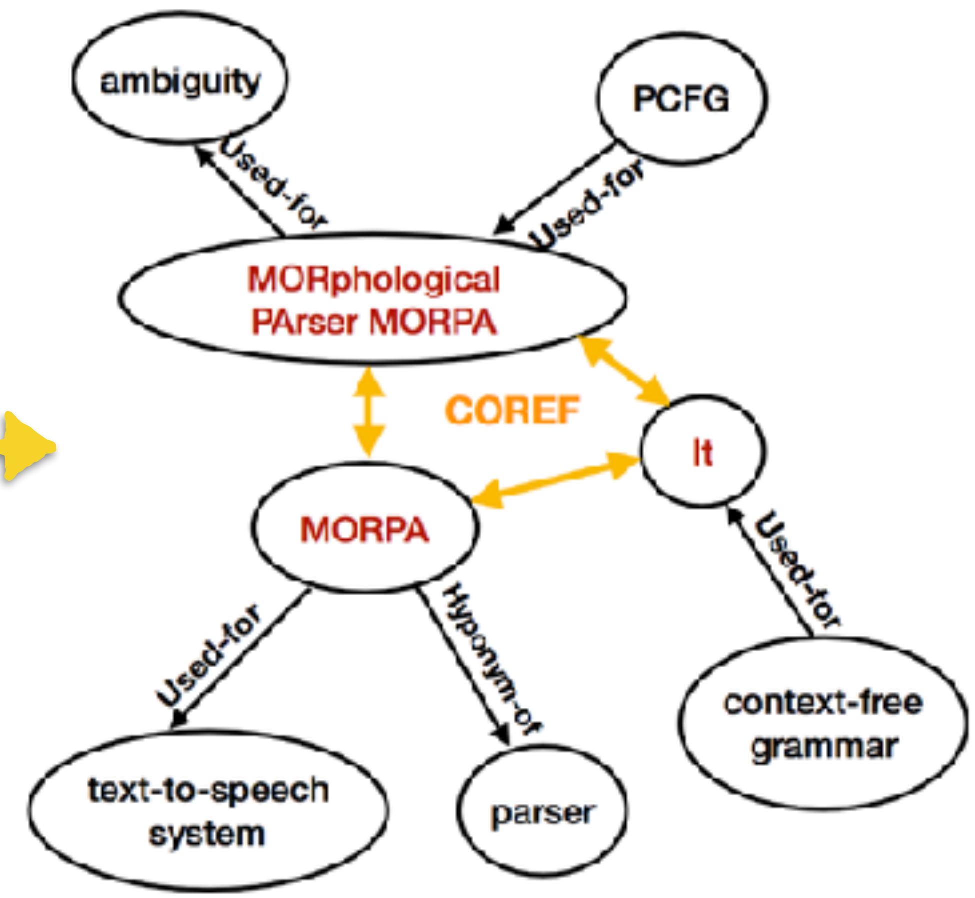


Annotation with **entities**, **relations**, and **coreference**

SciERC (Entity, Relation, Coreference): Luan et al., 2018

Another example: Relation Extraction on Scientific Documents



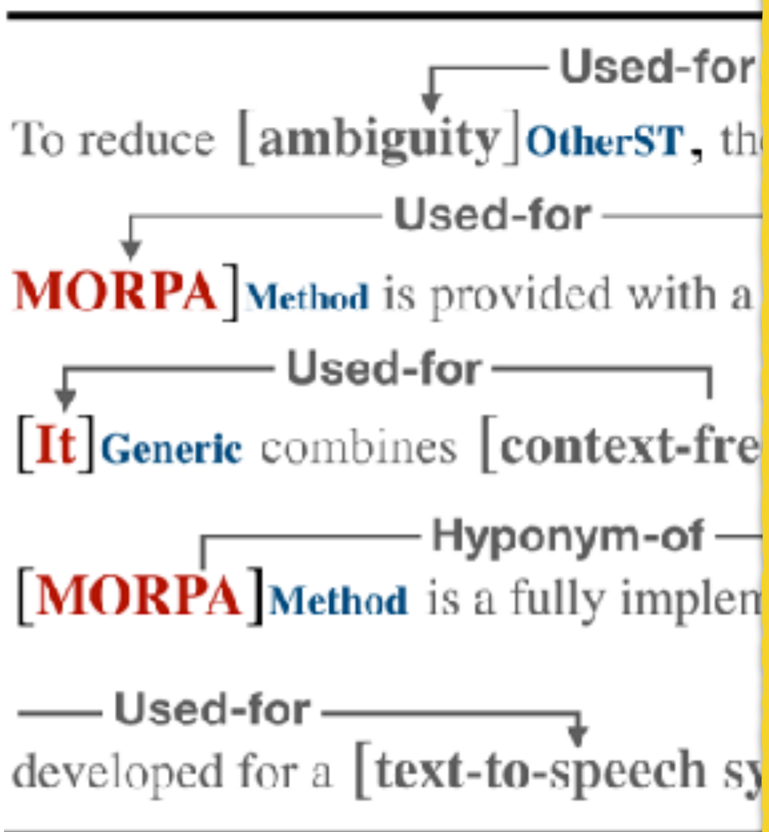


Annotation with **entities**, **relations**, and **coreference**

Document-level KG

SciERC (Entity, Relation, Coreference): Luan et al., 2018

Another example: Relation Extraction



Annotation with
relations, and **co**

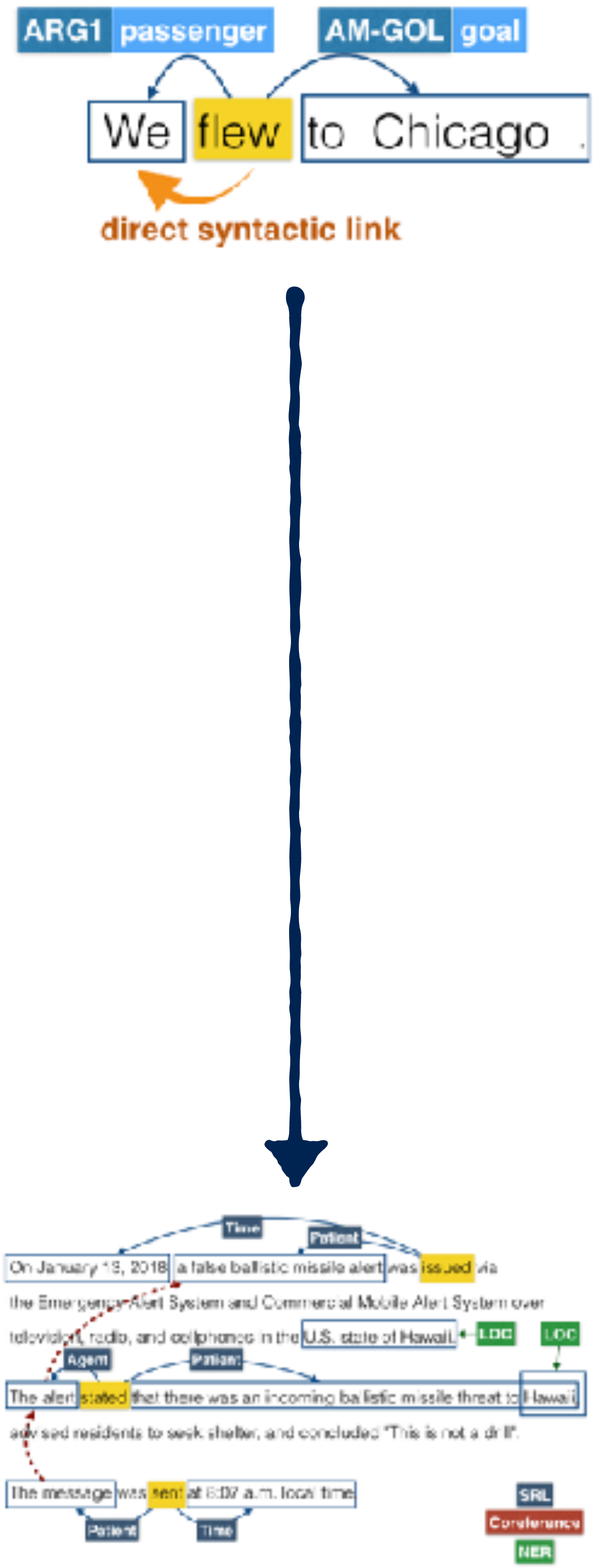
Goal: A unified model for all these tasks.

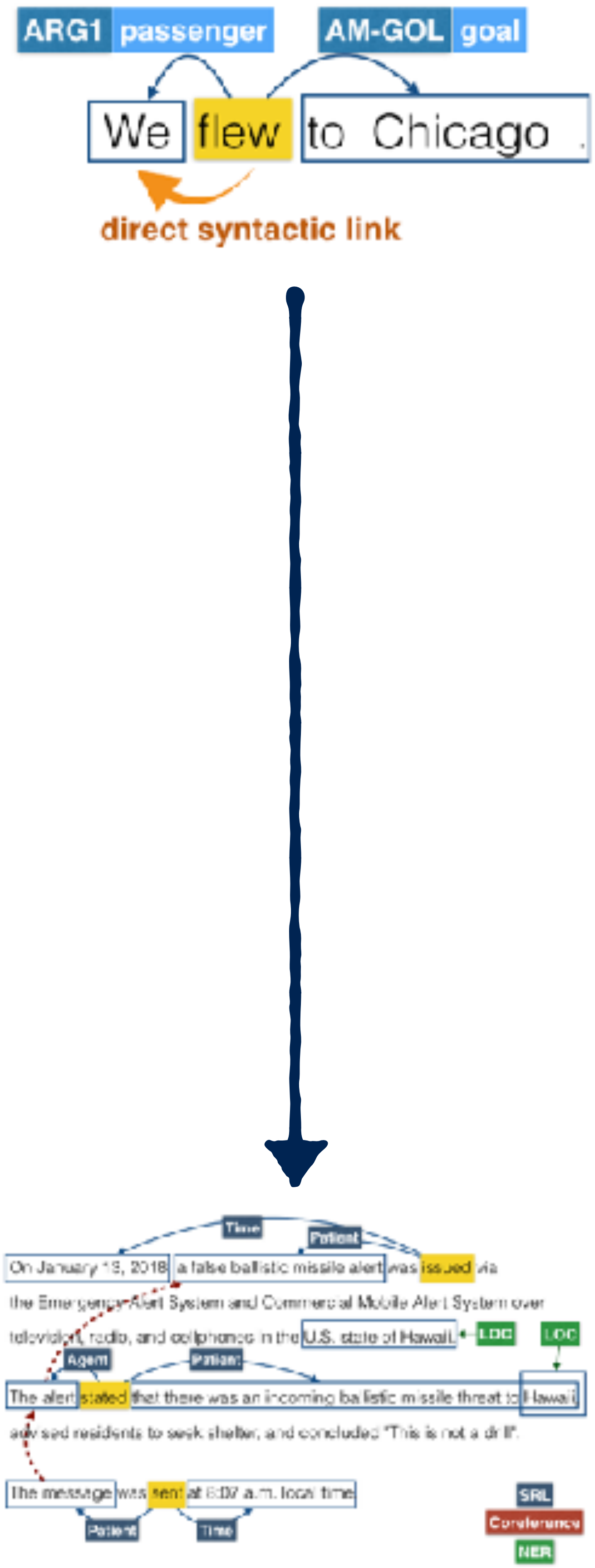
Challenge: Very different structures, task-specific pipelines/features/architectures …

This talk:
1) Build end-to-end models for SRL.
2) Generalizes such model to all tasks.

SciERC (Entity, Relation, Coreference): Luan et al., 2018

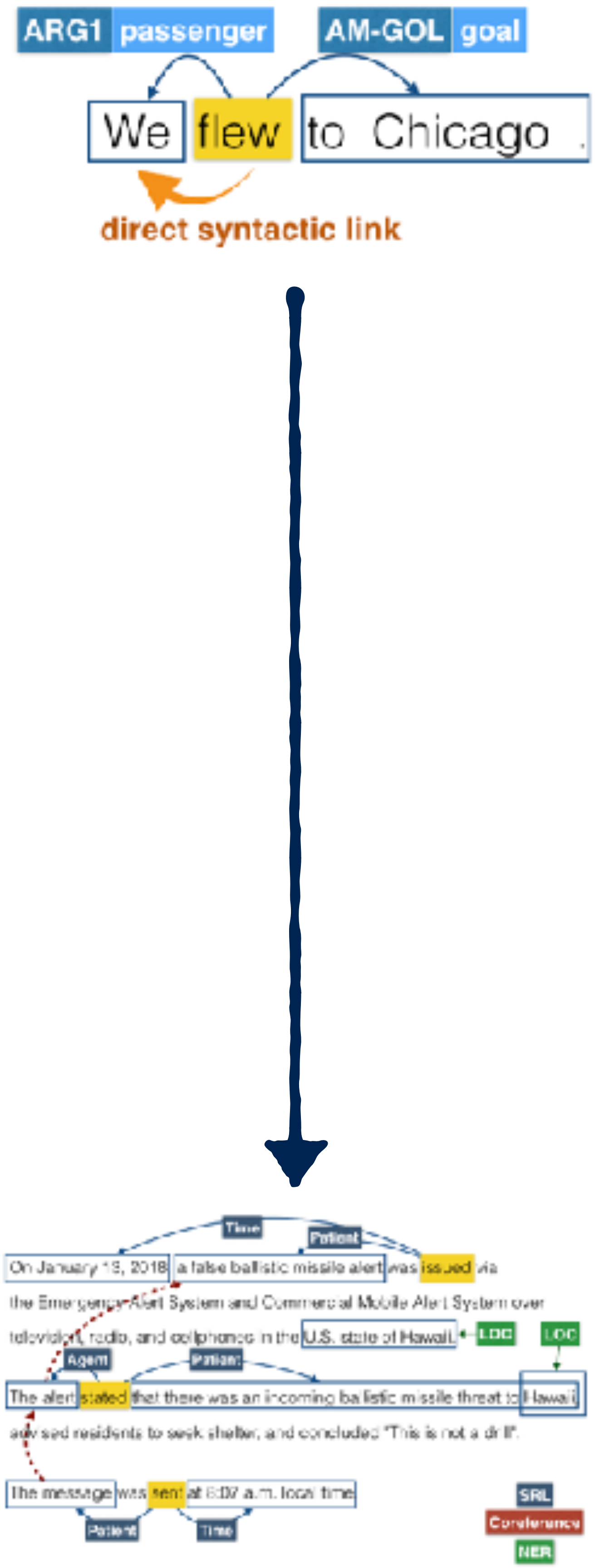
Contributions





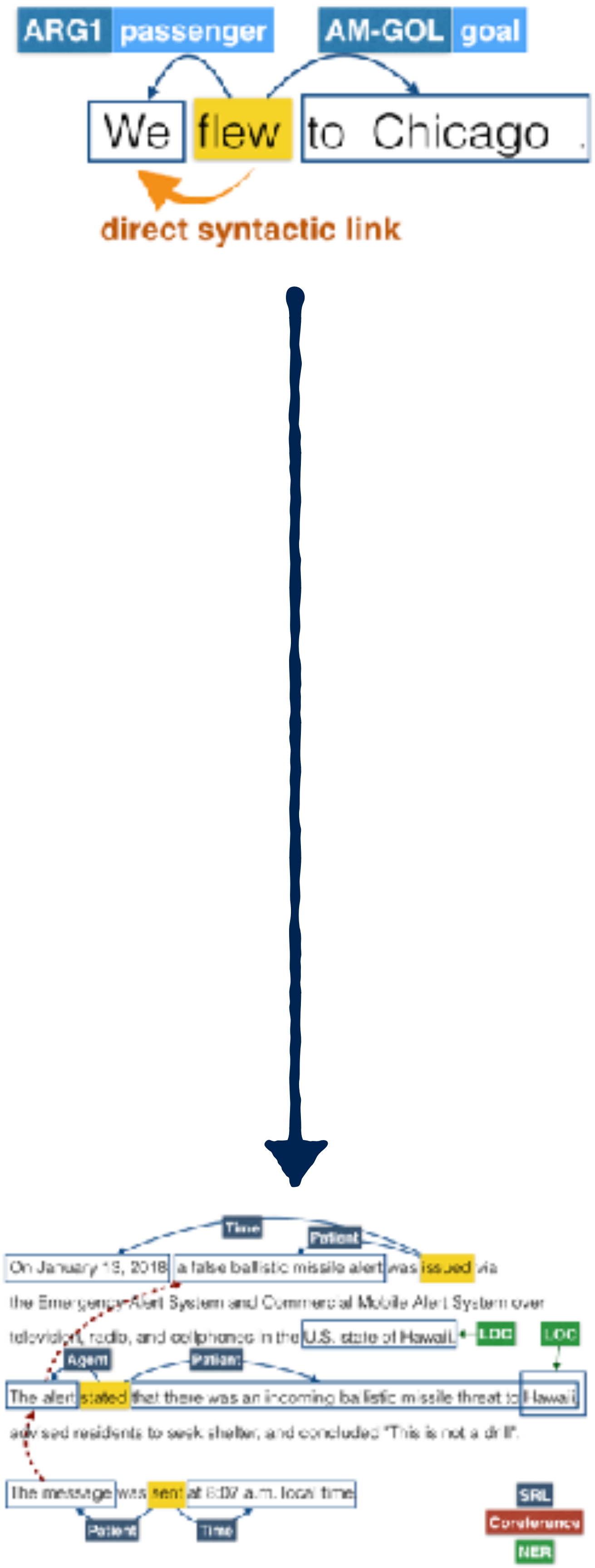
Contributions

- **End-to-end** prediction of SRL structure, without relying on NLP pipeline.



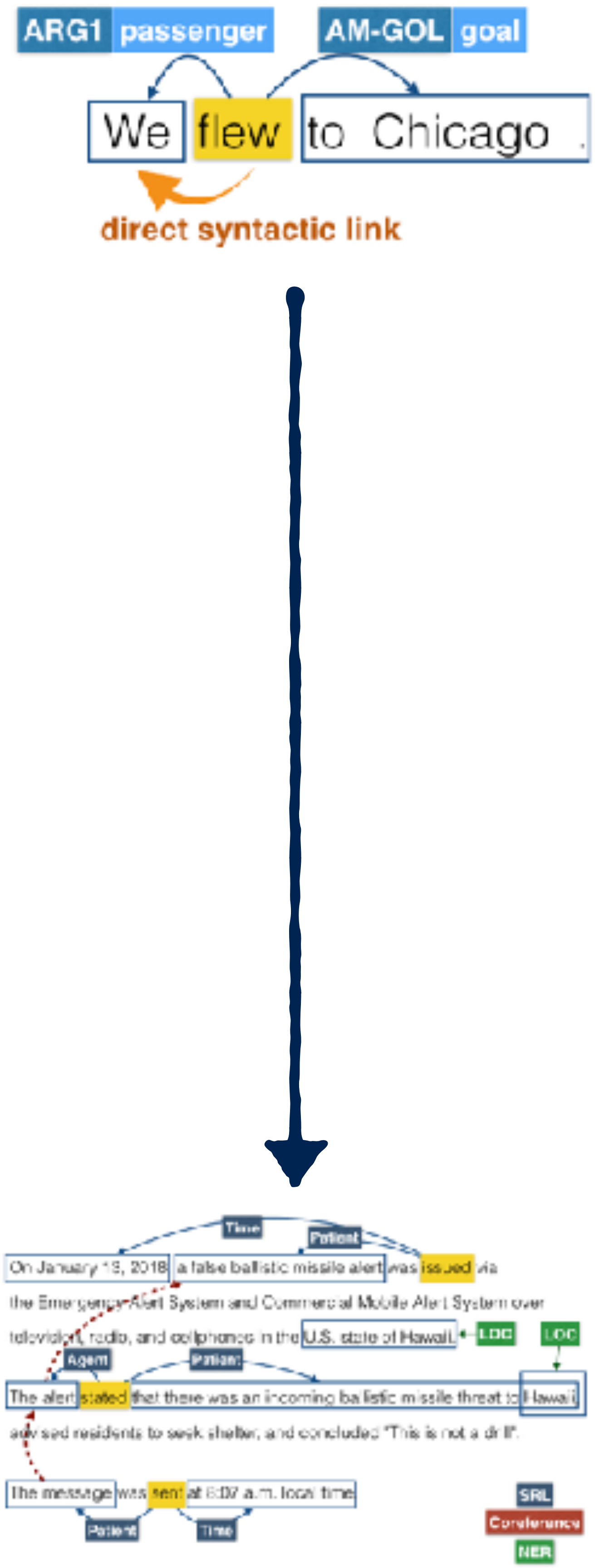
Contributions

- **End-to-end** prediction of SRL structure, without relying on NLP pipeline.
- **Almost 40% error reduction** over best pre-neural model despite being much simpler.



Contributions

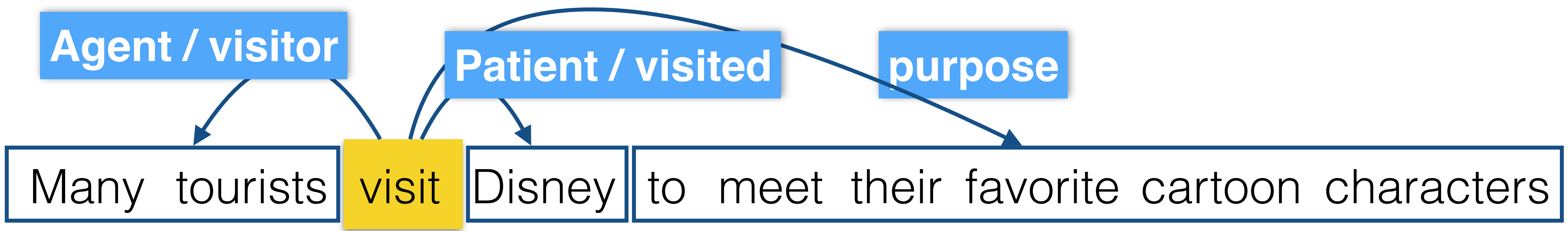
- **End-to-end** prediction of SRL structure, without relying on NLP pipeline.
- **Almost 40% error reduction** over best pre-neural model despite being much simpler.
- First end-to-end result for **jointly predicting predicates and argument spans**.



Contributions

- **End-to-end** prediction of SRL structure, without relying on NLP pipeline.
- **Almost 40% error reduction** over best pre-neural model despite being much simpler.
- First end-to-end result for **jointly predicting predicates and argument spans**.
- Joint modeling for a variety of span-based tasks, opens up opportunities for **full-text understanding**.

Semantic Role Labeling (SRL)



Predicate

visit

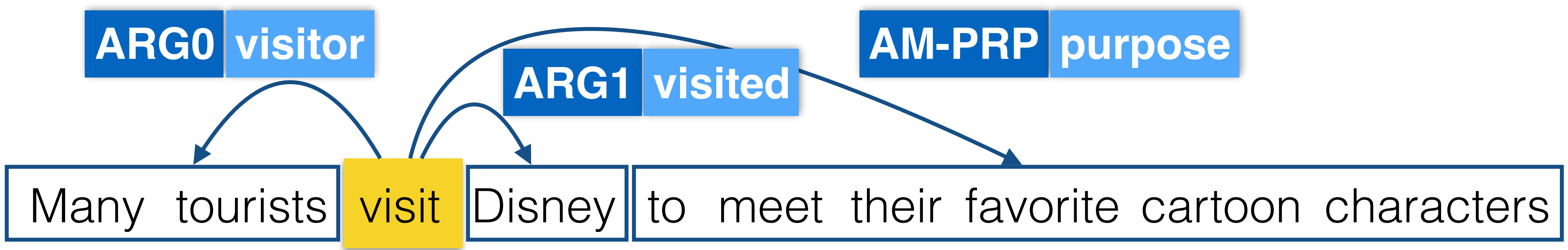
Arguments

Who is the **visitor**: [Many tourists]

What is **visited**: [Disney]

What **purpose**: [to meet … characters]

Semantic Role Labeling (SRL)



Predicate

visit

Arguments

ARG0: [Many tourists]

ARG1: [Disney]

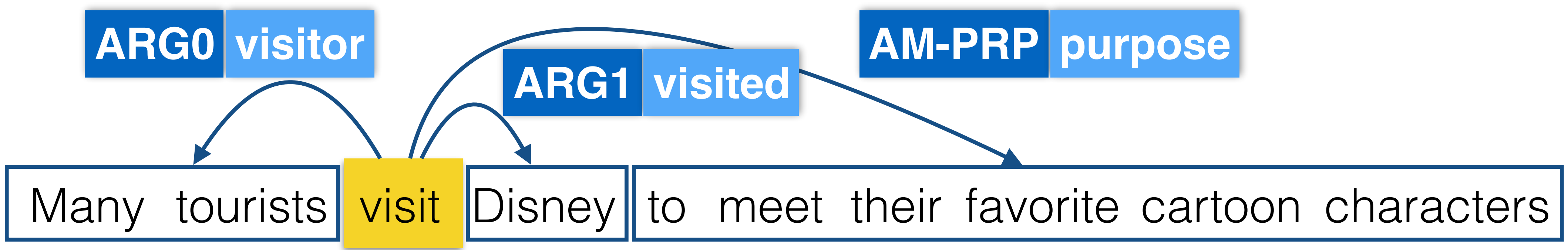
AM-PRP: [to meet … characters]

Frame: *visit.01*

role	description
ARG0	visitor
ARG1	visited

The Proposition Bank: An Annotated Corpus of Semantic Roles, Palmer et al., 2005

Semantic Role Labeling (SRL)



Predicate

visit

Arguments

ARG0: [Many tourists]

ARG1: [Disney]

AM-PRP: [to meet … characters]

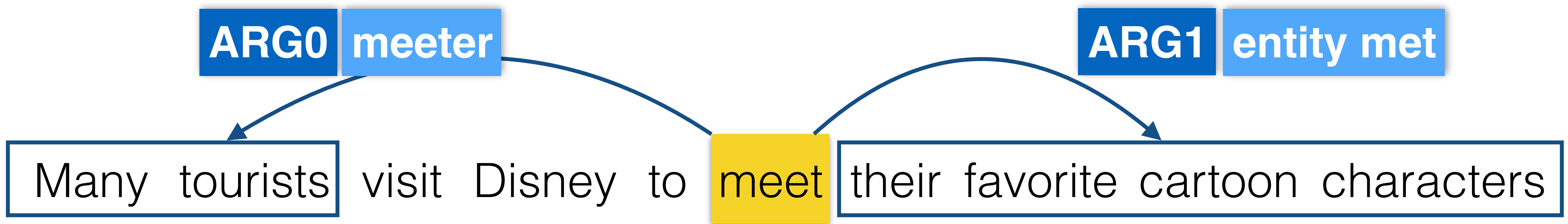
Frame: *visit.01*

role	description
ARG0	visitor
ARG1	visited

Core arguments: Verb-specific roles (A0-A5)
Adjuncts: Arg-modifier (AM-) roles shared across verbs

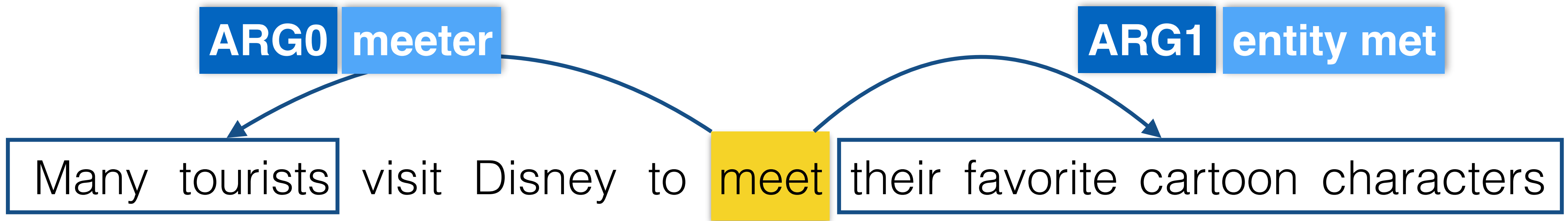
The Proposition Bank: An Annotated Corpus of Semantic Roles, Palmer et al., 2005

SRL Task: Given (gold) predicate, predict arguments



Predicate	Arguments
visit	**ARG0**: [Many tourists]
	ARG1: [Disney]
	AM-PRP: [to meet their favorite cartoon characters]
meet	**ARG0**: [Many tourists]
	ARG1: [their favorite cartoon characters]

SRL Task: Given (gold) predicate, predict arguments



Predicate	Arguments
visit	**ARG0**: [Many tourists] **ARG1**: [Disney] **AM-PRP**: [to meet their favorite cartoon characters]
meet	**ARG0**: [Many tourists] **ARG1**: [their favorite cartoon characters]

Most Span-based SRL Tasks: given gold predicates, predict the argument spans and labels.

Outline

Predicting SRL with Deep BiLSTMs

— DeepSRL (He et al., 2017)

An End-to-End, Span-based SRL Model

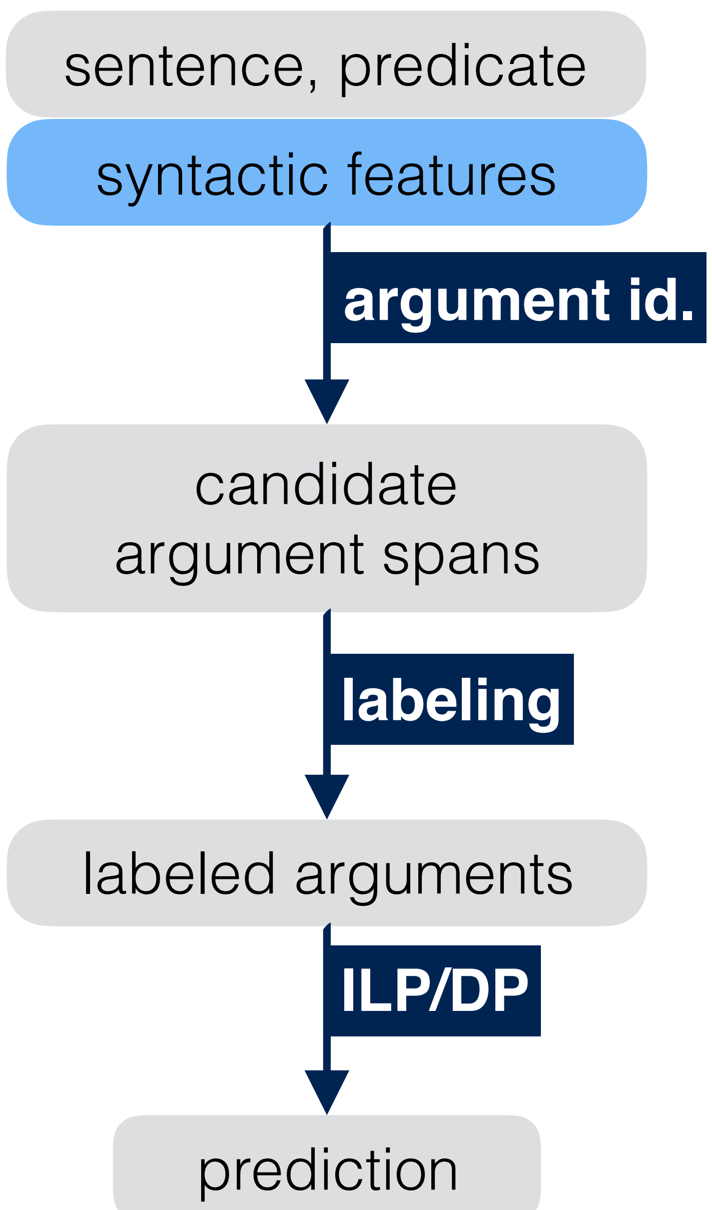
— Labeled Span Graph Network (He et al., 2018)

Towards Unified and Full-text Semantic Analysis

— Multi-task learning with LSGN; ScienceIE (Luan et al., 2018)

SRL Systems: Pipelined vs. BIO-based

Pipeline Systems



Punyakanok et al., 2008
Täckström et al., 2015
FitzGerald et al., 2015

SRL Systems: Pipelined vs. BIO-based

Pipeline Systems

sentence, predicate

syntactic features

argument id.

candidate argument spans

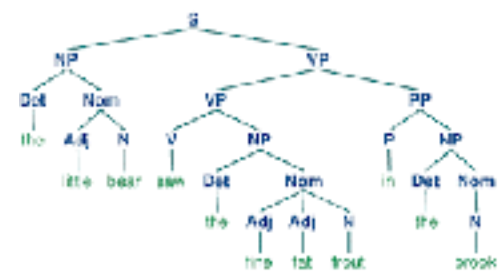
labeling

labeled arguments

ILP/DP

prediction

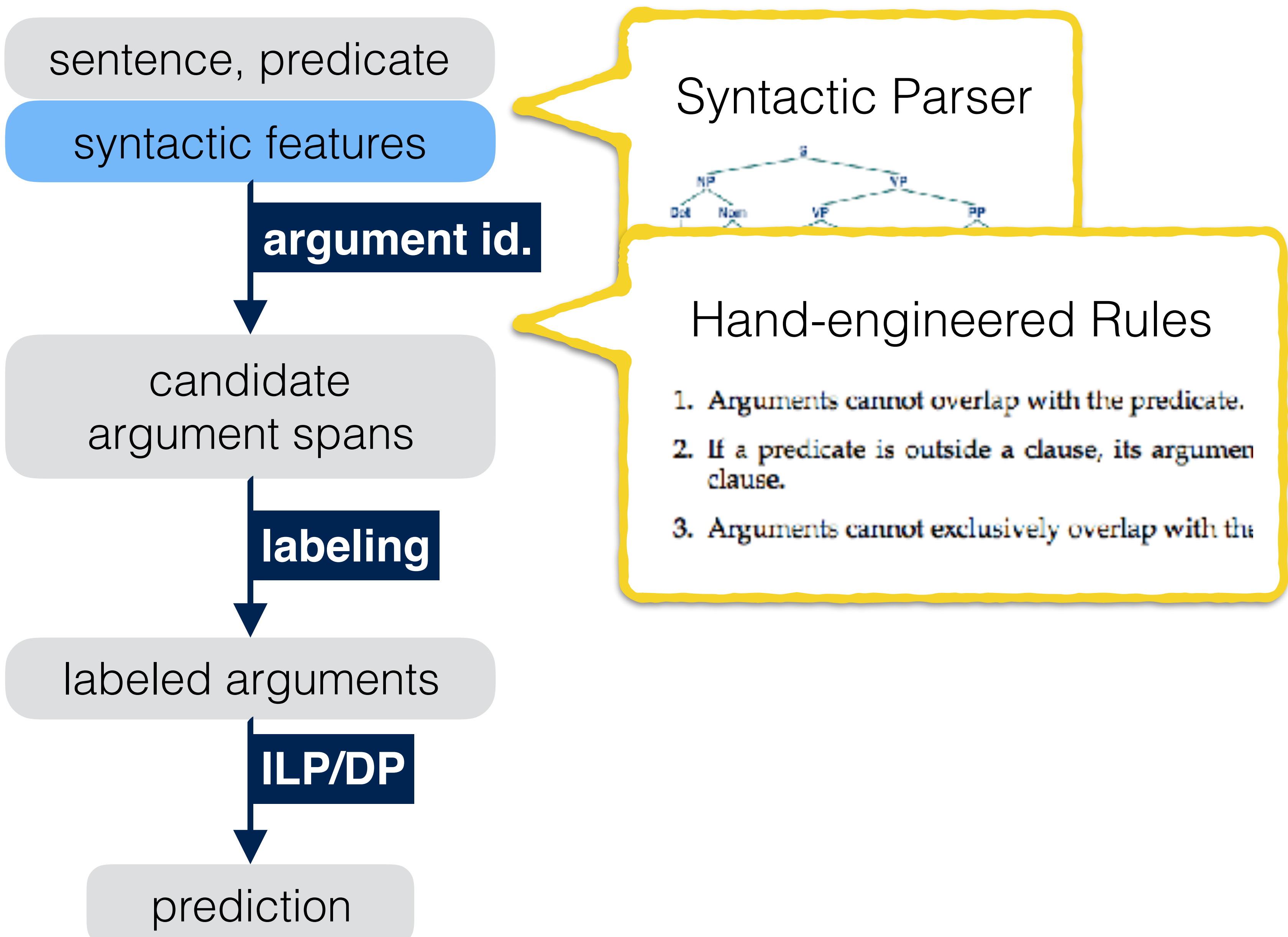
Syntactic Parser



Punyakanok et al., 2008
Täckström et al., 2015
FitzGerald et al., 2015

SRL Systems: Pipelined vs. BIO-based

Pipeline Systems



Punyakanok et al., 2008
Täckström et al., 2015
FitzGerald et al., 2015

SRL Systems: Pipelined vs. BIO-based

Pipeline Systems

sentence, predicate

syntactic features

argument id.

candidate argument spans

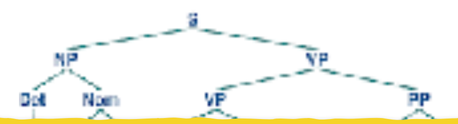
labeling

labeled arguments

ILP/DP

prediction

Syntactic Parser



Hand-engineered Rules

1. Arguments cannot overlap with the predicate.
2. If a predicate is outside a clause, its argumen clause.

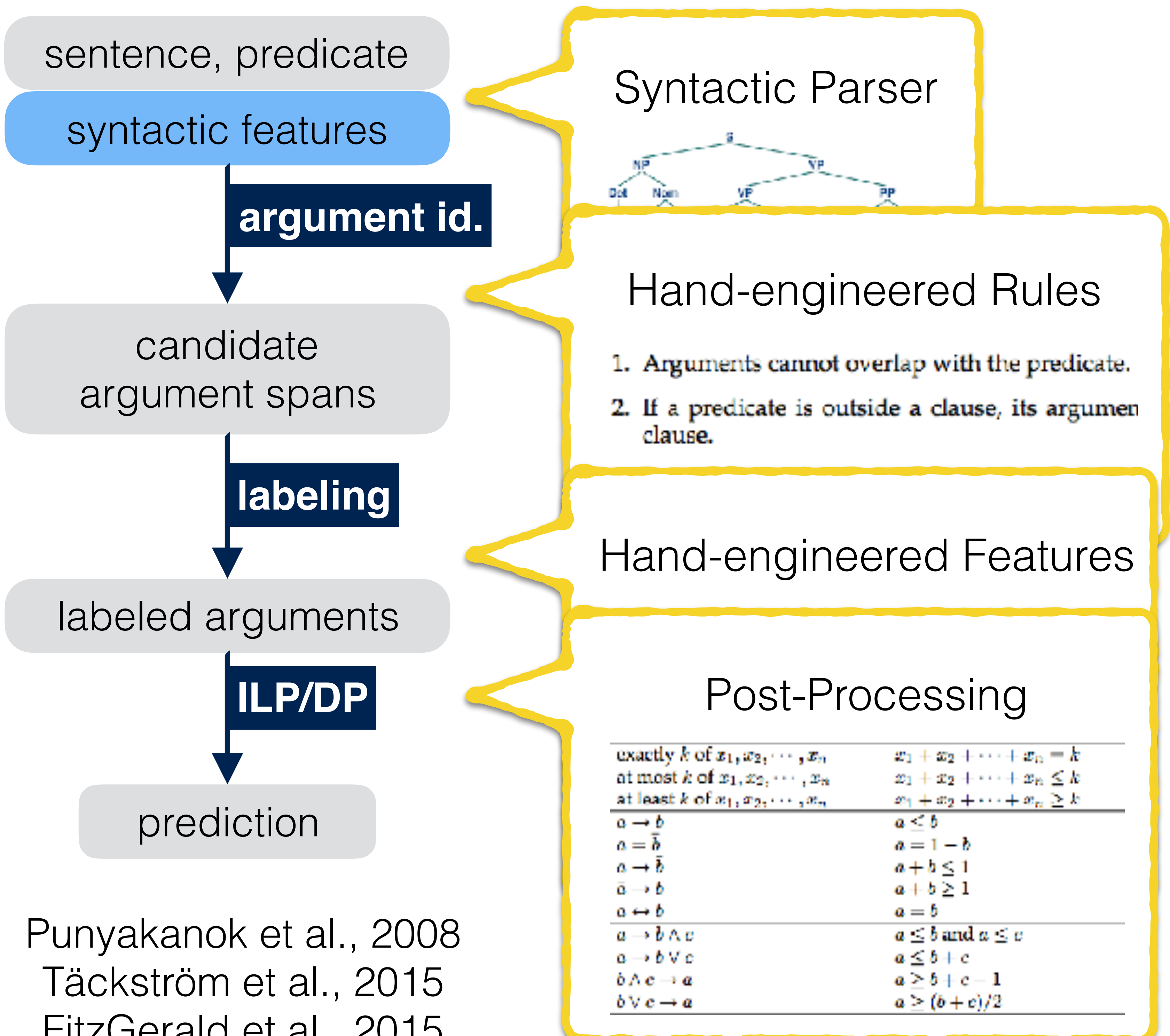
Hand-engineered Features

- starting word of s
- tag of the starting word of s
- ending word of s
- tag of the ending word of s
- head word of s
- tag of the head word of s
- bag of words in s
- bag of tags in s
- a bias feature
- cluster of s's head
- dependency path between s's head and t
- the set of dependency labels of t's children
- dependency path conjoined with the tag of s's head
- dep. path conjoined with the cluster of s's head
- *position* of s w.r.t. t (*before, after, overlap or same*)

Punyakanok et al., 2008
Täckström et al., 2015
FitzGerald et al., 2015

SRL Systems: Pipelined vs. BIO-based

Pipeline Systems



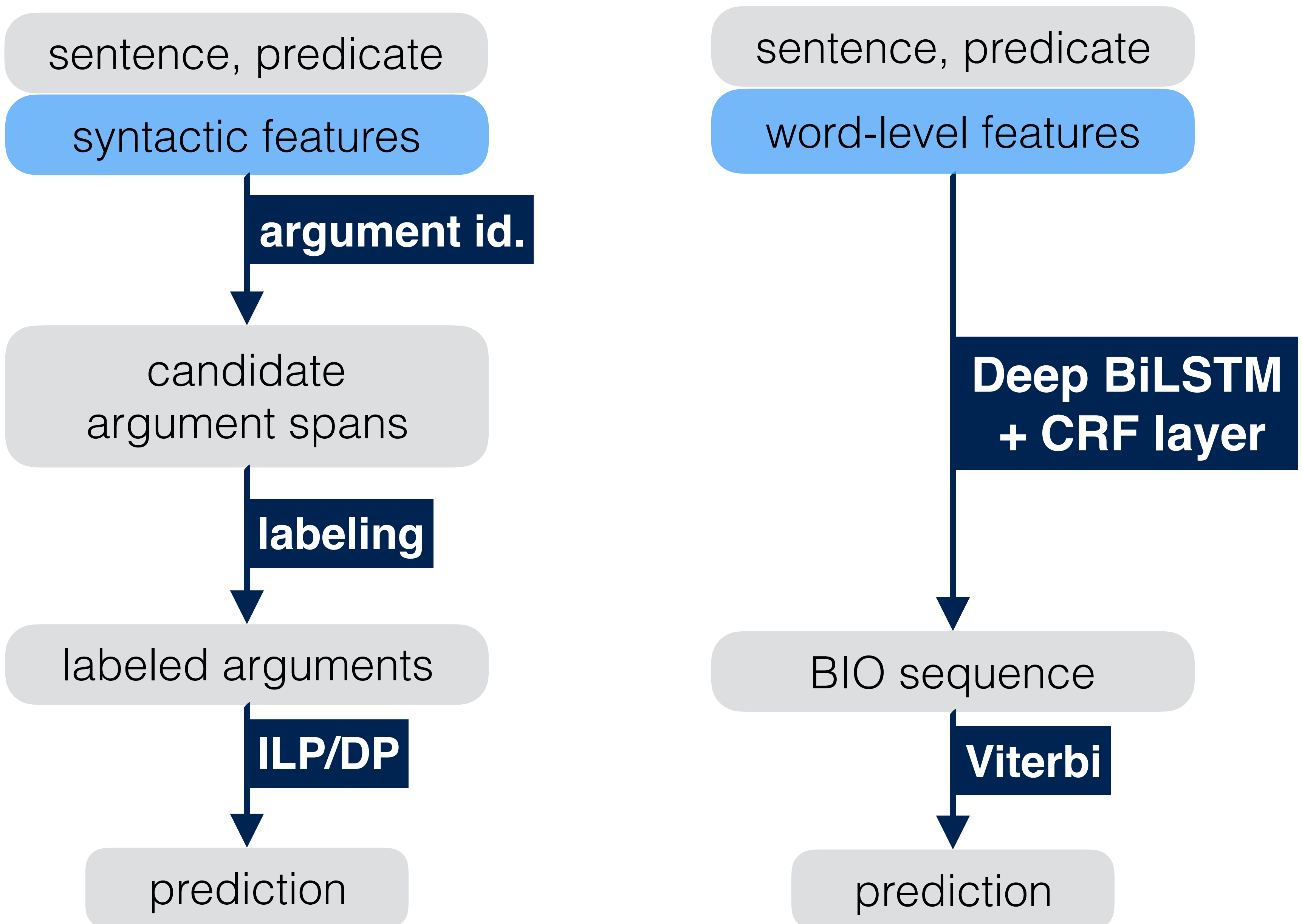
exactly k of $x_1, x_2, \cdots, x_n$	$x_1 + x_2 + \cdots + x_n = k$
at most k of $x_1, x_2, \cdots, x_n$	$x_1 + x_2 + \cdots + x_n \leq k$
at least k of $x_1, x_2, \cdots, x_n$	$x_1 + x_2 + \cdots + x_n \geq k$
$a \rightarrow b$	$a \leq b$
$a = \bar{b}$	$a = 1 - b$
$a \rightarrow \bar{b}$	$a + b \leq 1$
$\bar{a} \rightarrow b$	$a + b \geq 1$
$a \leftrightarrow b$	$a = b$
$a \rightarrow b \wedge c$	$a \leq b$ and $a \leq c$
$a \rightarrow b \vee c$	$a \leq b + c$
$b \wedge c \rightarrow a$	$a \geq b + c - 1$
$b \vee c \rightarrow a$	$a \geq (b + c)/2$

Punyakanok et al., 2008
Täckström et al., 2015
FitzGerald et al., 2015

SRL Systems: Pipelined vs. BIO-based

Pipeline Systems

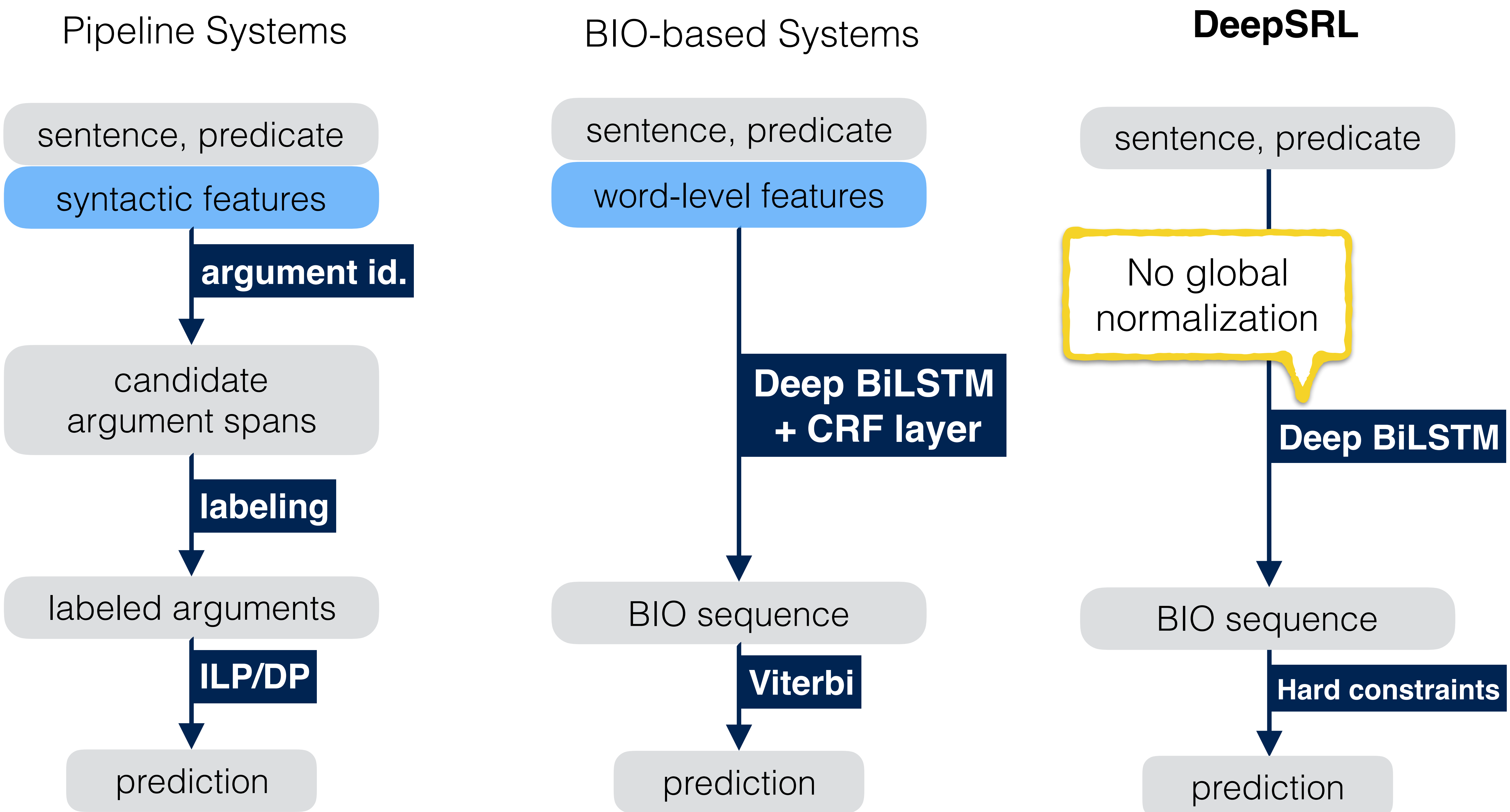
BIO-based Systems



Punyakanok et al., 2008
Täckström et al., 2015
FitzGerald et al., 2015

Collobert et al., 2011
Zhou and Xu, 2015
Wang et. al, 2015

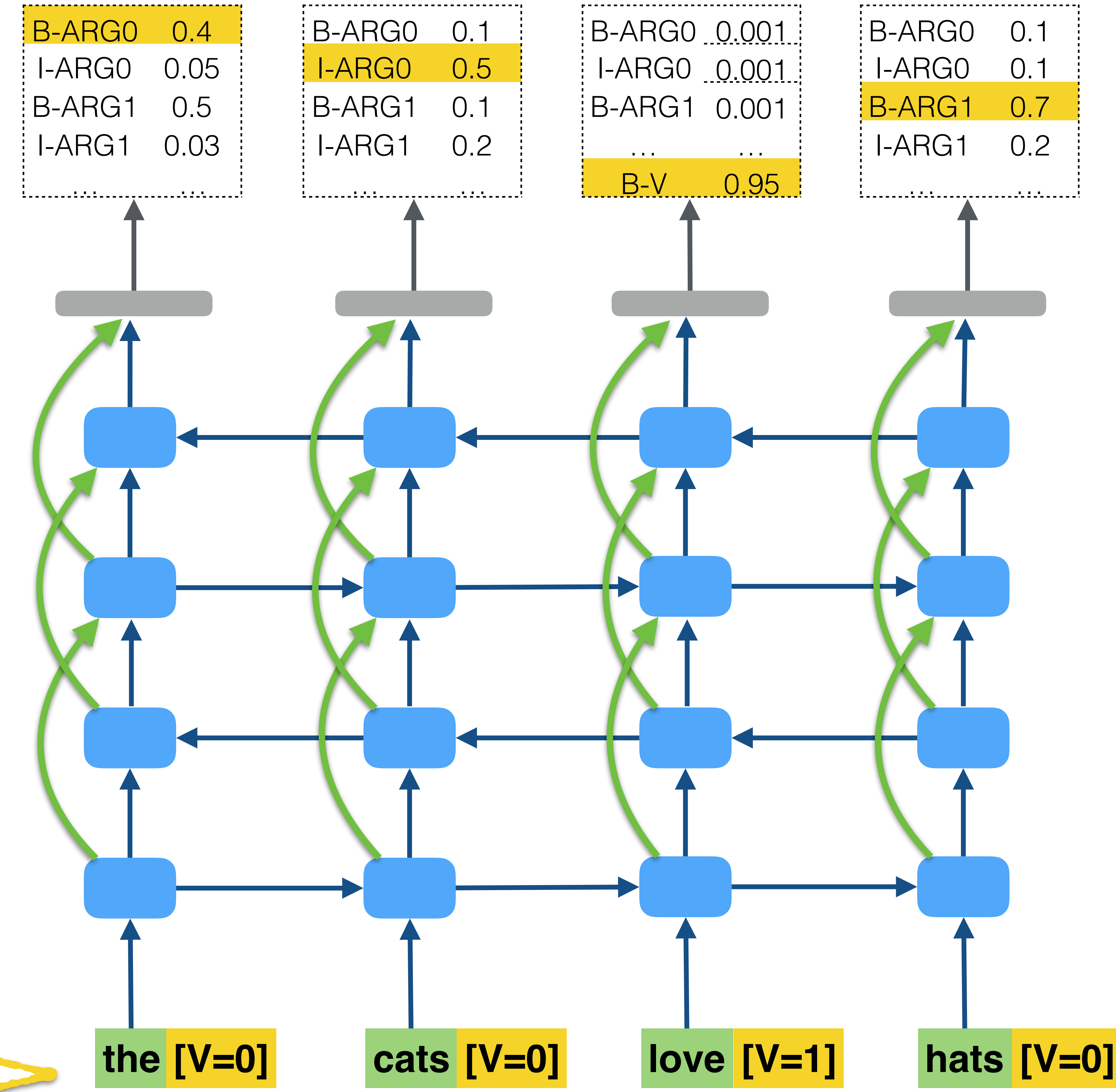
SRL Systems: Pipelined vs. BIO-based



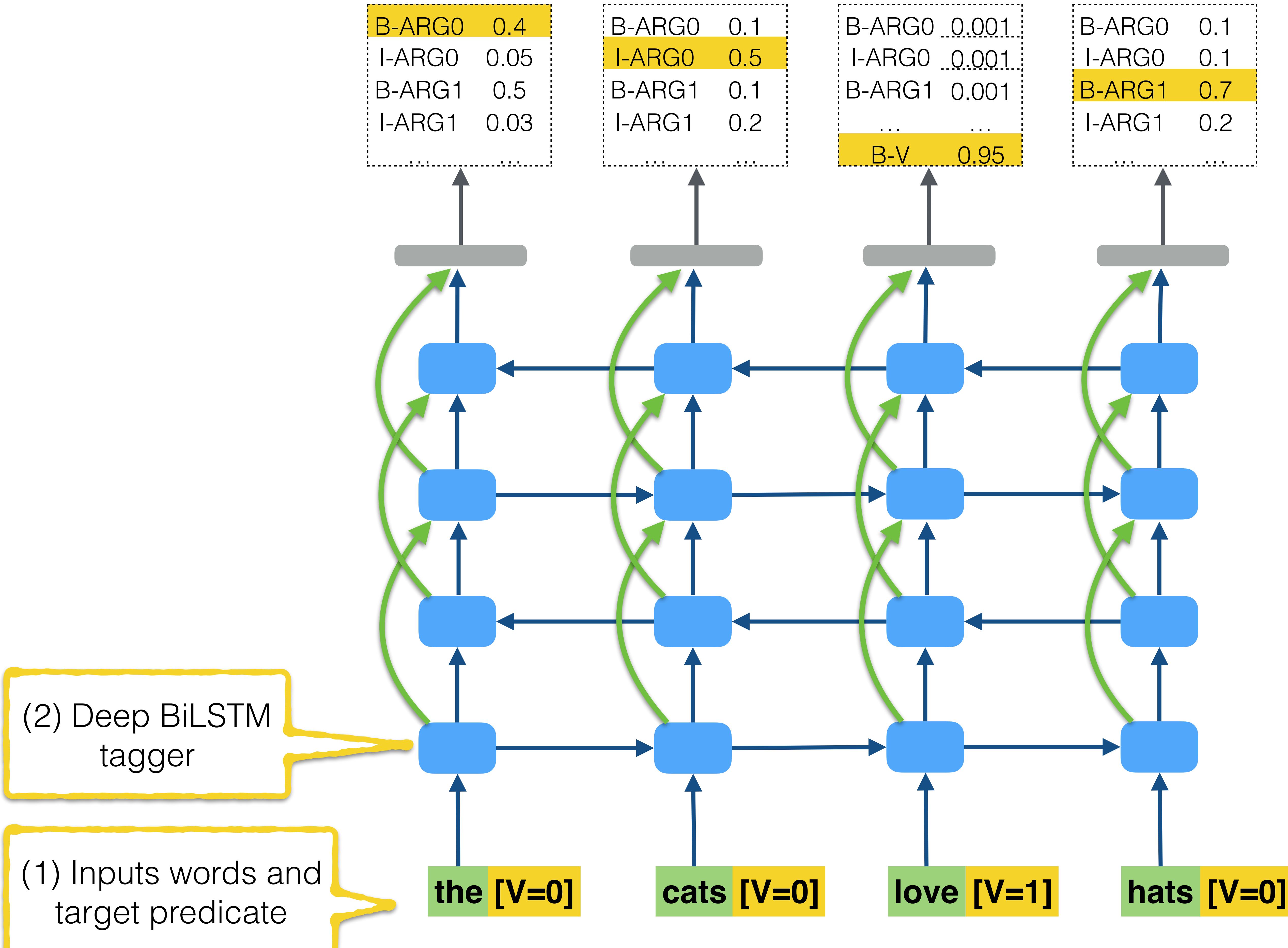
Punyakanok et al., 2008
Täckström et al., 2015
FitzGerald et al., 2015

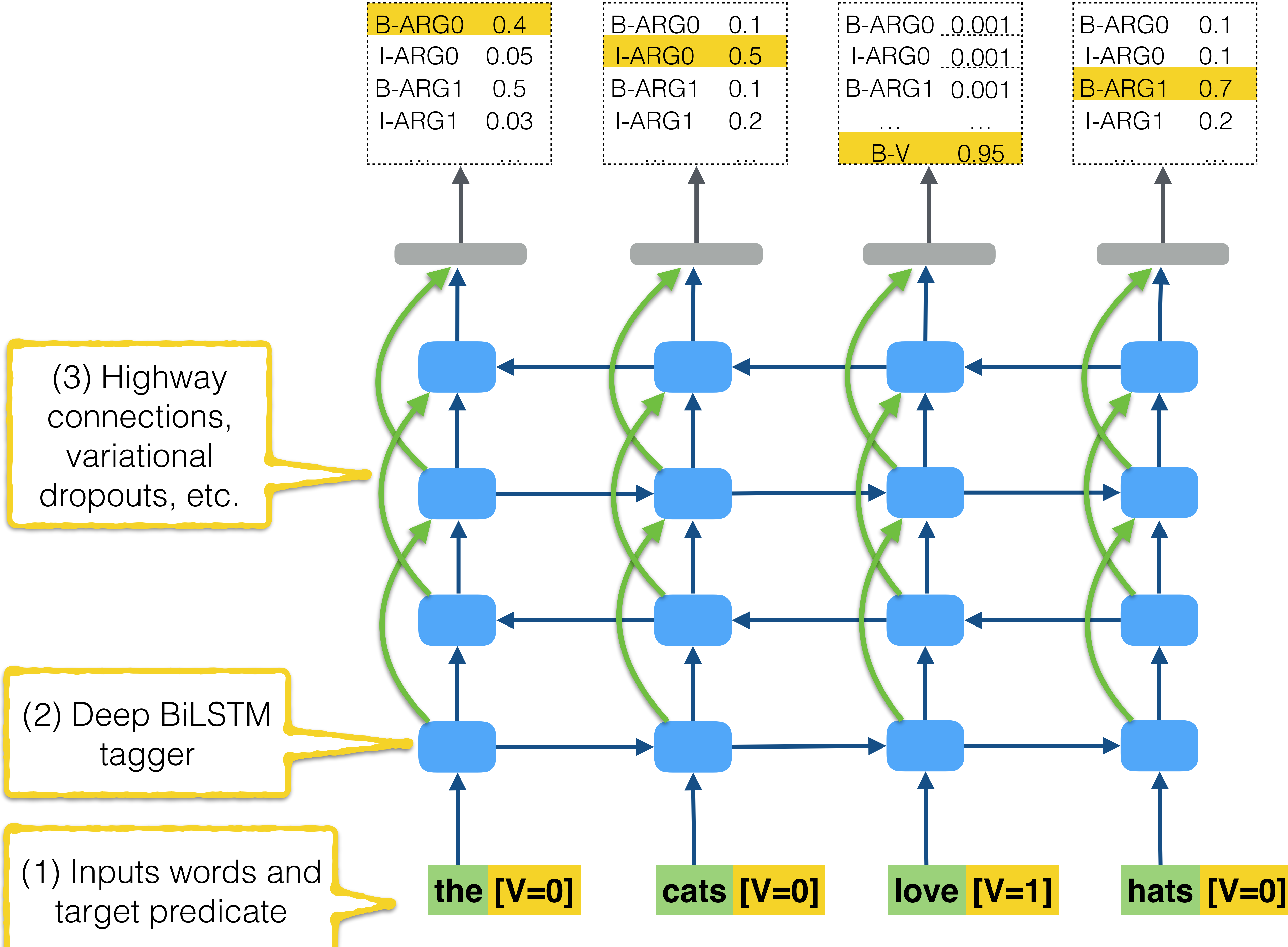
Collobert et al., 2011
Zhou and Xu, 2015
Wang et. al, 2015

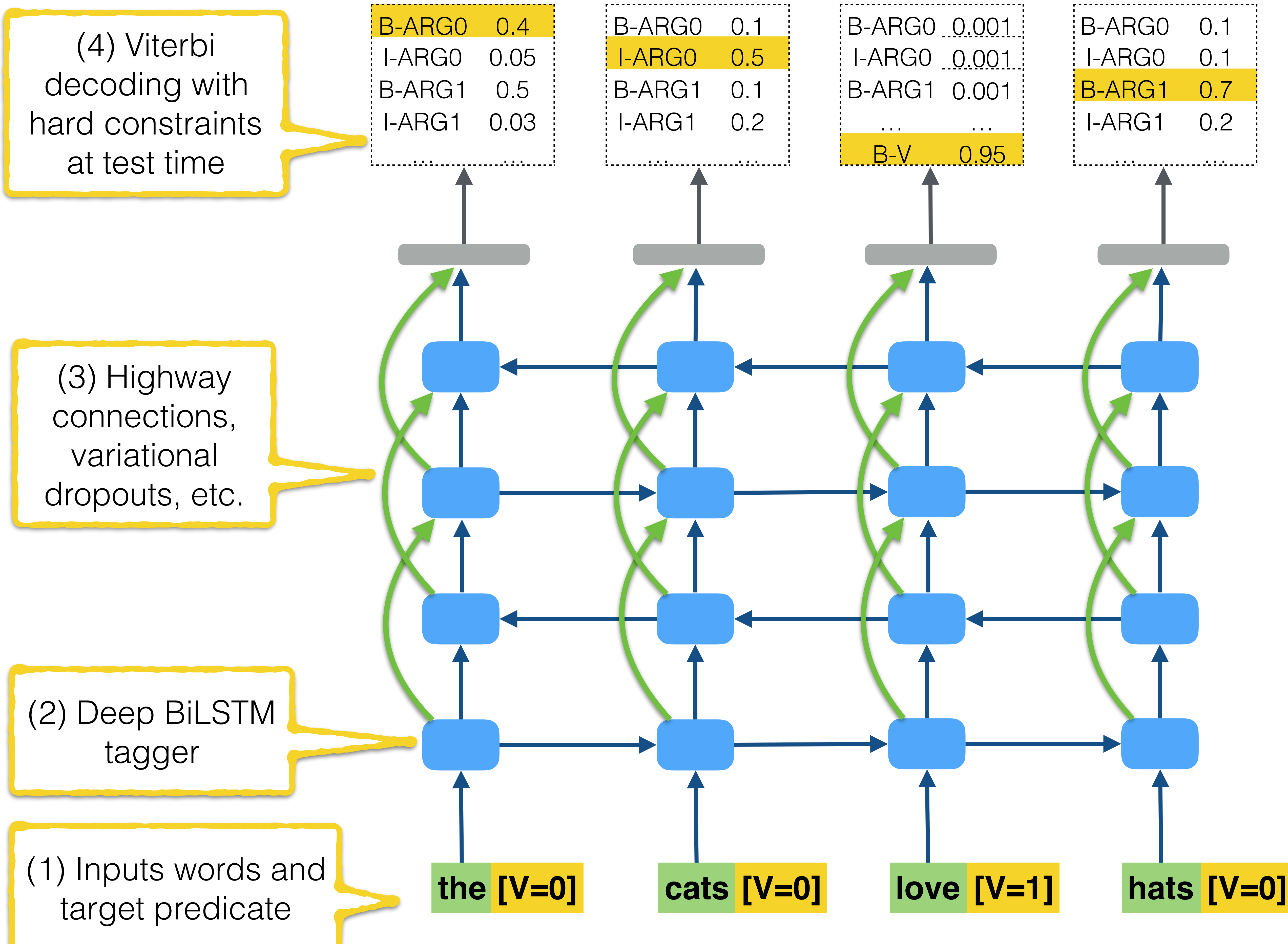
He et al., 2017

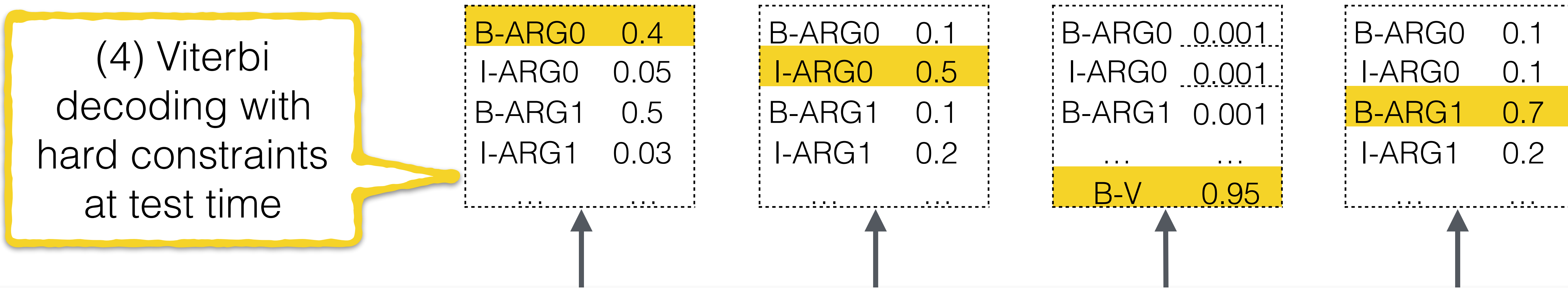


(1) Inputs words and target predicate


B-ARG0 0.4
I-ARG0 0.05
B-ARG1 0.5
I-ARG1 0.03
... ...
B-ARG0 0.1
I-ARG0 0.5
B-ARG1 0.1
I-ARG1 0.2
... ...
B-ARG0 0.001
I-ARG0 0.001
B-ARG1 0.001
... ...
B-V 0.95
B-ARG0 0.1
I-ARG0 0.1
B-ARG1 0.7
I-ARG1 0.2
... ...
(2) Deep BiLSTM tagger
(1) Inputs words and target predicate
the [V=0]
cats [V=0]
love [V=1]
hats [V=0]


B-ARG0 0.4
I-ARG0 0.05
B-ARG1 0.5
I-ARG1 0.03
... ...
B-ARG0 0.1
I-ARG0 0.5
B-ARG1 0.1
I-ARG1 0.2
... ...
B-ARG0 0.001
I-ARG0 0.001
B-ARG1 0.001
... ...
B-V 0.95
B-ARG0 0.1
I-ARG0 0.1
B-ARG1 0.7
I-ARG1 0.2
... ...
(3) Highway connections, variational dropouts, etc.
(2) Deep BiLSTM tagger
(1) Inputs words and target predicate
the [V=0]
cats [V=0]
love [V=1]
hats [V=0]


(4) Viterbi decoding with hard constraints at test time
(3) Highway connections, variational dropouts, etc.
(2) Deep BiLSTM tagger
(1) Inputs words and target predicate
B-ARG0 0.4
I-ARG0 0.05
B-ARG1 0.5
I-ARG1 0.03
B-ARG0 0.1
I-ARG0 0.5
B-ARG1 0.1
I-ARG1 0.2
B-ARG0 0.001
I-ARG0 0.001
B-ARG1 0.001
B-V 0.95
B-ARG0 0.1
I-ARG0 0.1
B-ARG1 0.7
I-ARG1 0.2
the [V=0]
cats [V=0]
love [V=1]
hats [V=0]



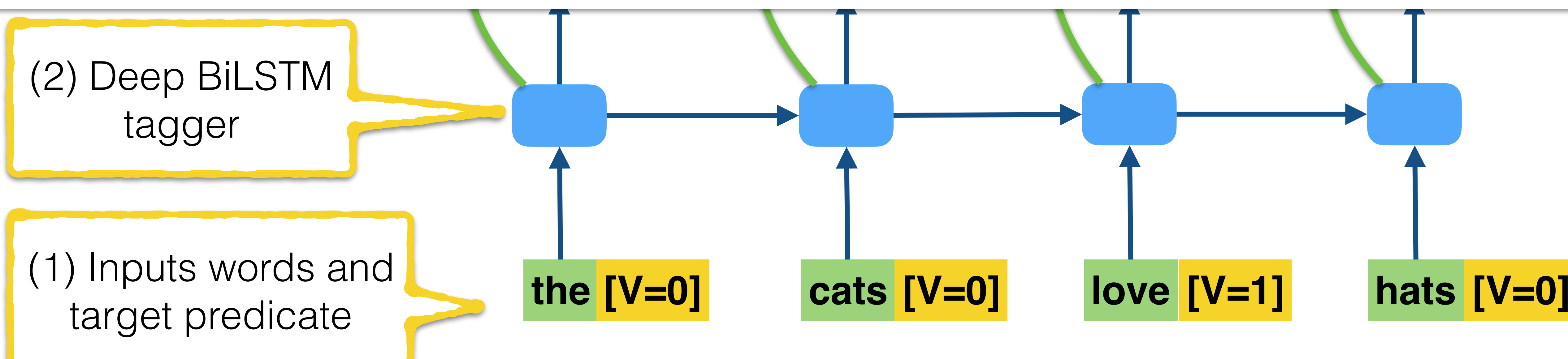
Strengths:

No syntactic preprocessing;

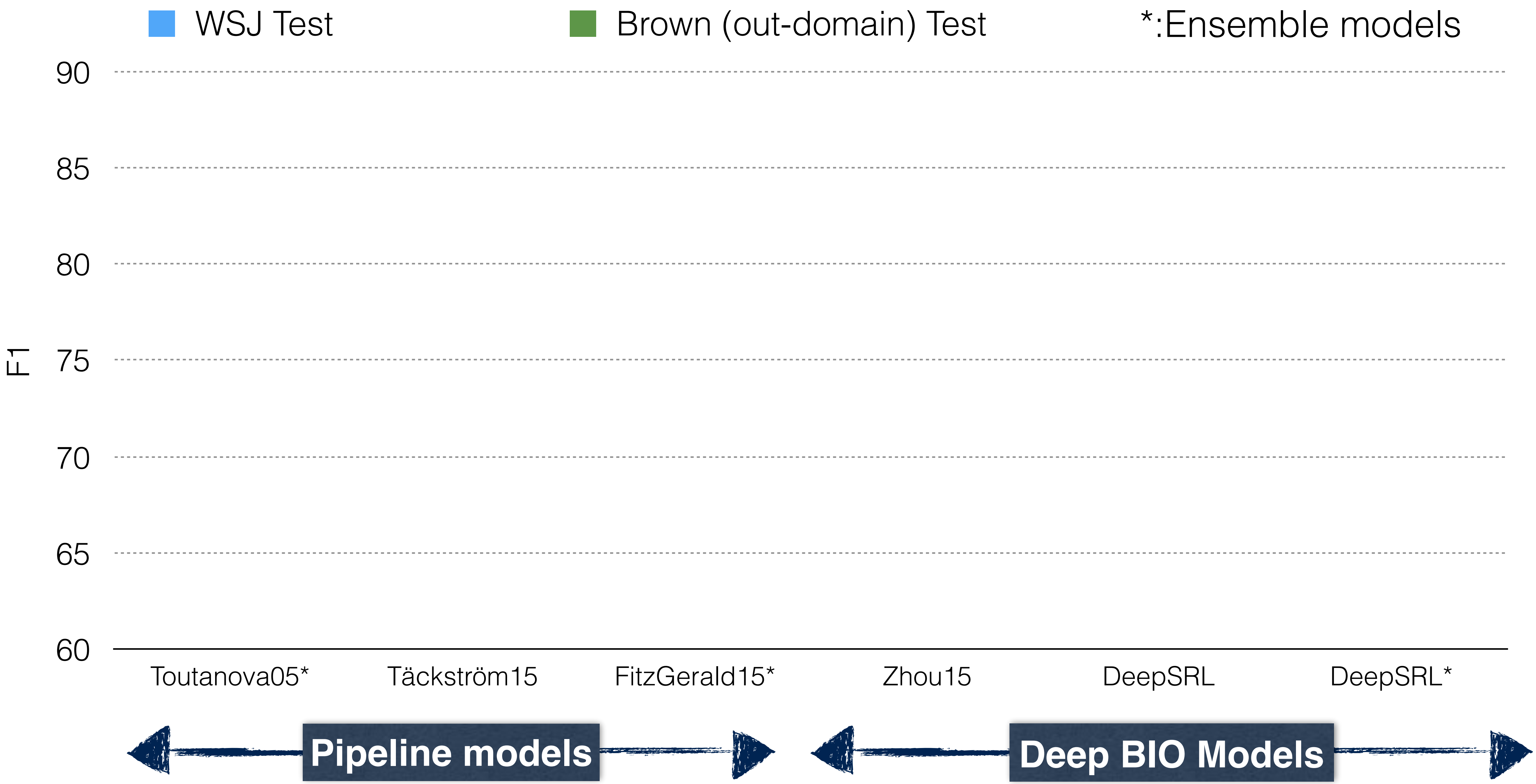
Easy to implement (can use off-the-shelf sequential tagger)

Limitations:

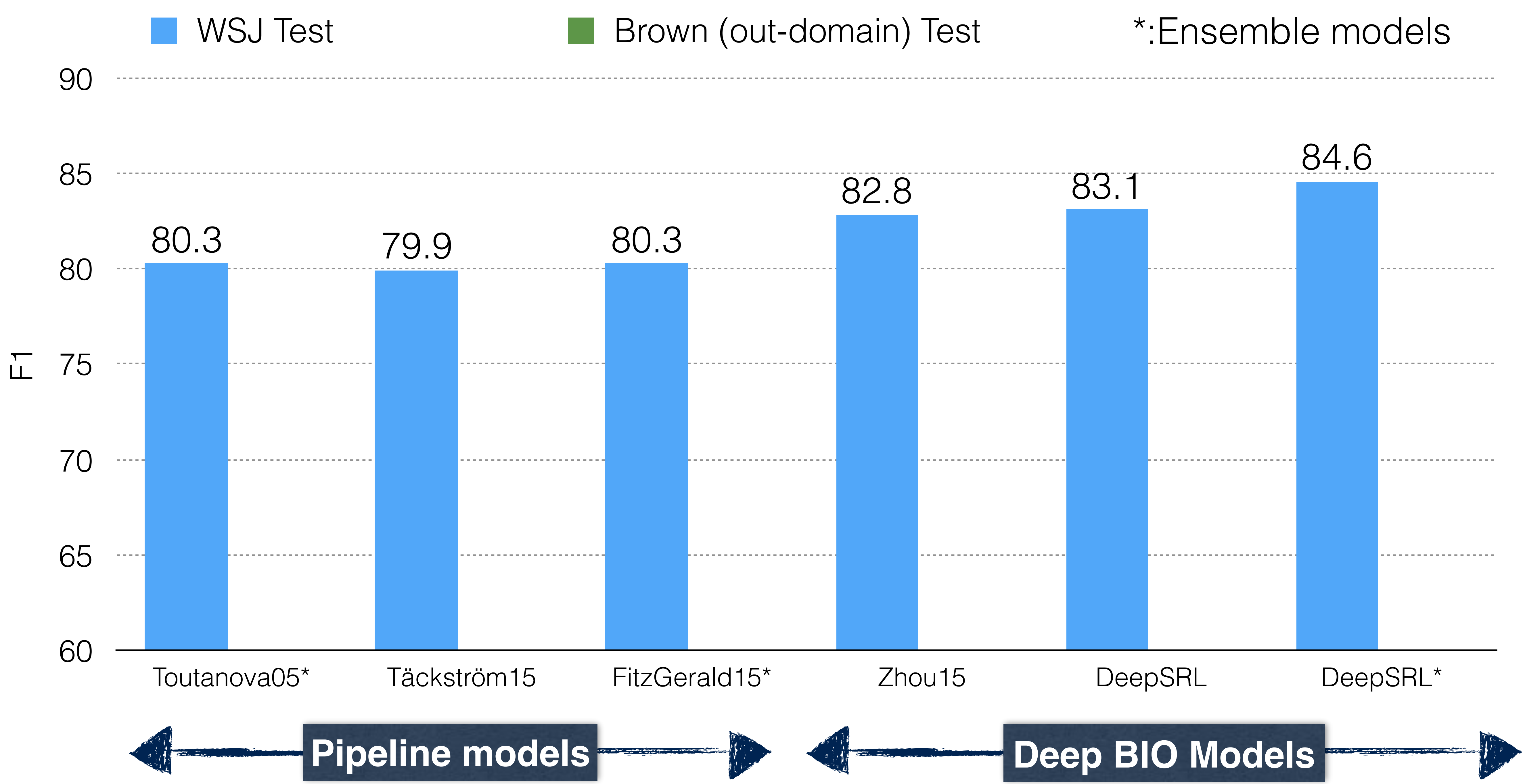
Needs to re-process the same sentence multiple times, if sentence has multiple predicates



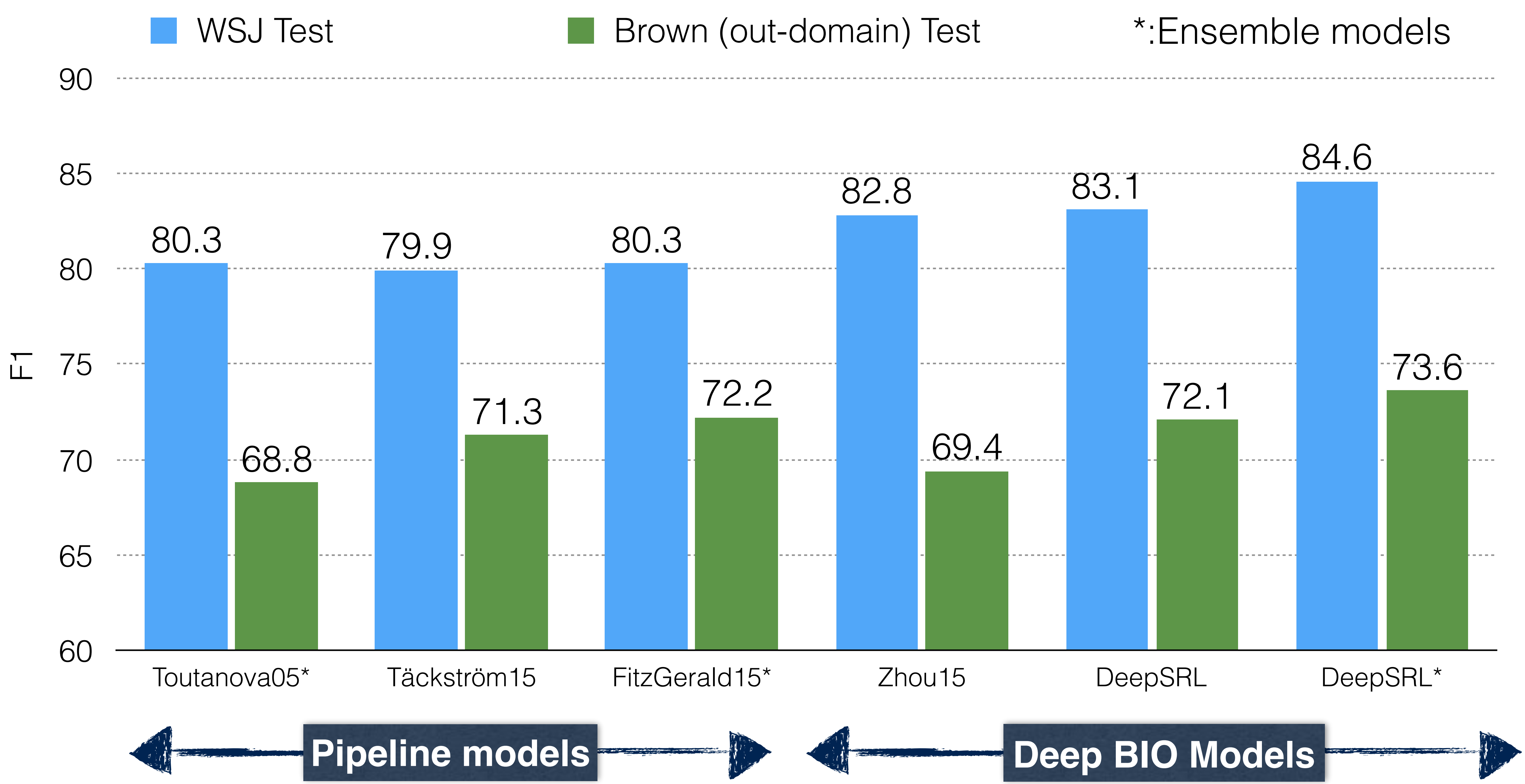
CoNLL 2005 (Original PropBank) Results



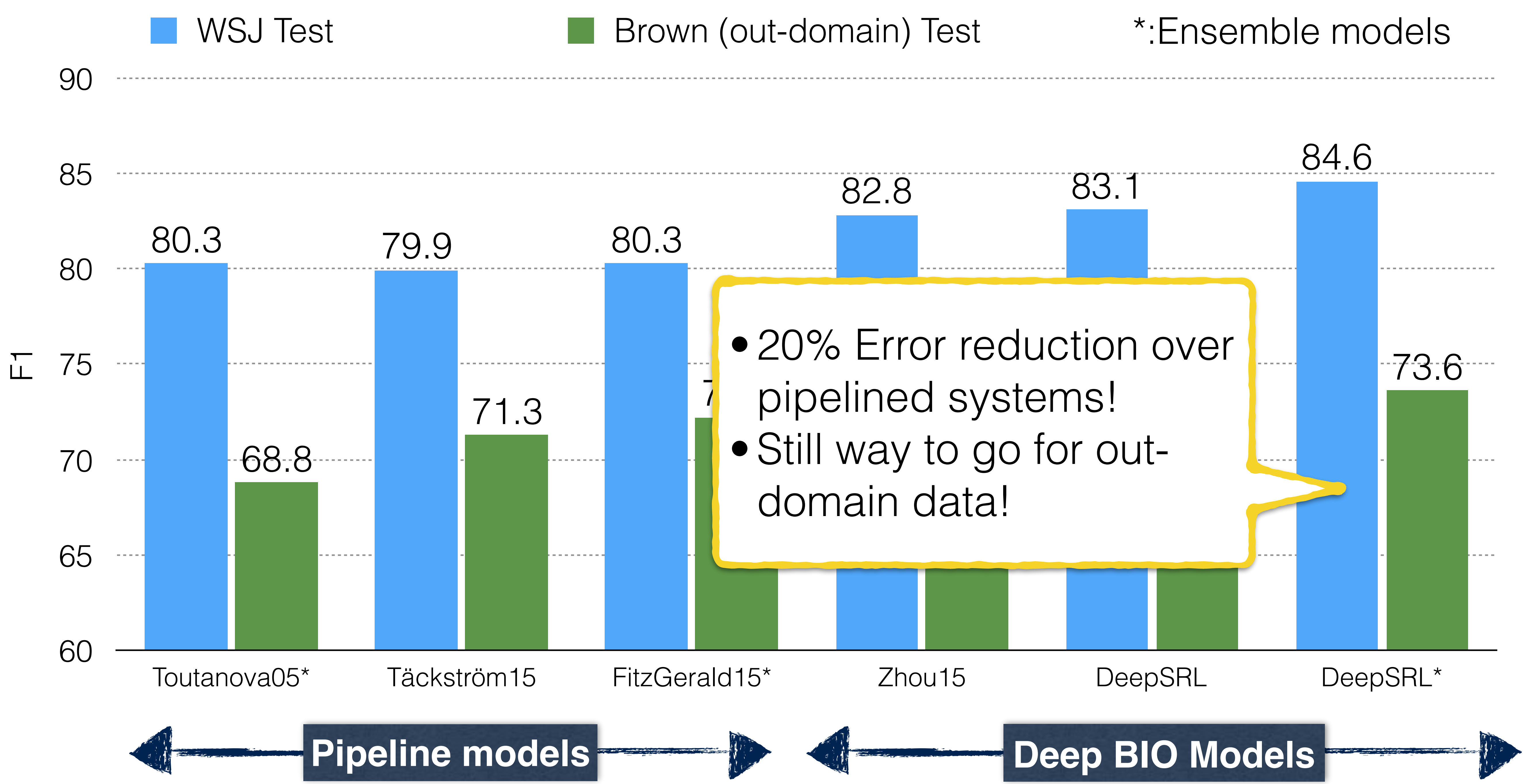
CoNLL 2005 (Original PropBank) Results



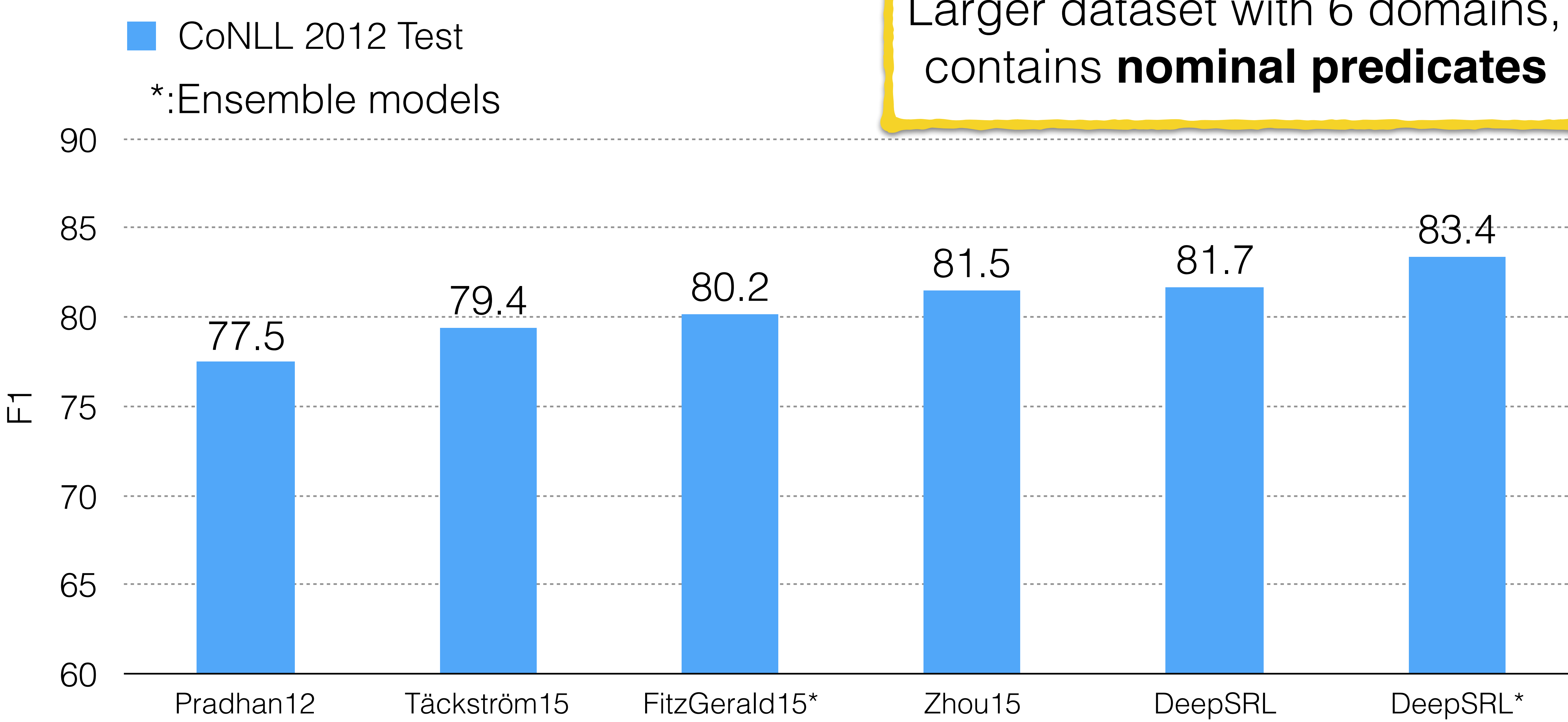
CoNLL 2005 (Original PropBank) Results



CoNLL 2005 (Original PropBank) Results

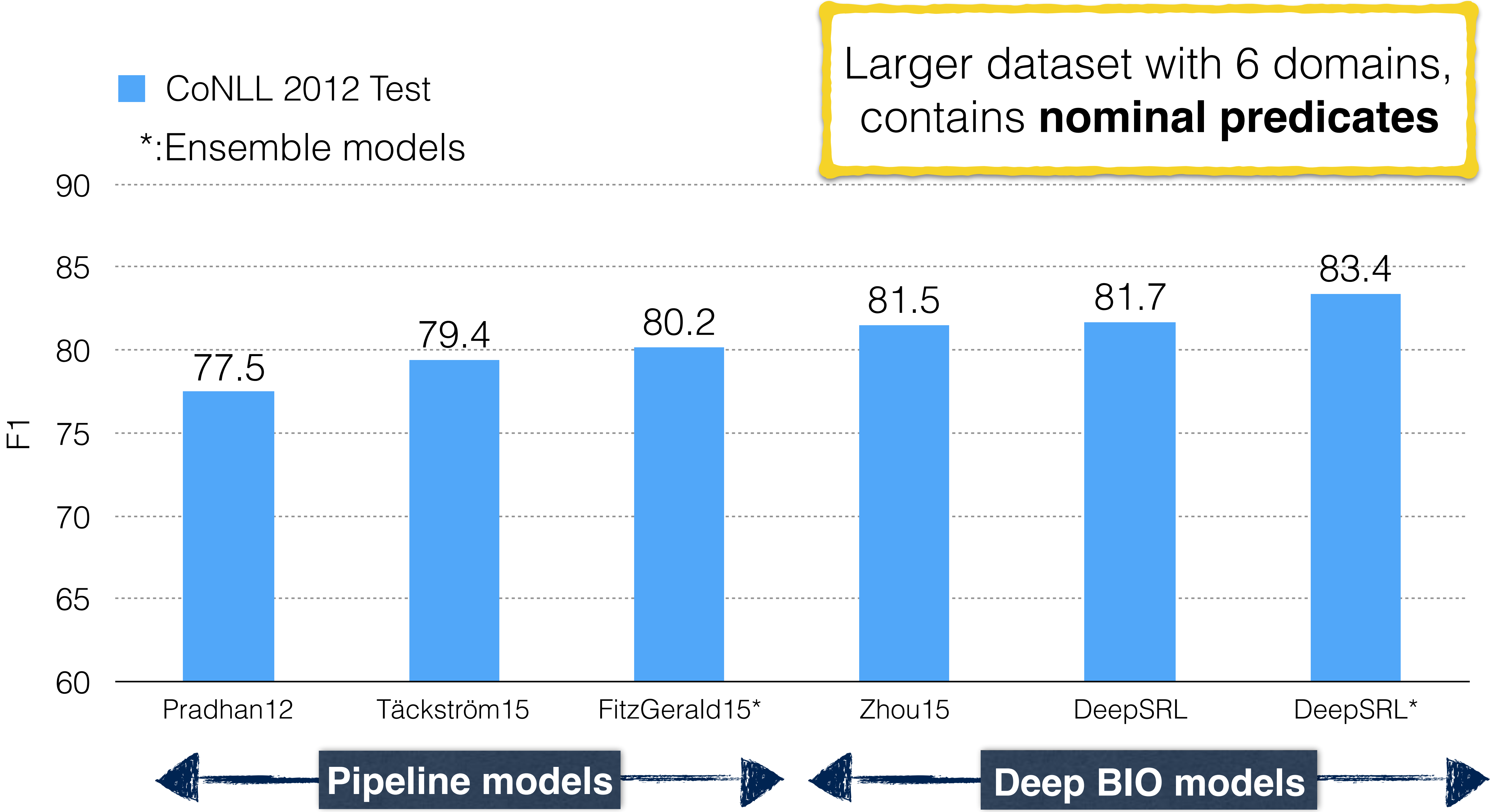


CoNLL 2012 (OntoNotes) Results

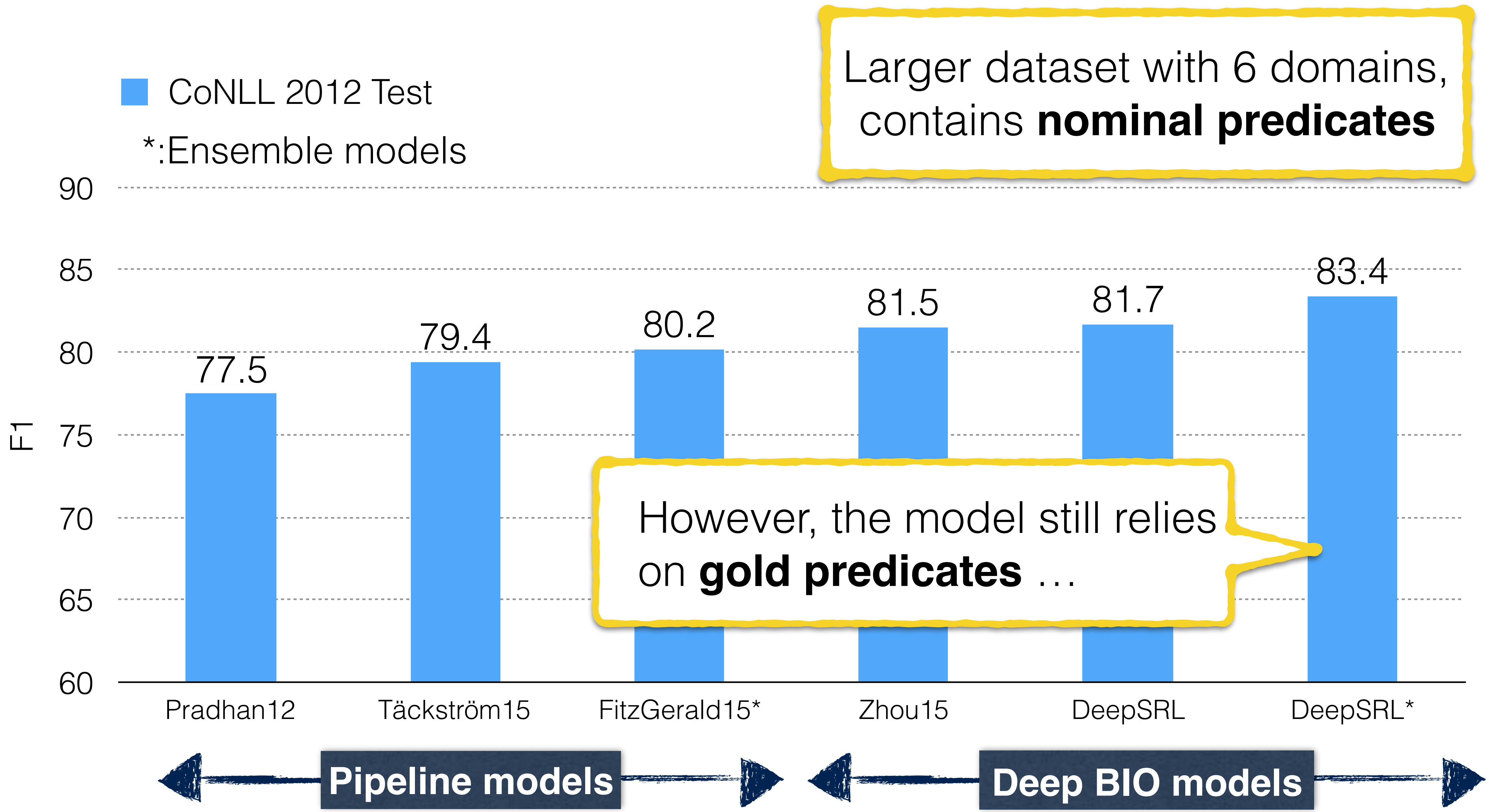


Larger dataset with 6 domains, contains **nominal predicates**

CoNLL 2012 (OntoNotes) Results

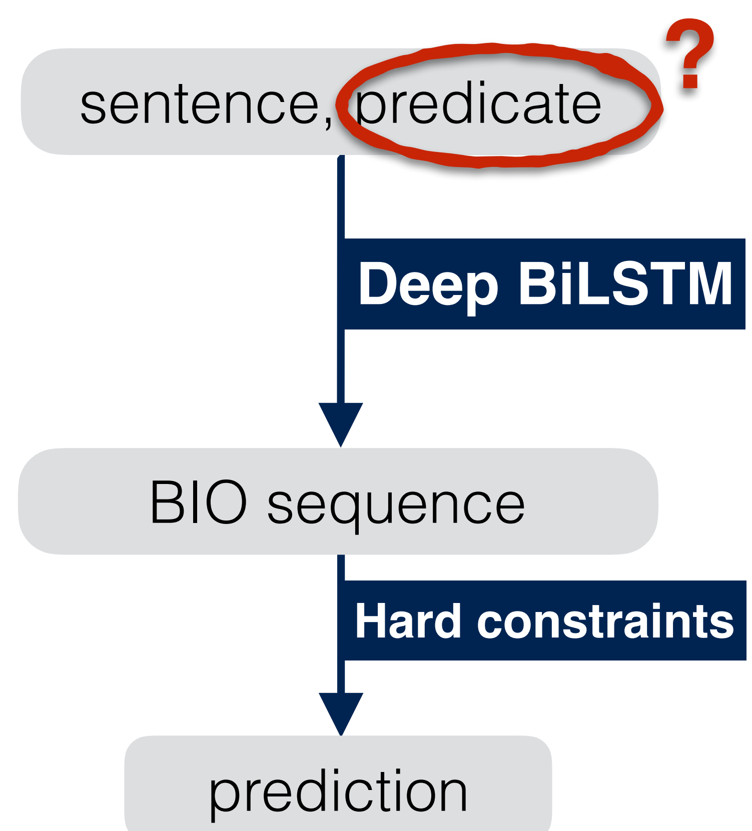


CoNLL 2012 (OntoNotes) Results



Real Scenario: No Gold Predicates!

End-to-end SRL:
Given sentence, predict all predicates as well as their arguments.

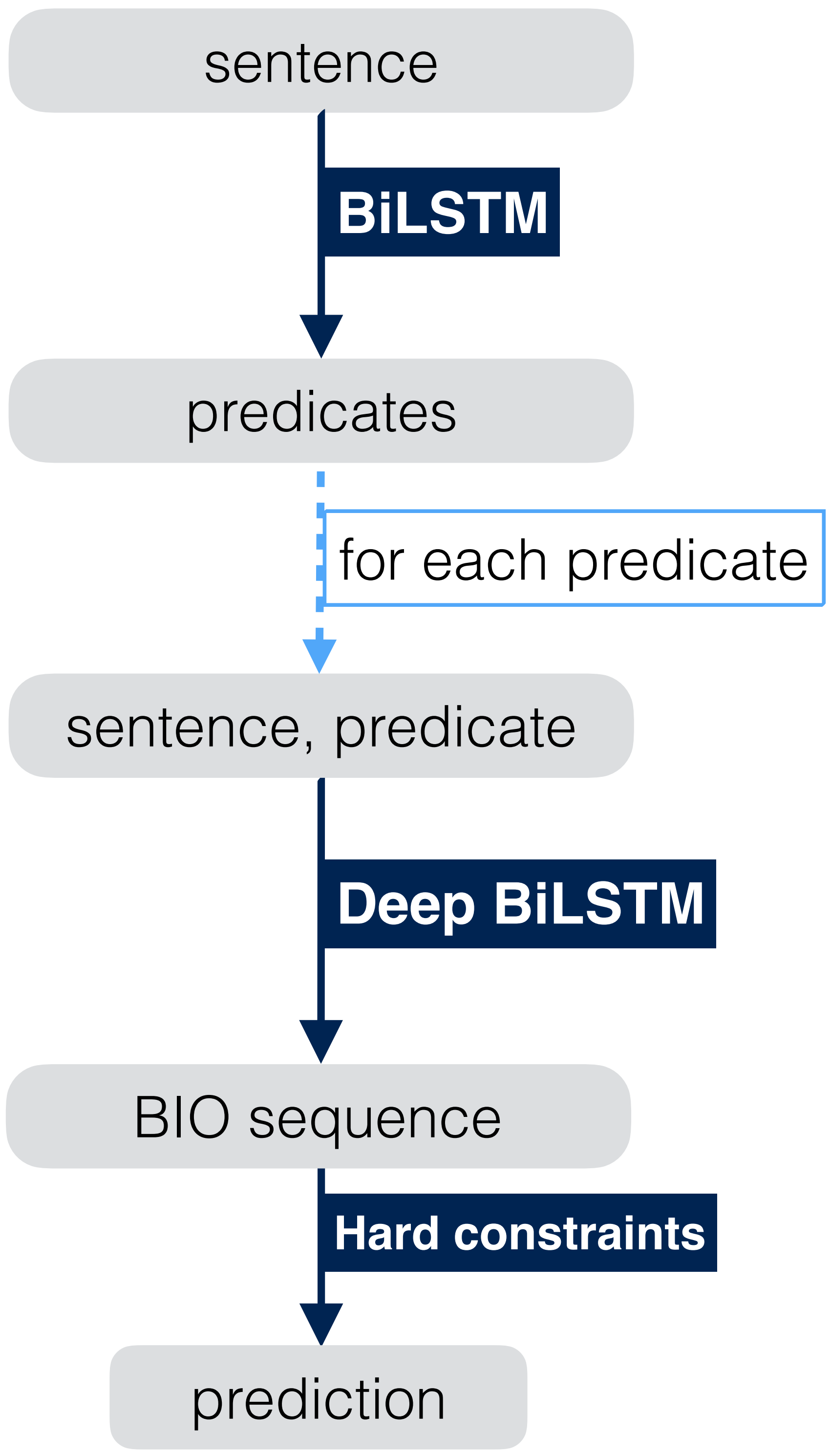


Real Scenario: No Gold Predicates!

End-to-end SRL:
Given sentence, predict all predicates as well as their arguments.

Pipelined approach:
Identify predicates first, then run the BIO tagger for each predicate.

(No way to recover from recall loss at predicate ID stage …)

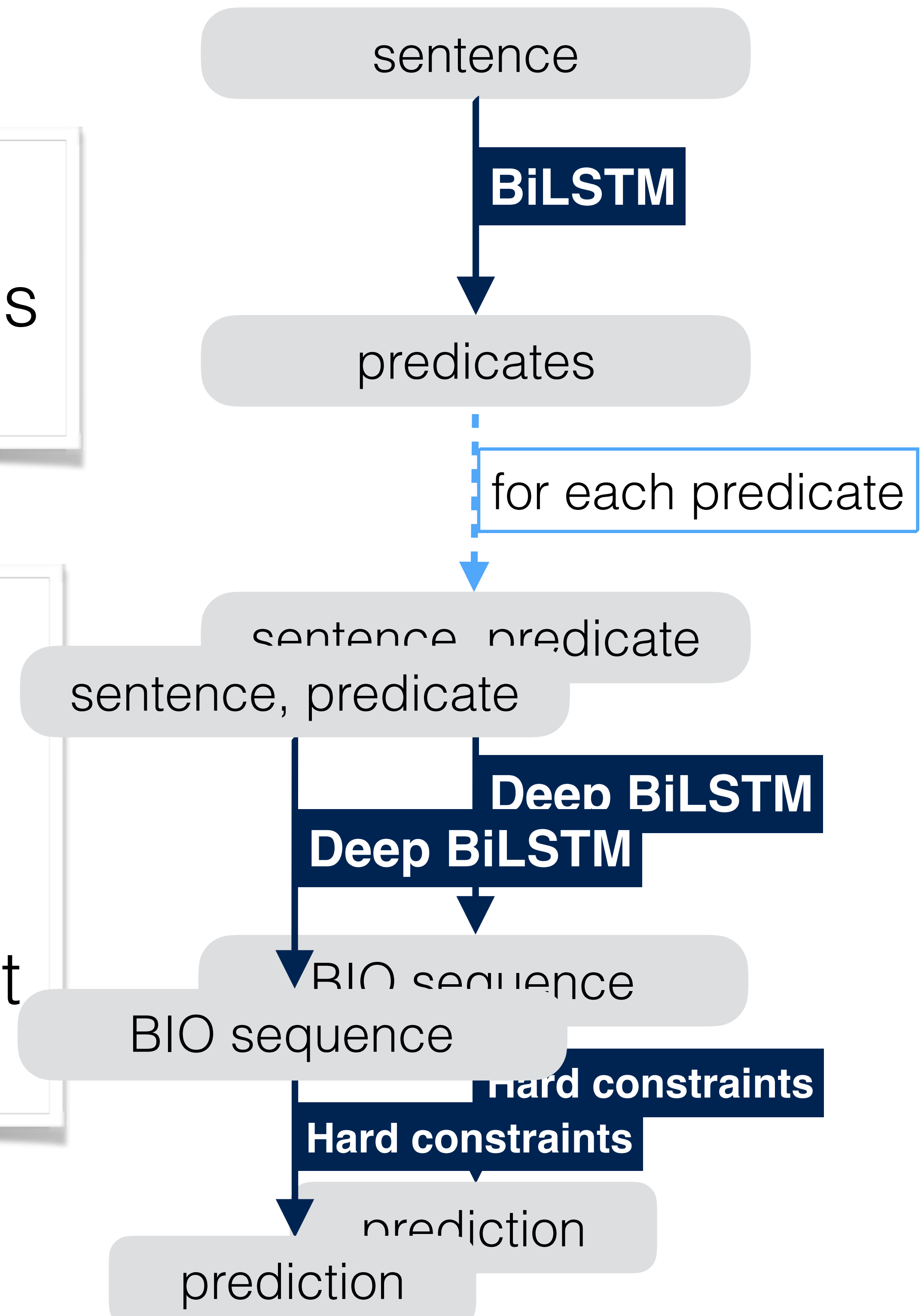


Real Scenario: No Gold Predicates!

End-to-end SRL:
Given sentence, predict all predicates as well as their arguments.

Pipelined approach:
Identify predicates first, then run the BIO tagger for each predicate.

(No way to recover from recall loss at predicate ID stage …)

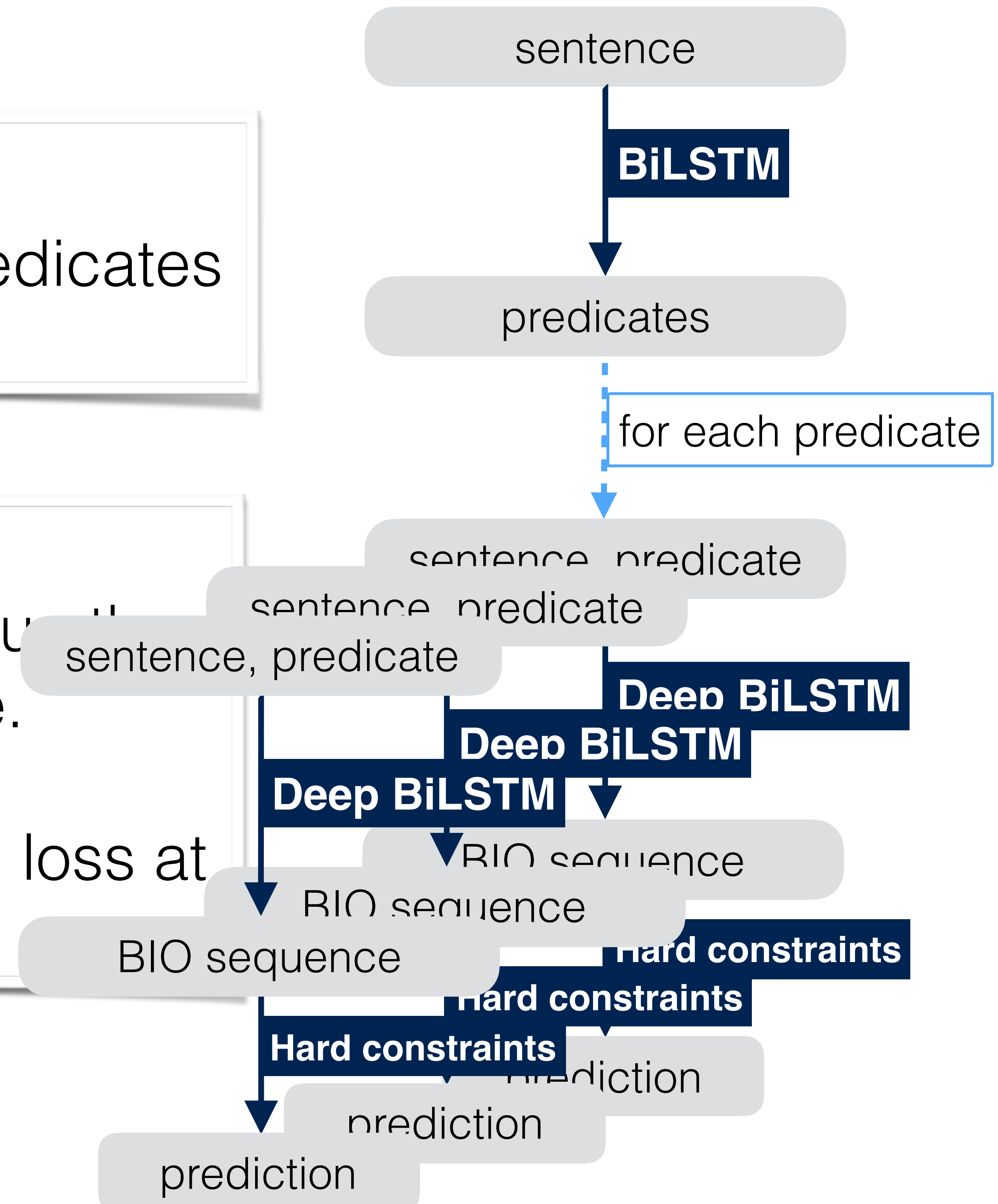


Real Scenario: No Gold Predicates!

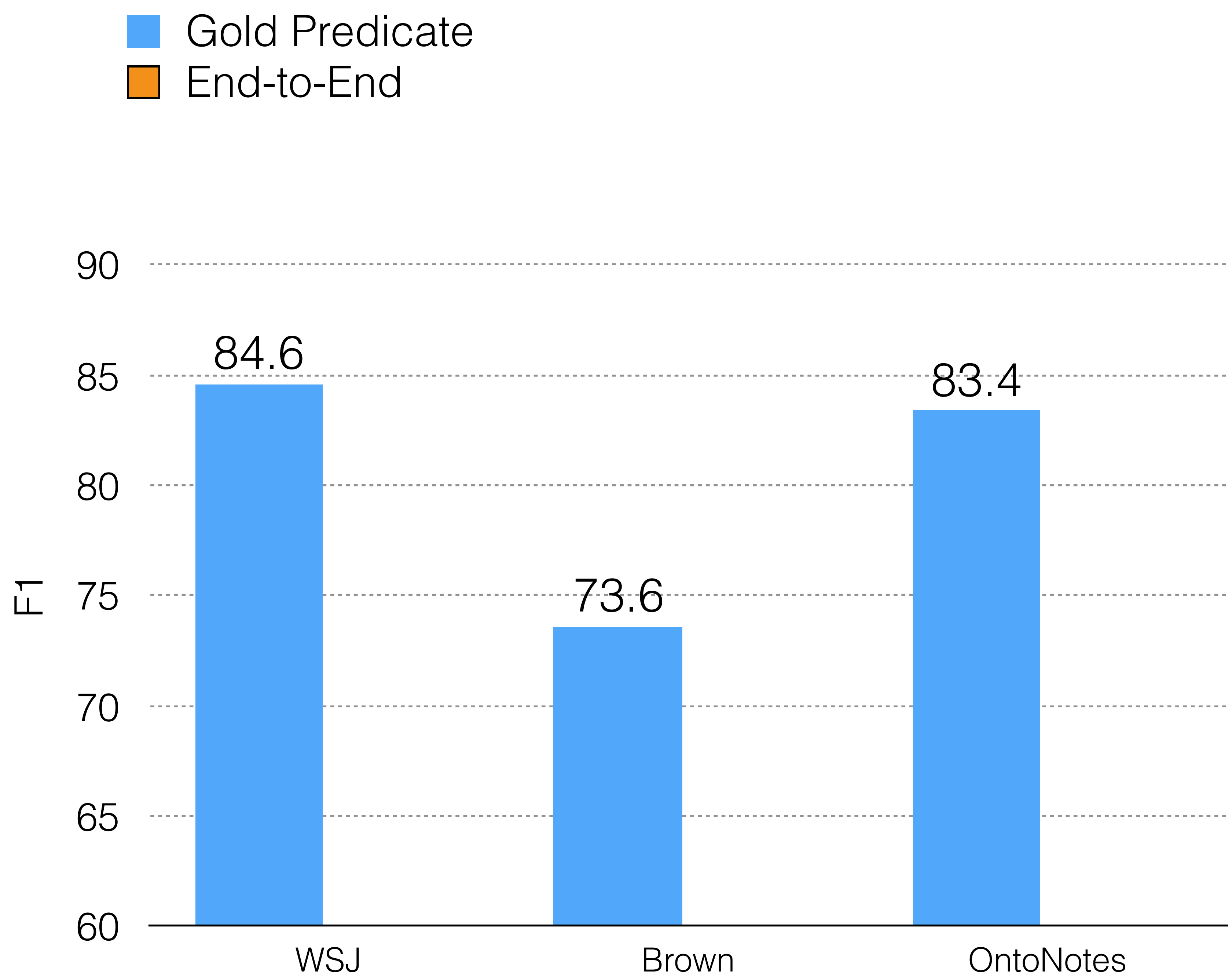
End-to-end SRL:
Given sentence, predict all predicates as well as their arguments.

Pipelined approach:
Identify predicates first, then ru
BIO tagger for each predicate.

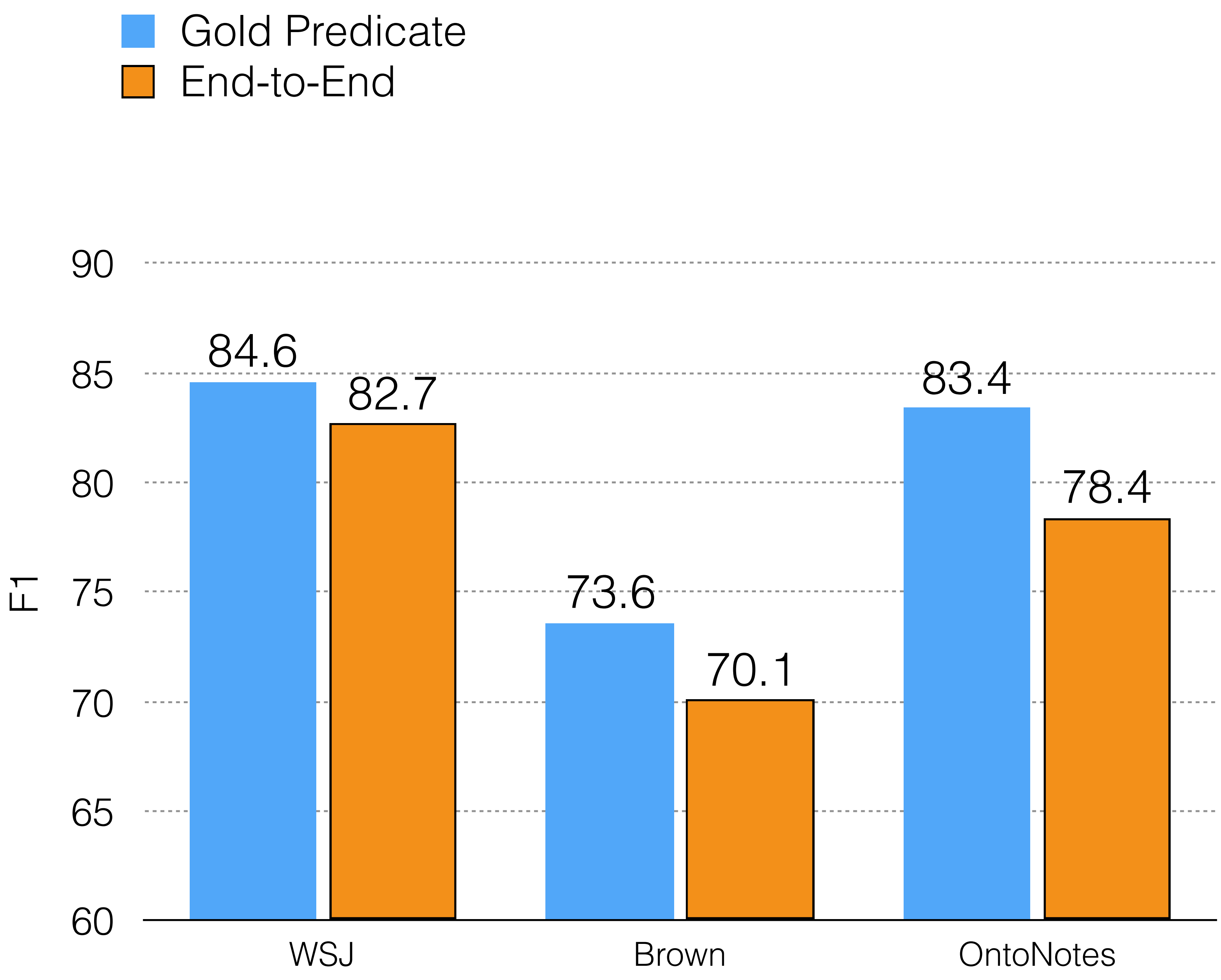
(No way to recover from recall loss at predicate ID stage …)



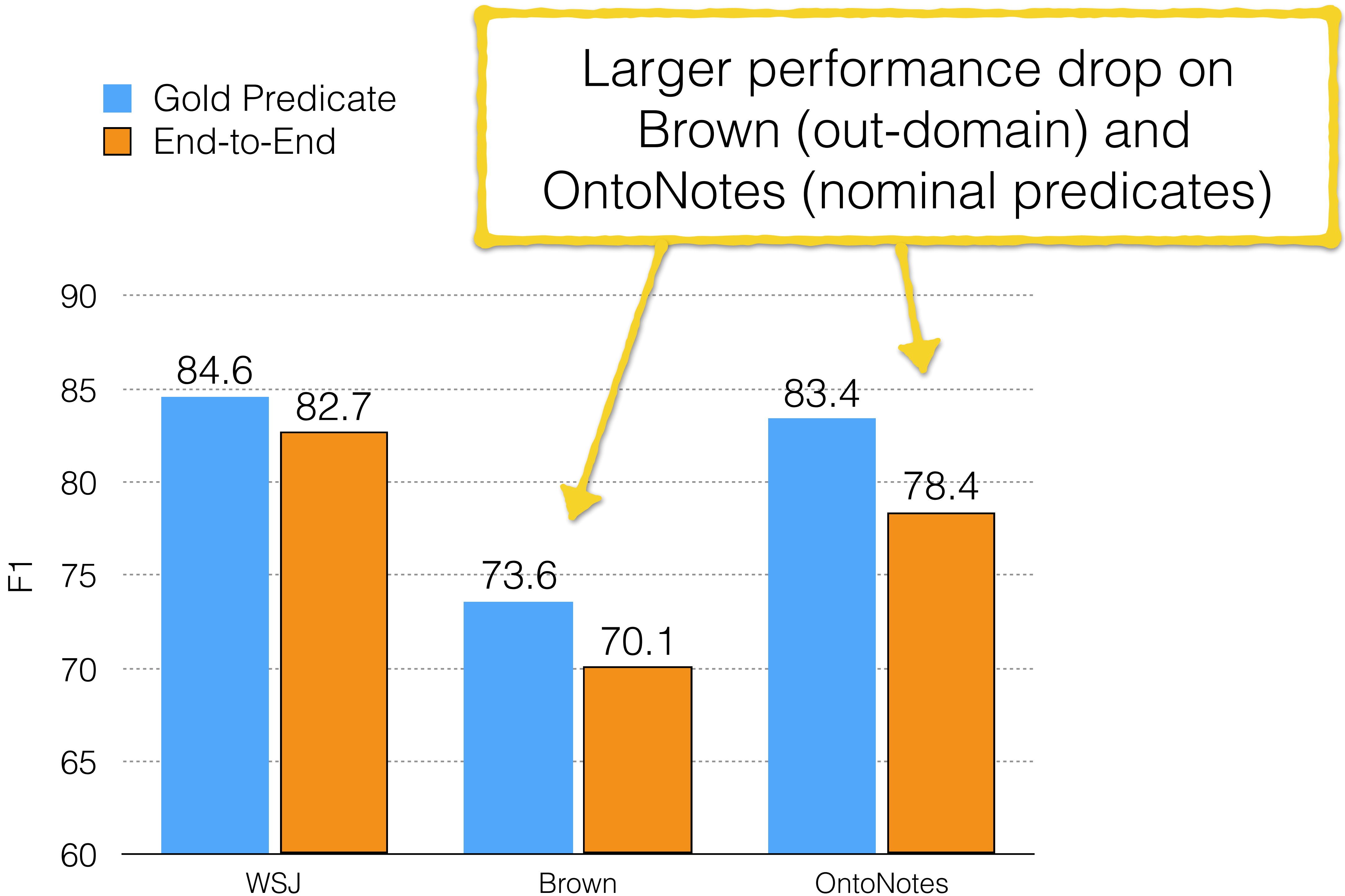
End-to-End SRL Result



End-to-End SRL Result



End-to-End SRL Result



Outline

☑ Accurate
☑ No NLP pipeline
☐ Joint predicate ID
☐ Full-text Semantics

Predicting SRL with Deep BiLSTMs

— DeepSRL

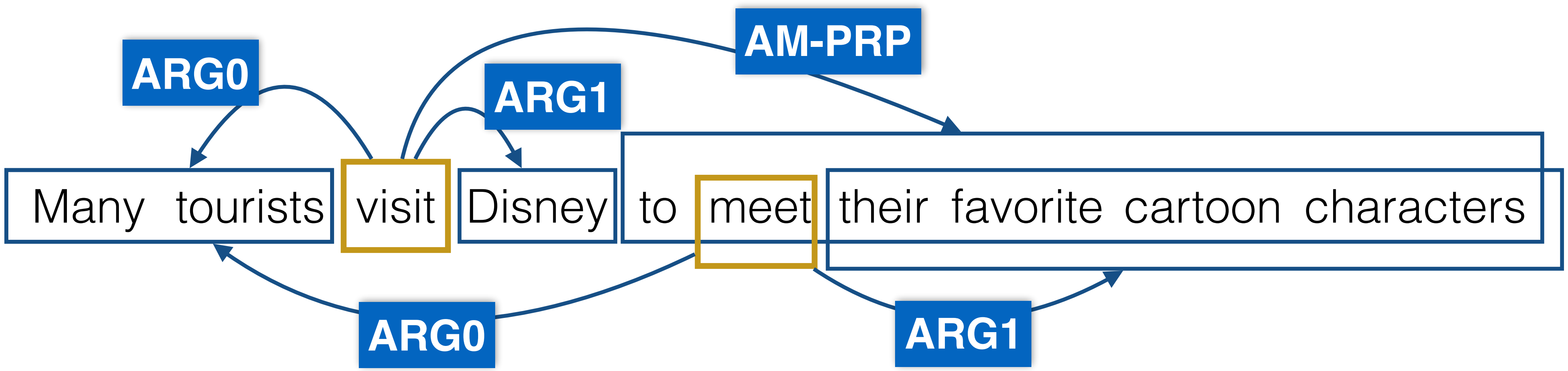
An End-to-End, Span-based SRL Model

— Labeled Span Graph Network (LSGN)

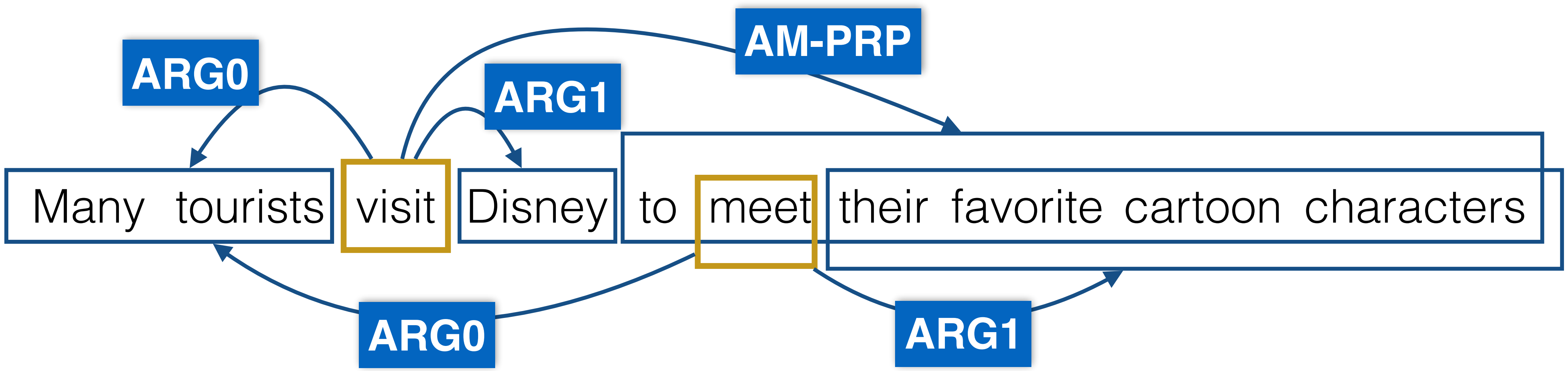
Towards Unified and Full-text Semantic Analysis

— Multi-task learning with LSGN; ScienceIE (Luan et al., 2018)

Intuition: SRL as Span-Span Relations

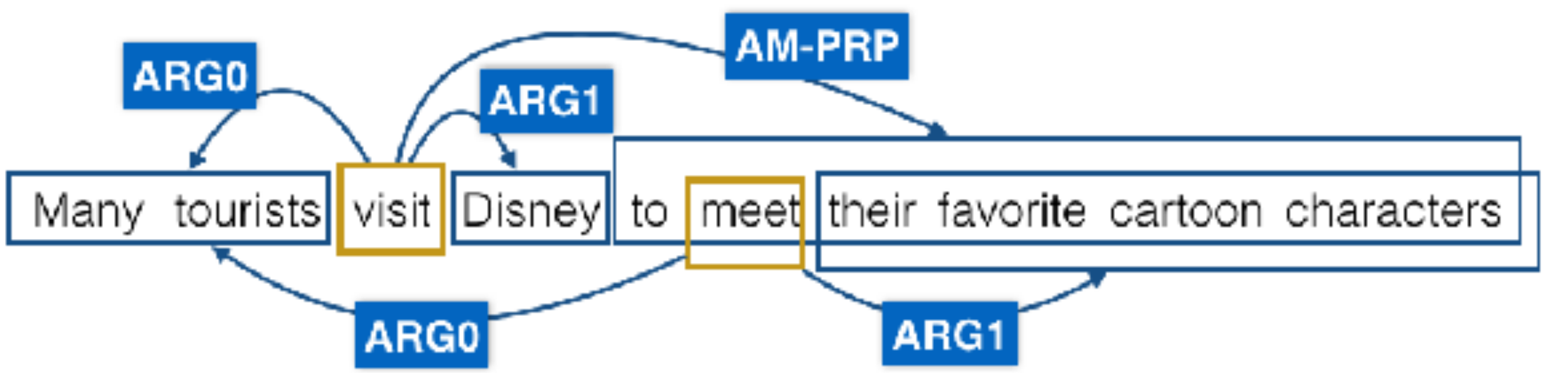


Intuition: SRL as Span-Span Relations



Challenges:
1. Span can nest within each other.
2. Too many possible edges (n^2 argument spans & n predicates).

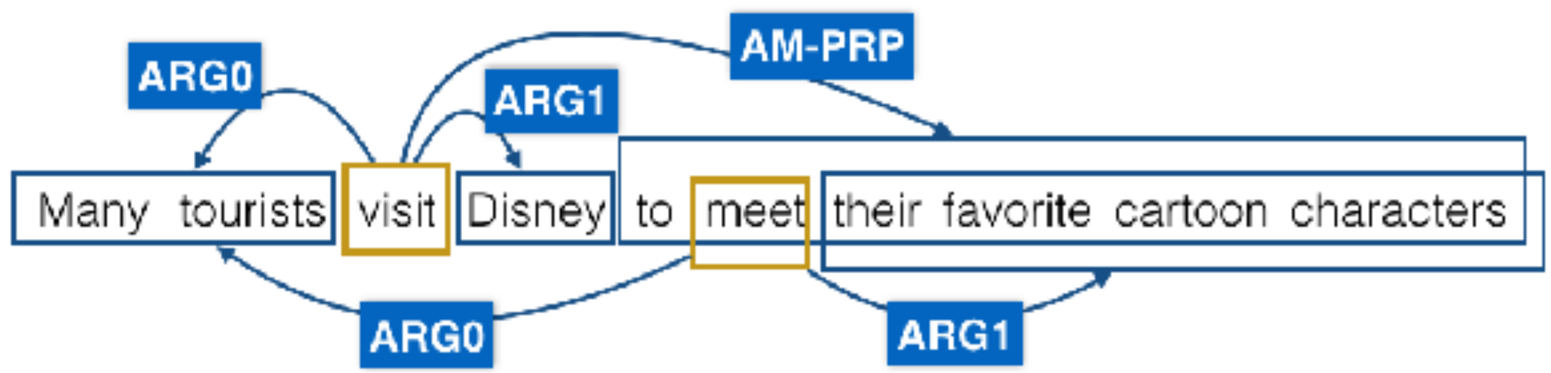
Labeled Span Graph Network (LSGN)



LSG: A graph with nodes as spans and labeled edges.

LSGN: An end-to-end network for predicting an LSG.

Labeled Span Graph Network (LSGN)



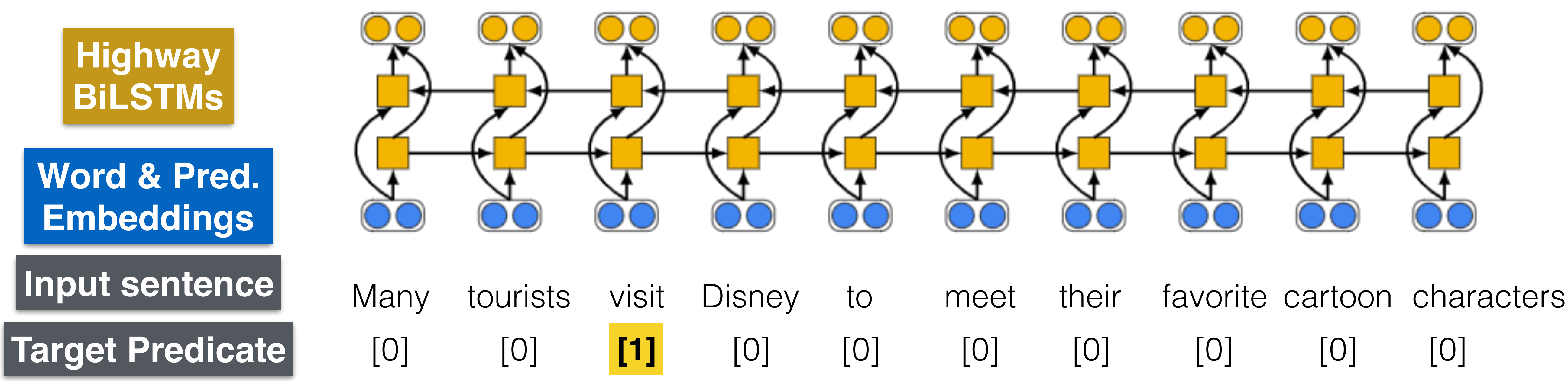
LSG: A graph with nodes as spans and labeled edges.

LSGN: An end-to-end network for predicting an LSG.

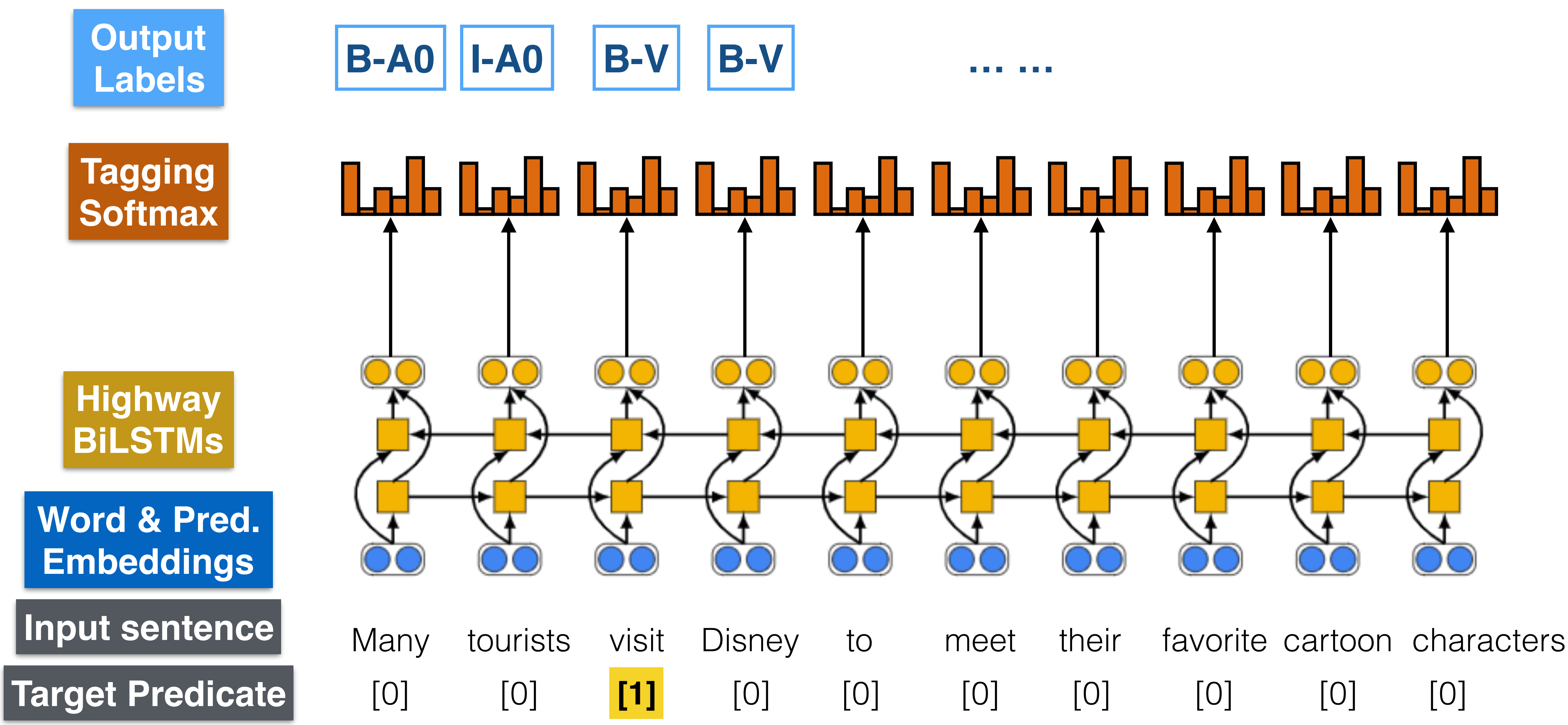
Many NLP structures can be considered as an LSG:

- **SRL (First end-to-end model!)**
- Coreference resolution (Lee et al. 2017)
- Named entity recognition and relation extraction (Luan et al., 2018)

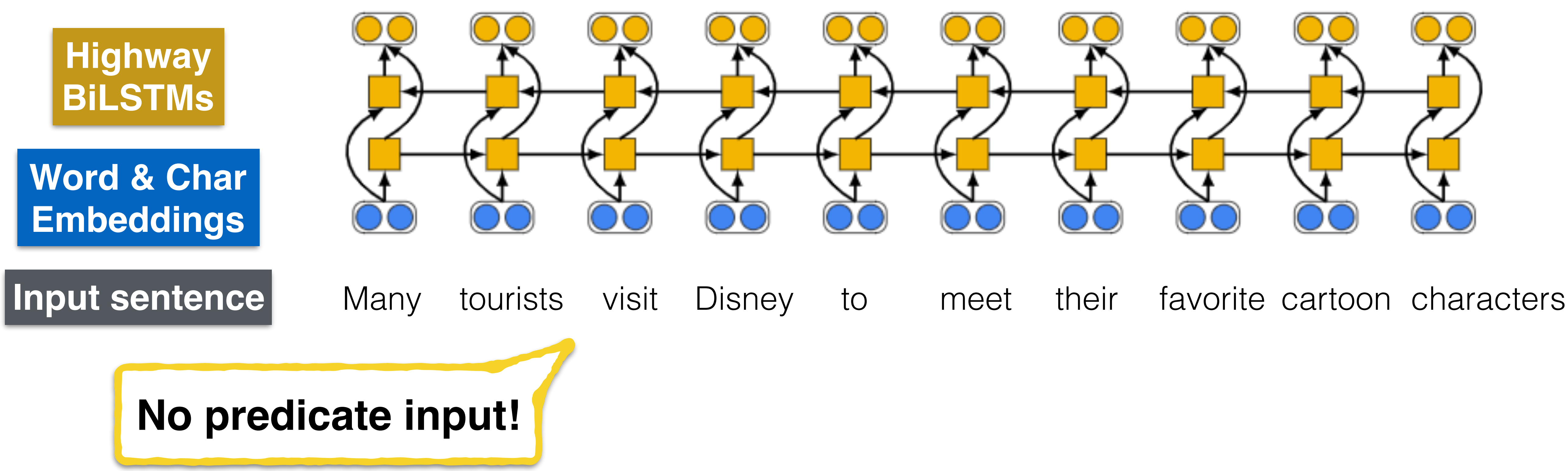
DeepSRL Architecture (Revisit)



DeepSRL Architecture (Revisit)

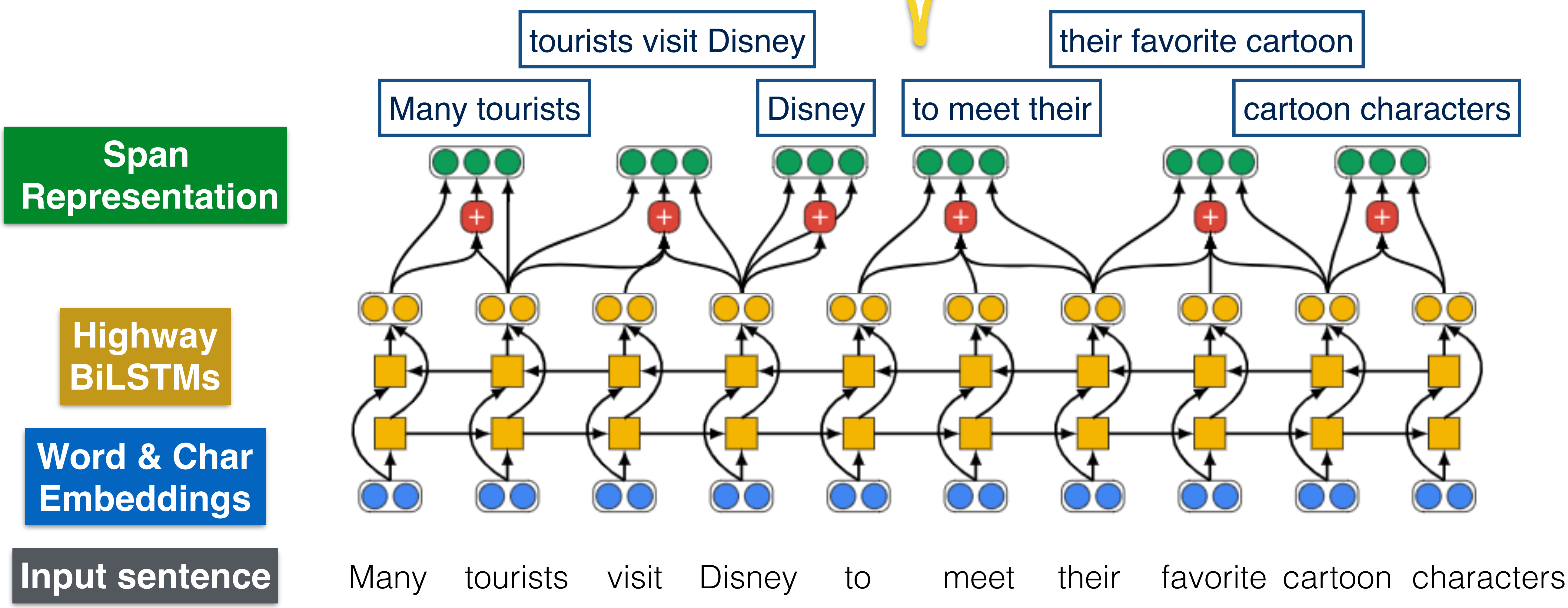


LSGN Architecture: Overview



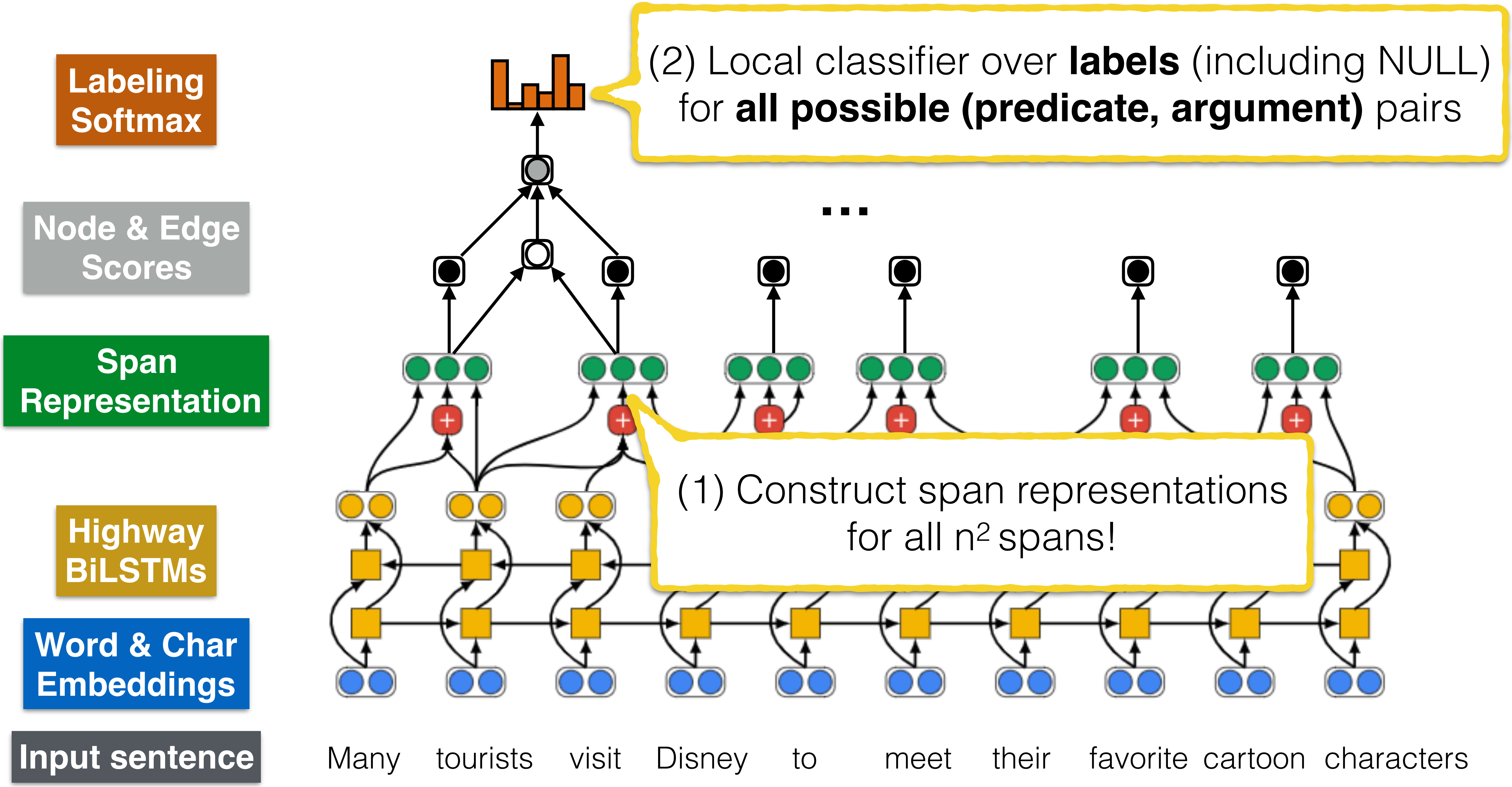
LSGN Architecture: Overview

(1) Construct span representations for all n^2 spans!

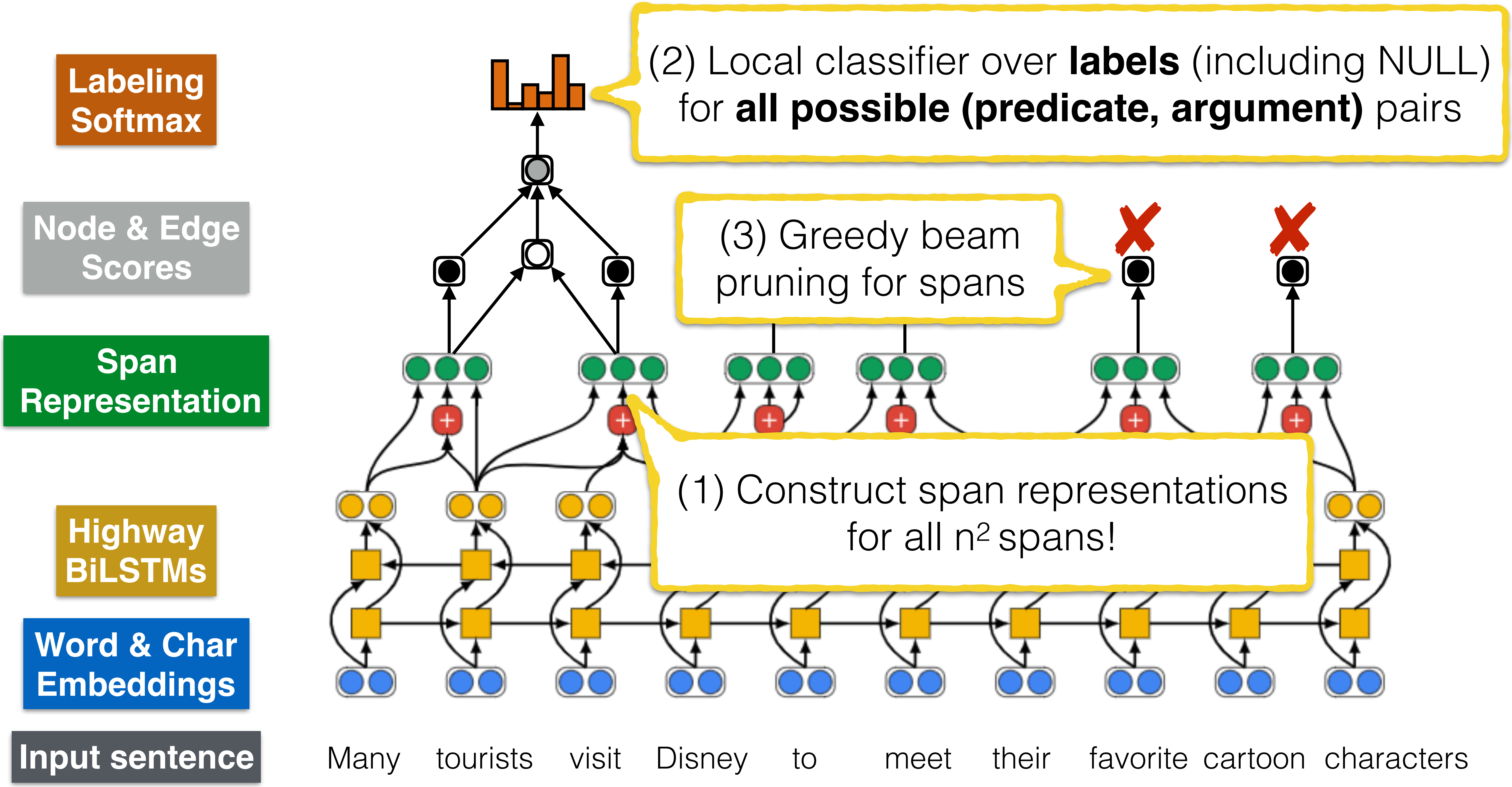


No predicate input!

LSGN Architecture: Overview



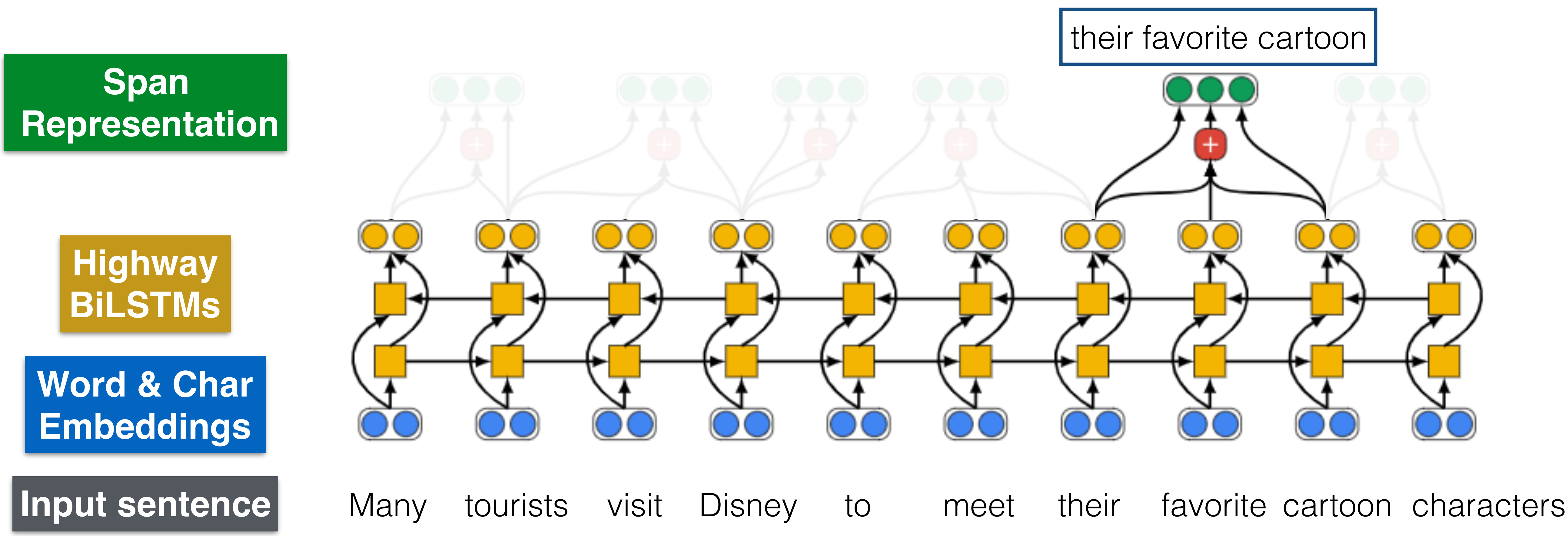
LSGN Architecture: Overview



(1) Span Representations

(2) Local Label Classifiers

(3) Span Pruning



(Same as Lee et al., 2017)

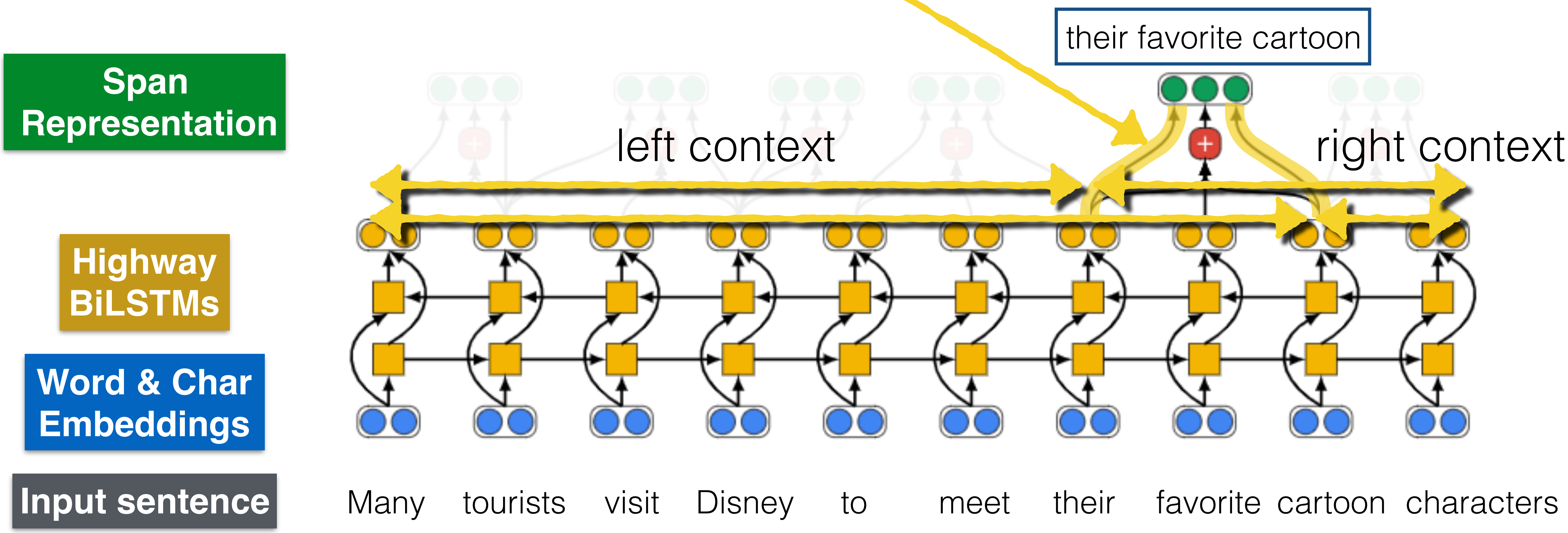
(1) Span Representations

(2) Local Label Classifiers

(3) Span Pruning

LSTM boundary points

$$[\text{BiLSTM}(w_1 : w_n)_{\text{START}}, \text{BiLSTM}(w_1 : w_n)_{\text{END}}]$$



(Same as Lee et al., 2017)

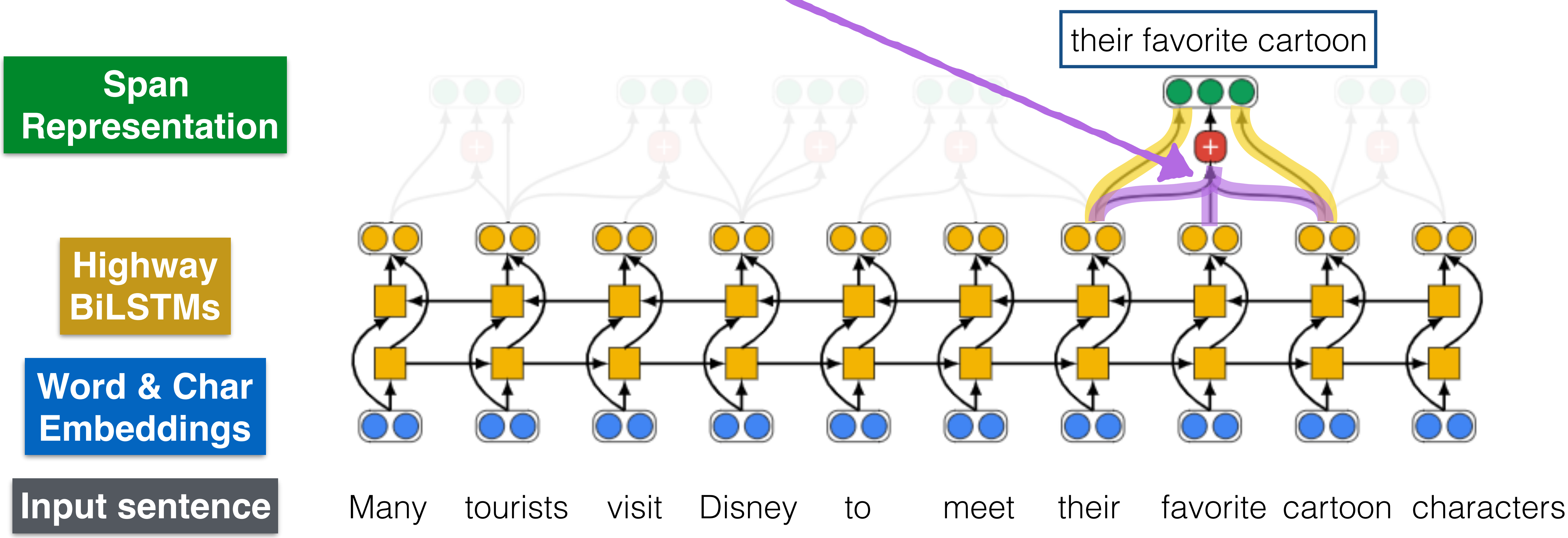
(1) Span Representations

(2) Local Label Classifiers

(3) Span Pruning

LSTM boundary points $[\textsc{BiLstm}(w_1 : w_n)_{\textsc{Start}}, \textsc{BiLstm}(w_1 : w_n)_{\textsc{End}}]$

Attention over words $\sum_{i=\textsc{Start}}^{\textsc{End}} \textsc{SoftMax}(a_{\textsc{Start}} : a_{\textsc{End}})_i \; w_i$

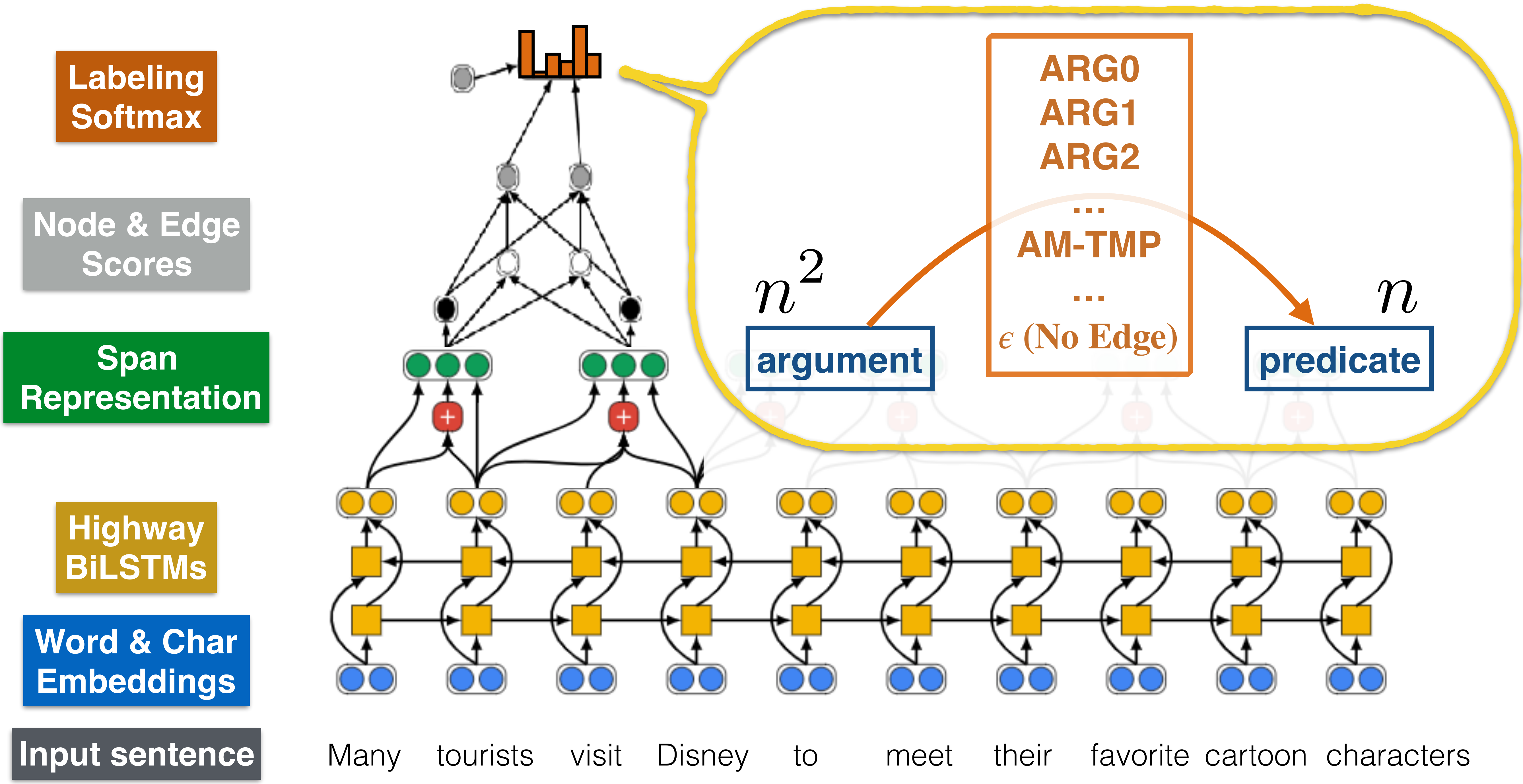


(Same as Lee et al., 2017)

(1) Span Representations

(2) Local Label Classifiers

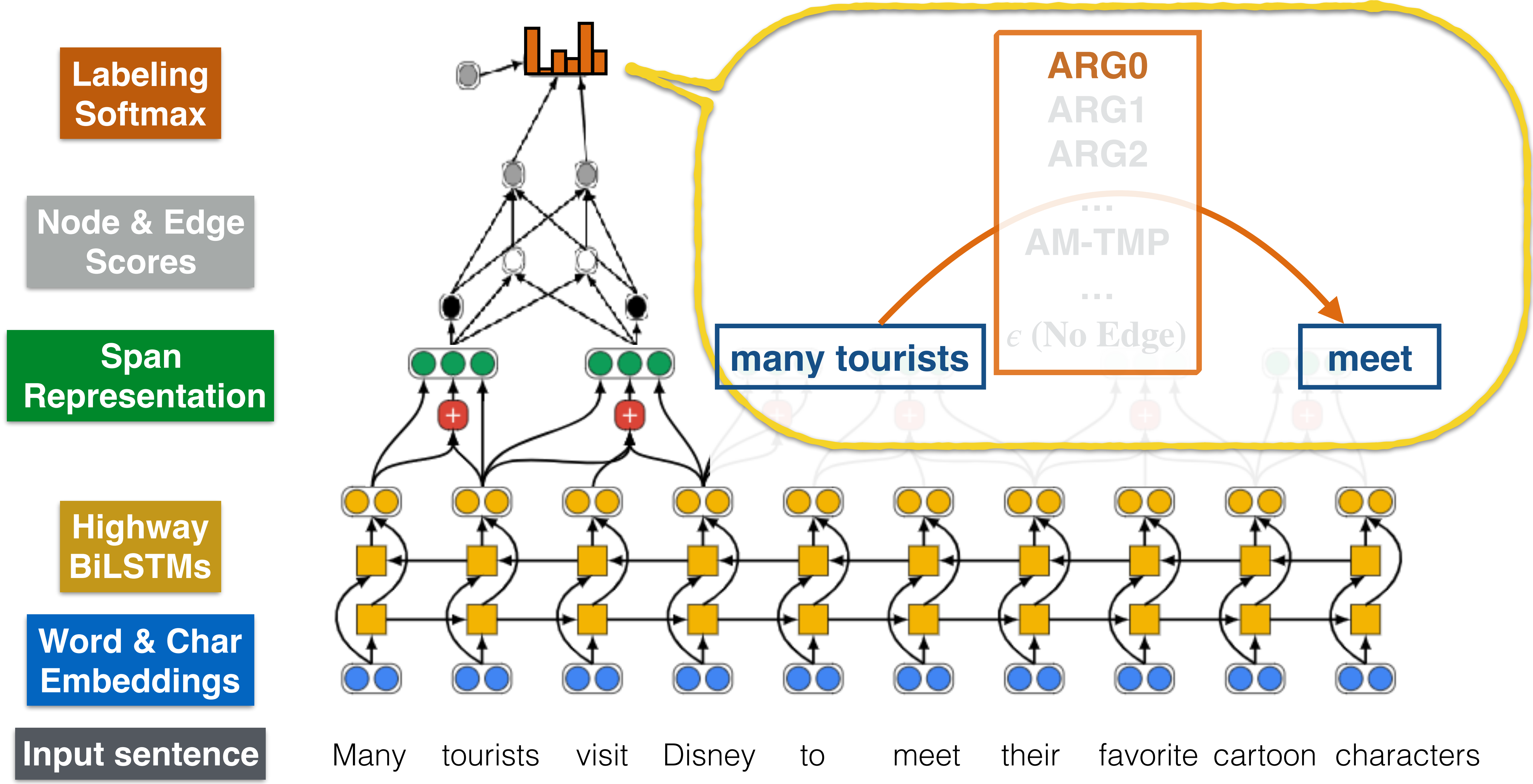
(3) Span Pruning



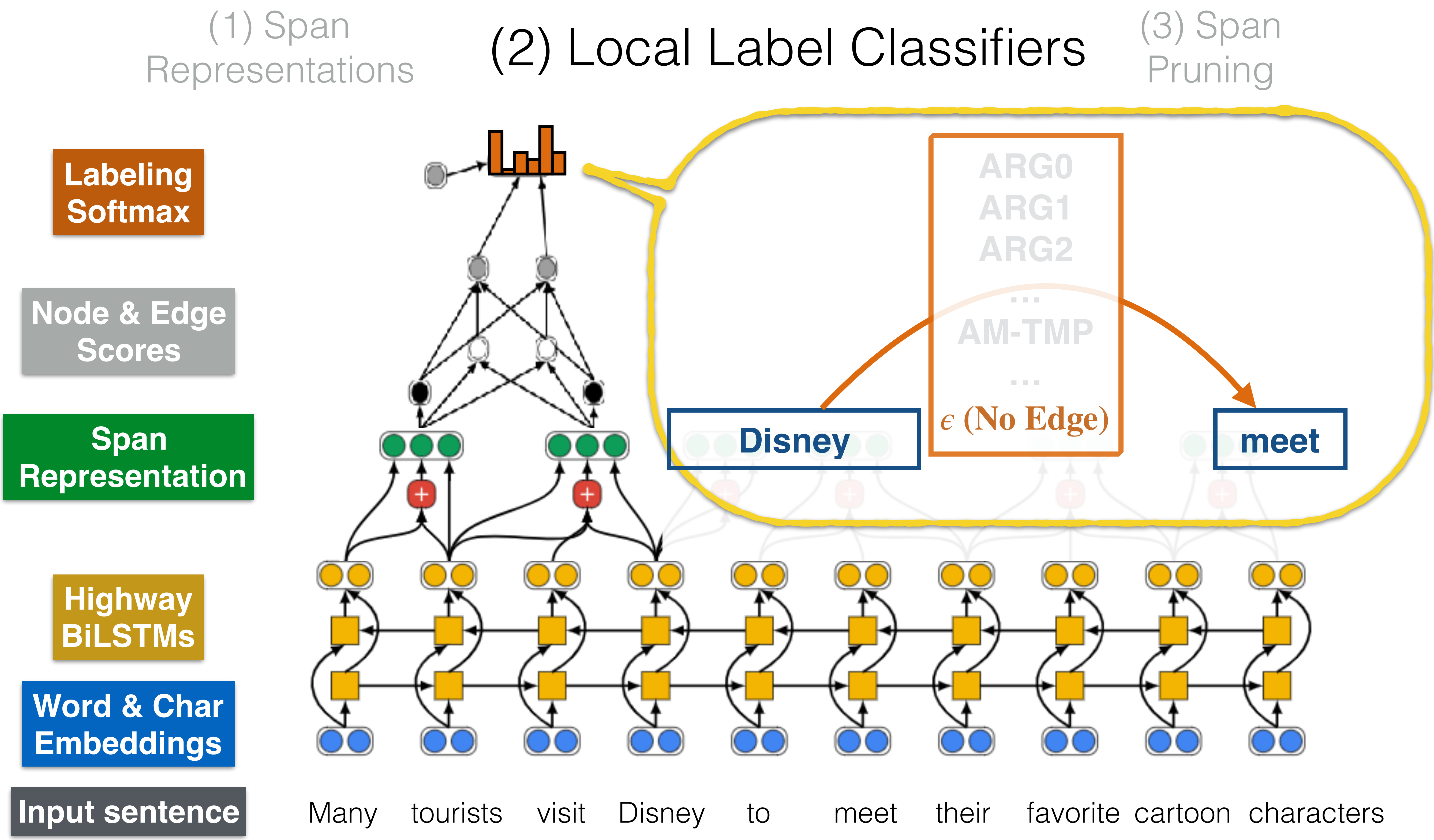
(1) Span Representations

(2) Local Label Classifiers

(3) Span Pruning



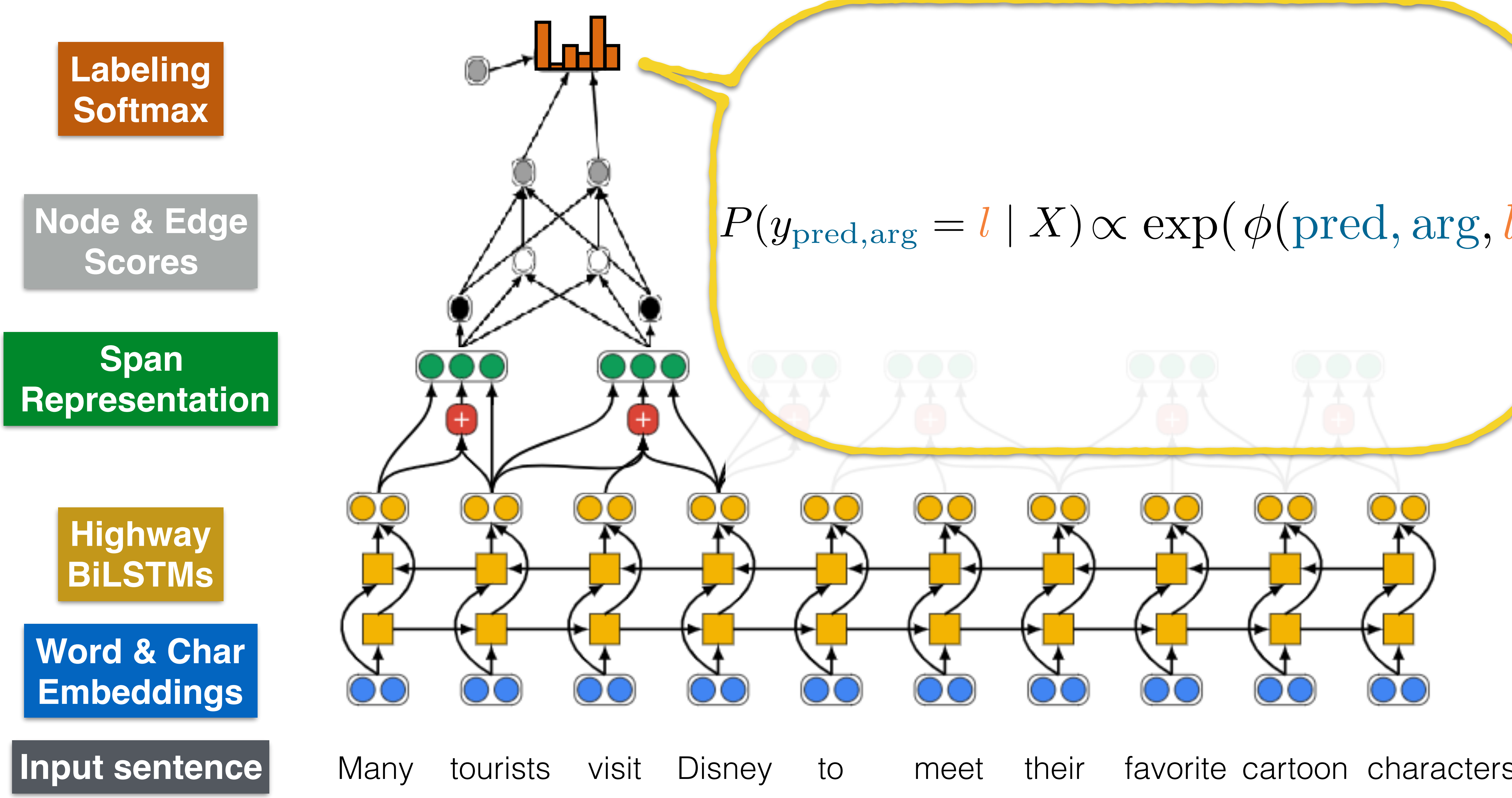
(2) Local Label Classifiers



(1) Span Representations

(2) Local Label Classifiers

(3) Span Pruning



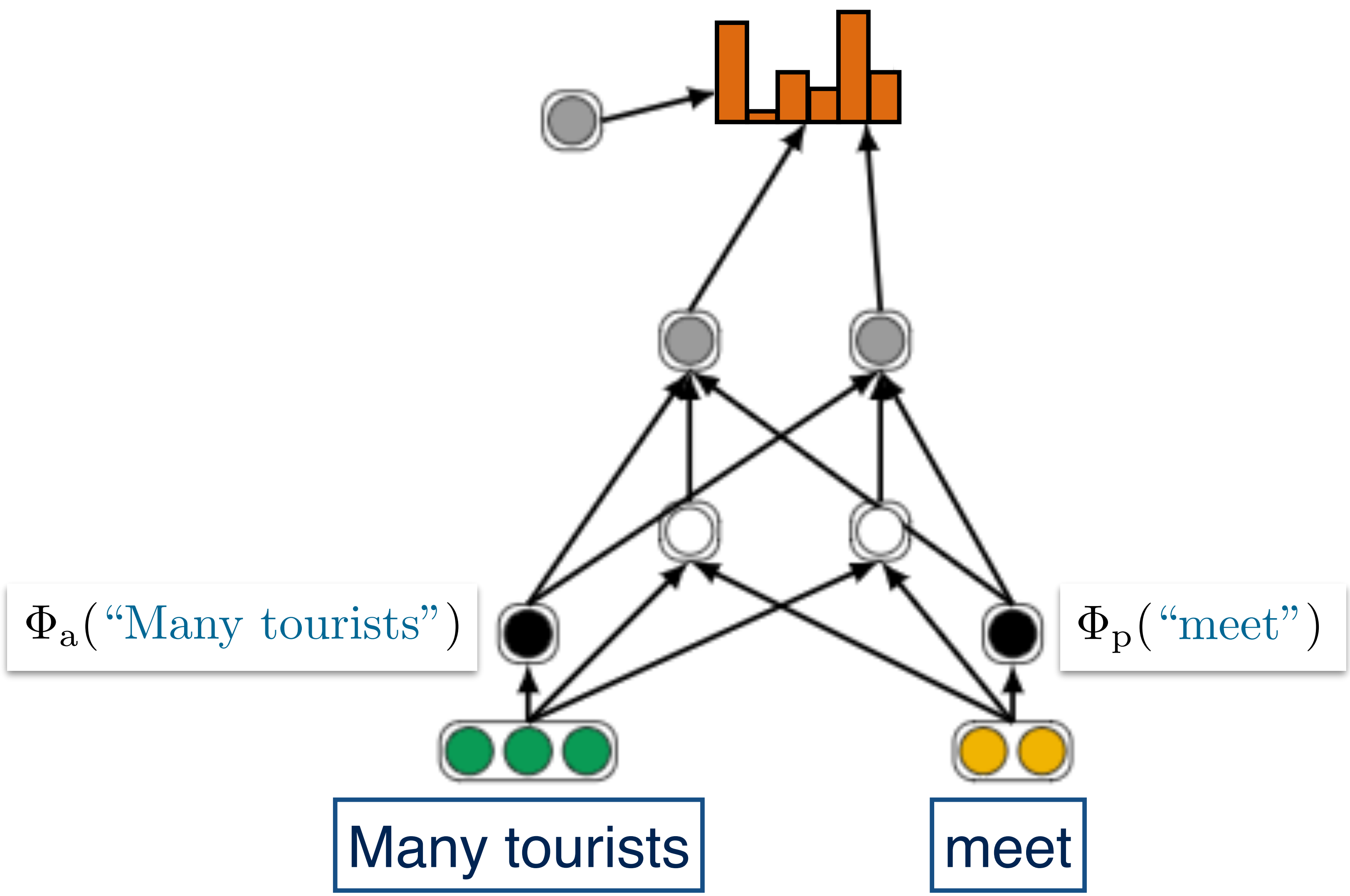
$$P(y_{\text{pred},\text{arg}} = l \mid X) \propto \exp(\phi(\text{pred}, \text{arg}, l))$$

(1) Span Representations

(2) Local Label Classifiers

(3) Span Pruning

$$\phi(\mathrm{pred}, \mathrm{arg}, l) = \boxed{\Phi_{\mathrm{a}}(\mathrm{arg})} + \boxed{\Phi_{\mathrm{p}}(\mathrm{pred})} + \Phi_{\mathrm{rel}}^{(l)}(\mathrm{arg}, \mathrm{pred})$$



Pred./Arg. score

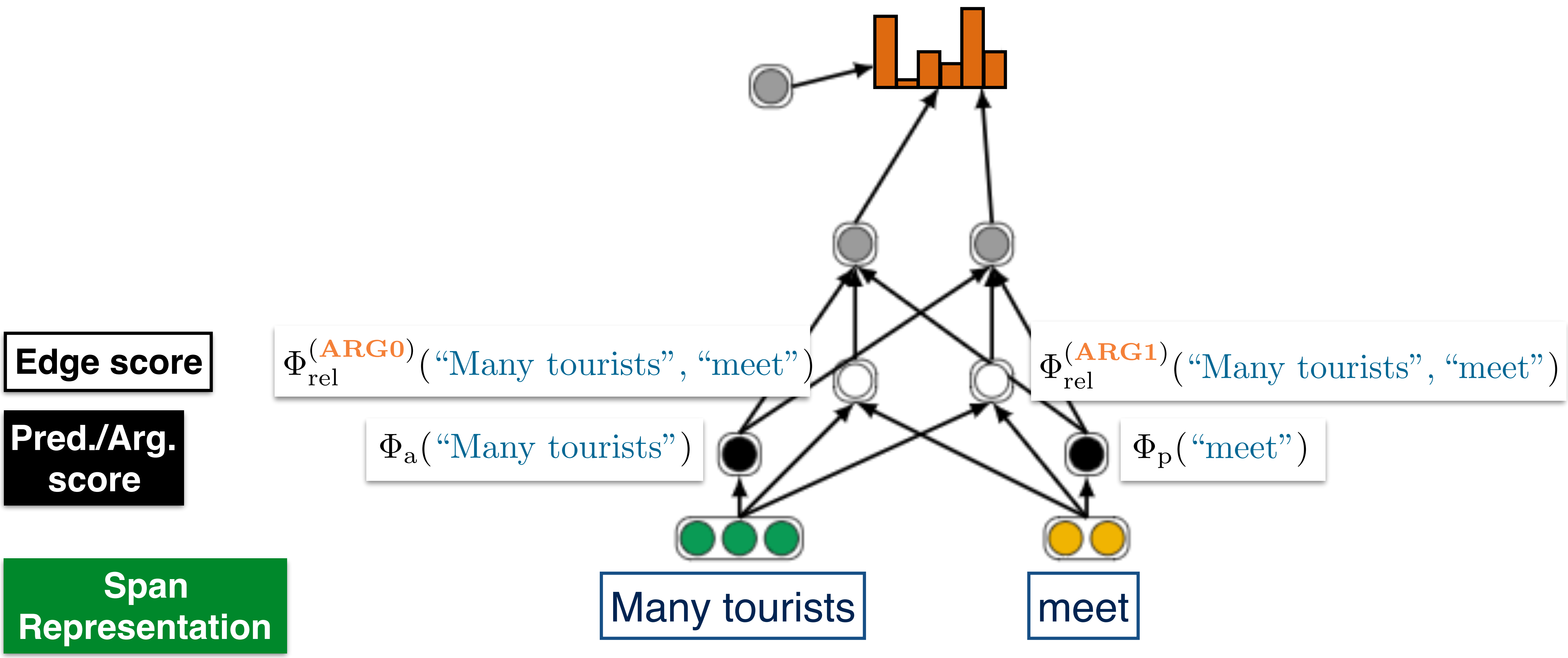
Span Representation

(1) Span Representations

(2) Local Label Classifiers

(3) Span Pruning

$$\phi(\text{pred}, \text{arg}, l) = \Phi_{\text{a}}(\text{arg}) + \Phi_{\text{p}}(\text{pred}) + \boxed{\Phi_{\text{rel}}^{(l)}(\text{arg}, \text{pred})}$$

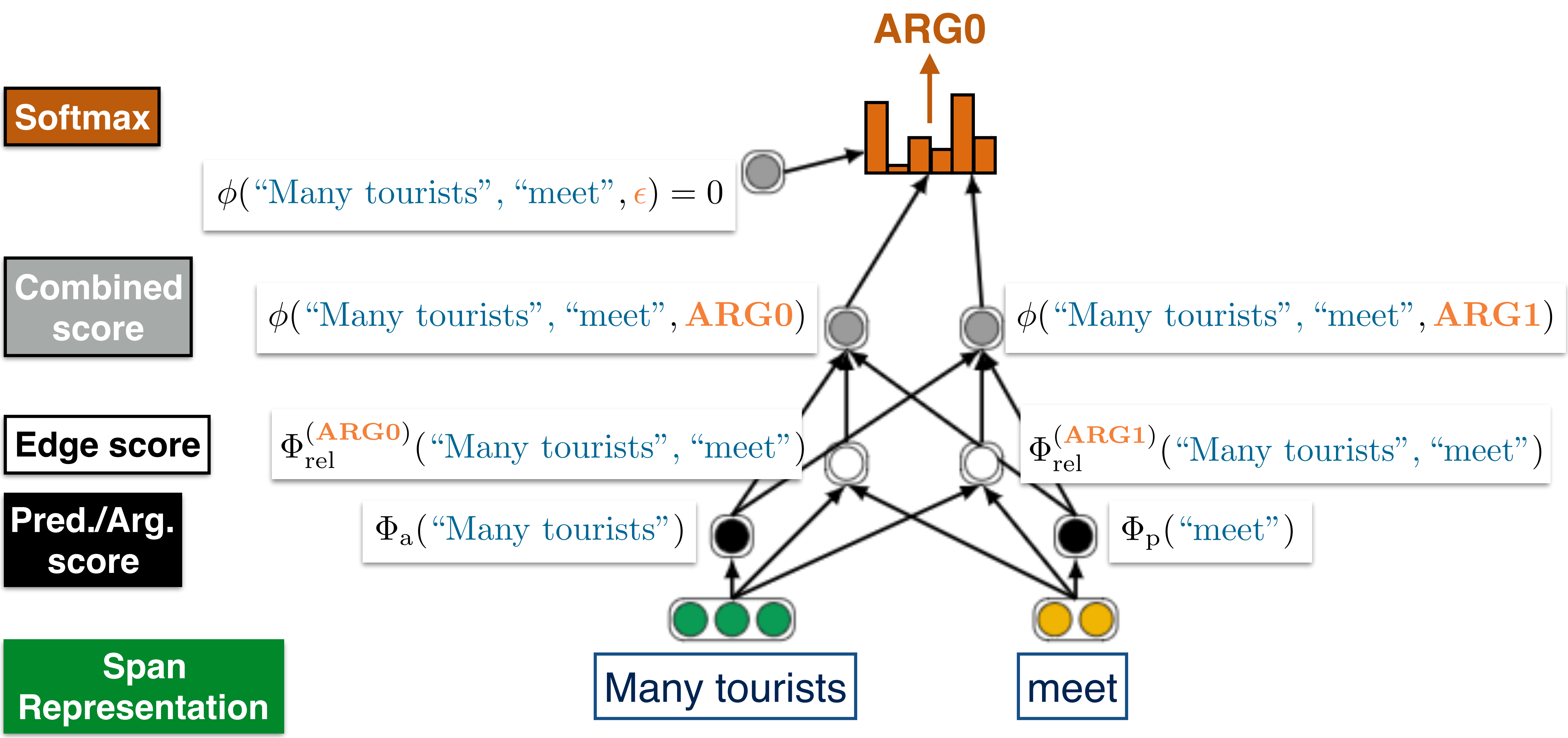


(1) Span Representations

(2) Local Label Classifiers

(3) Span Pruning

$$\boxed{\phi(\text{pred}, \text{arg}, l)} = \Phi_\text{a}(\text{arg}) + \Phi_\text{p}(\text{pred}) + \Phi_\text{rel}^{(l)}(\text{arg}, \text{pred})$$

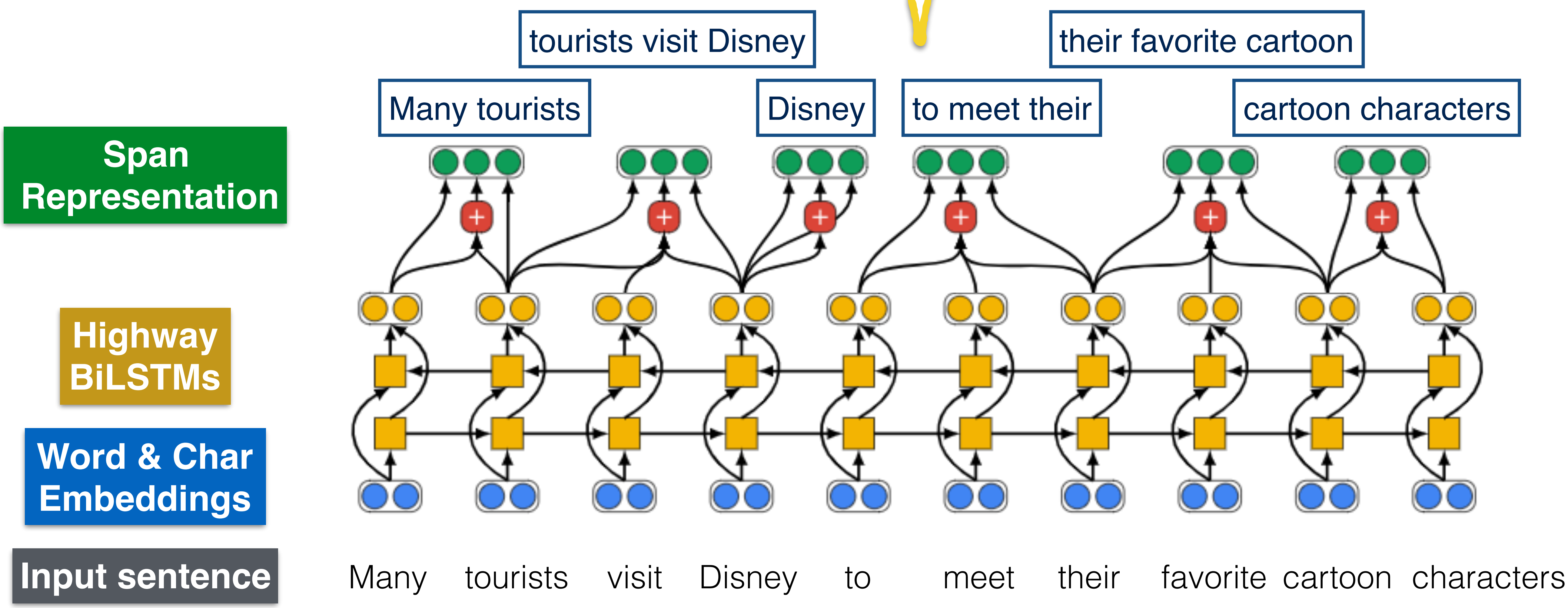


(1) Span Representations

(2) Local Label Classifiers

(3) Span Pruning

O(n^2) arguments, O(n) predicates,
—> O(n^3) edges!



(1) Span Representations

(2) Local Label Classifiers

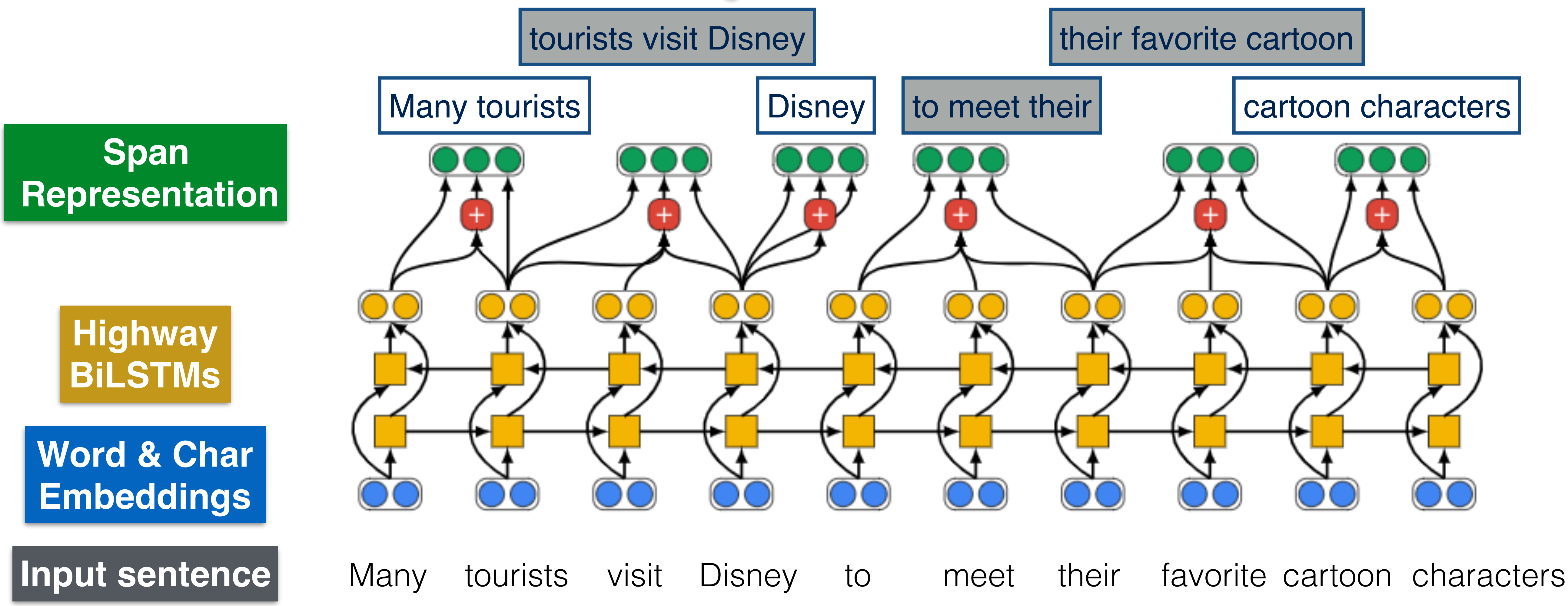
(3) Span Pruning

Only keep top **O(n)** spans using their unary scores

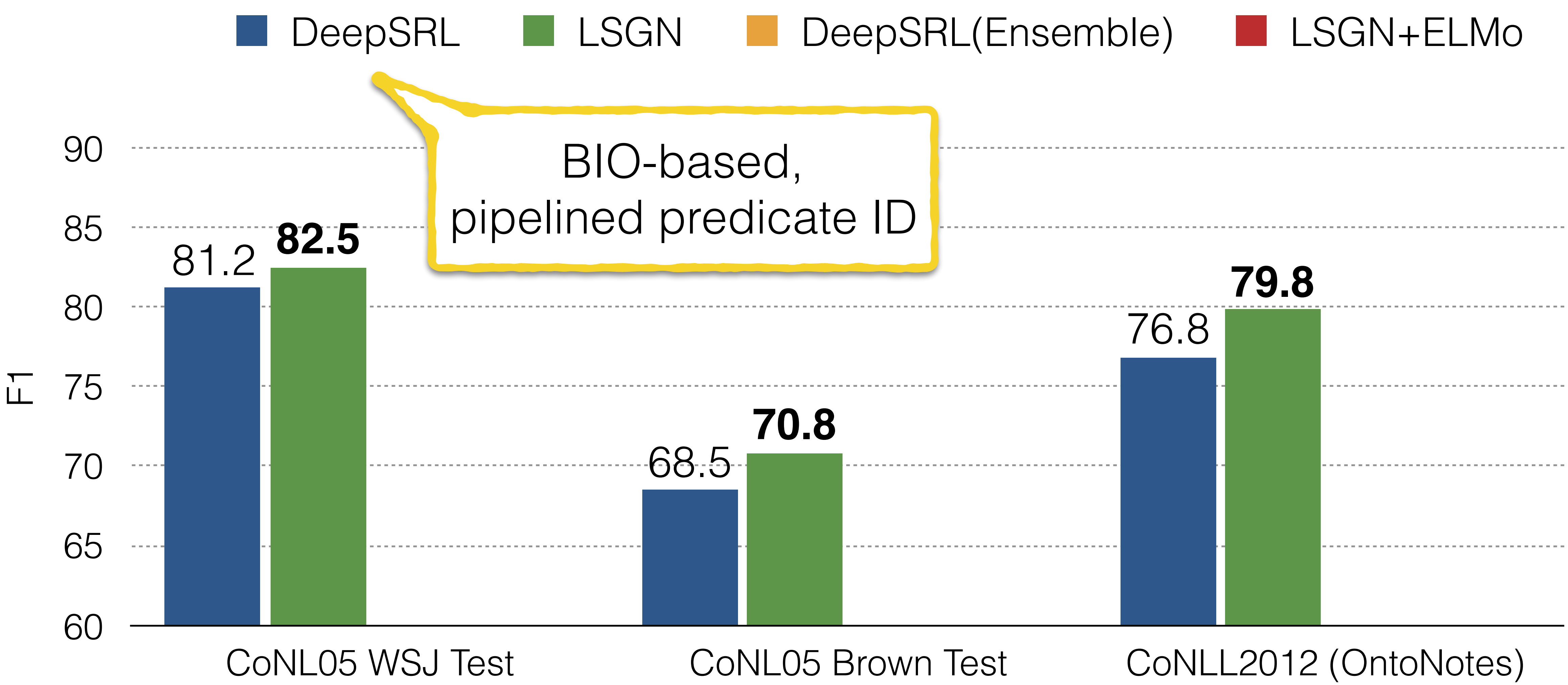
$$\Phi_a(\text{“many tourists”}) = 2.5$$

$$\Phi_a(\text{“tourists visit Disney”}) = -0.8$$

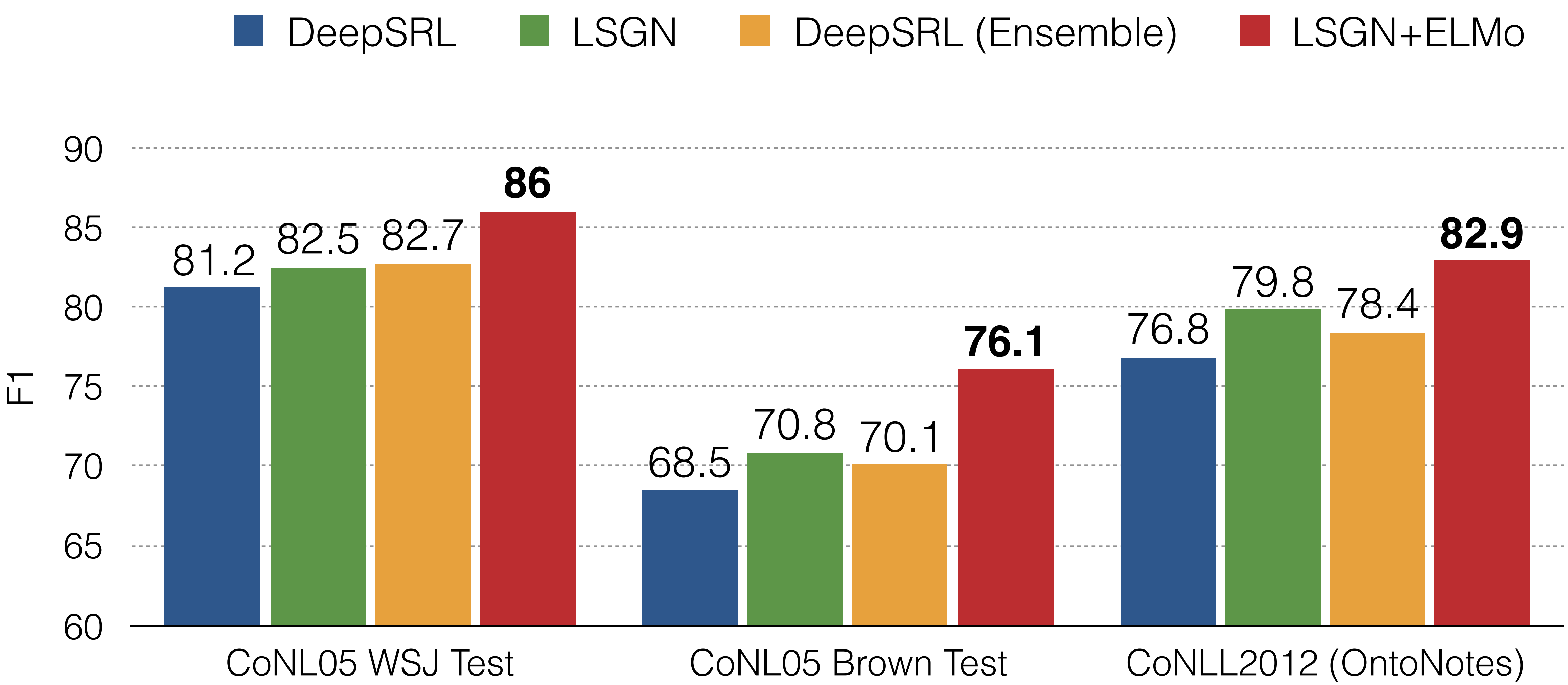
$$\ldots$$



End-to-End SRL Results



End-to-End SRL Results

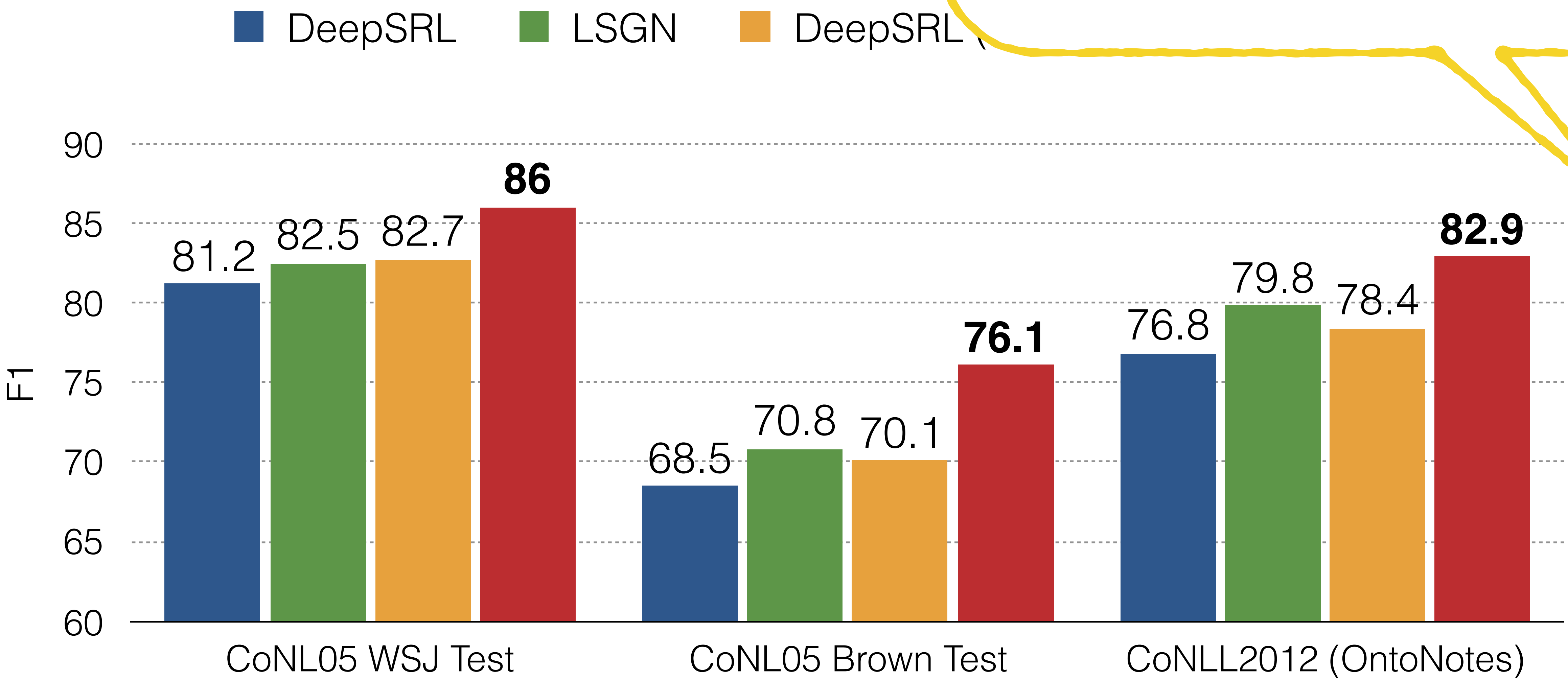


With **ELMo**, over 3 points improvement over SotA ensemble!

*ELMo: Deep Contextualized Word Representations, Peters et al., 2018

End-to-End SRL

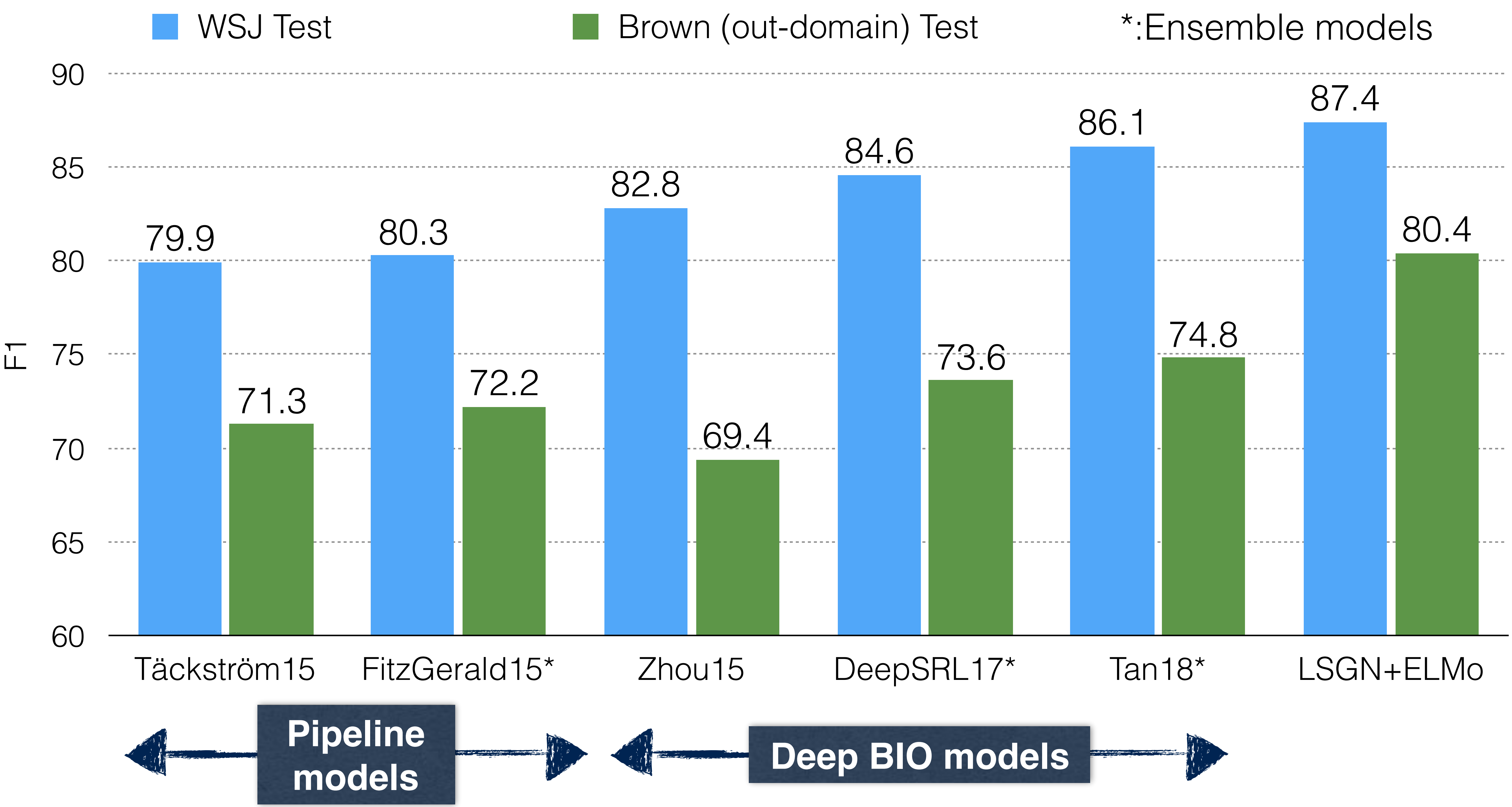
- New SotA (Strubell et al., 2018) with syntax-informed transformer model.



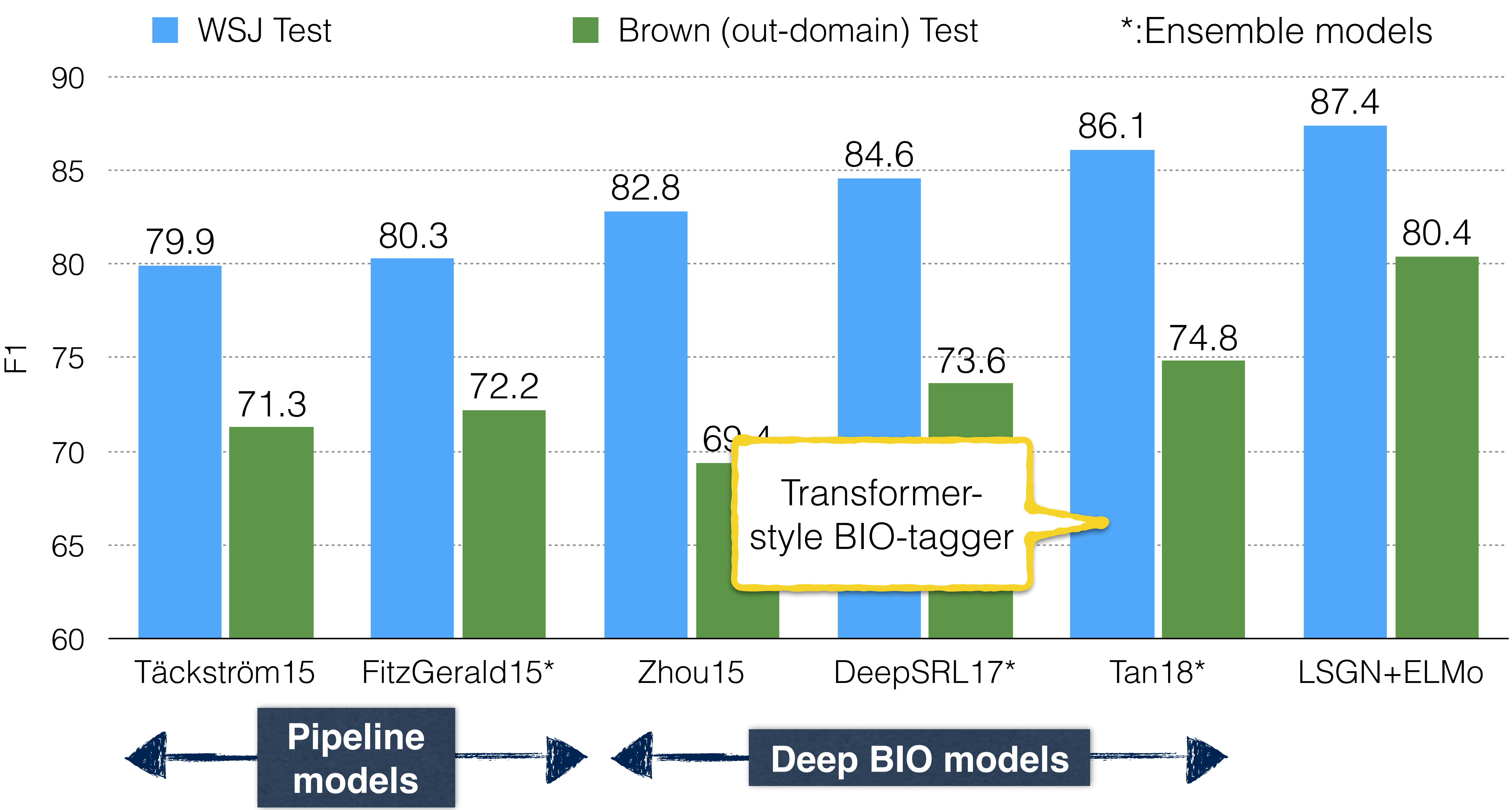
With **ELMo**, over 3 points improvement over SotA ensemble!

*ELMo: Deep Contextualized Word Representations, Peters et al., 2018

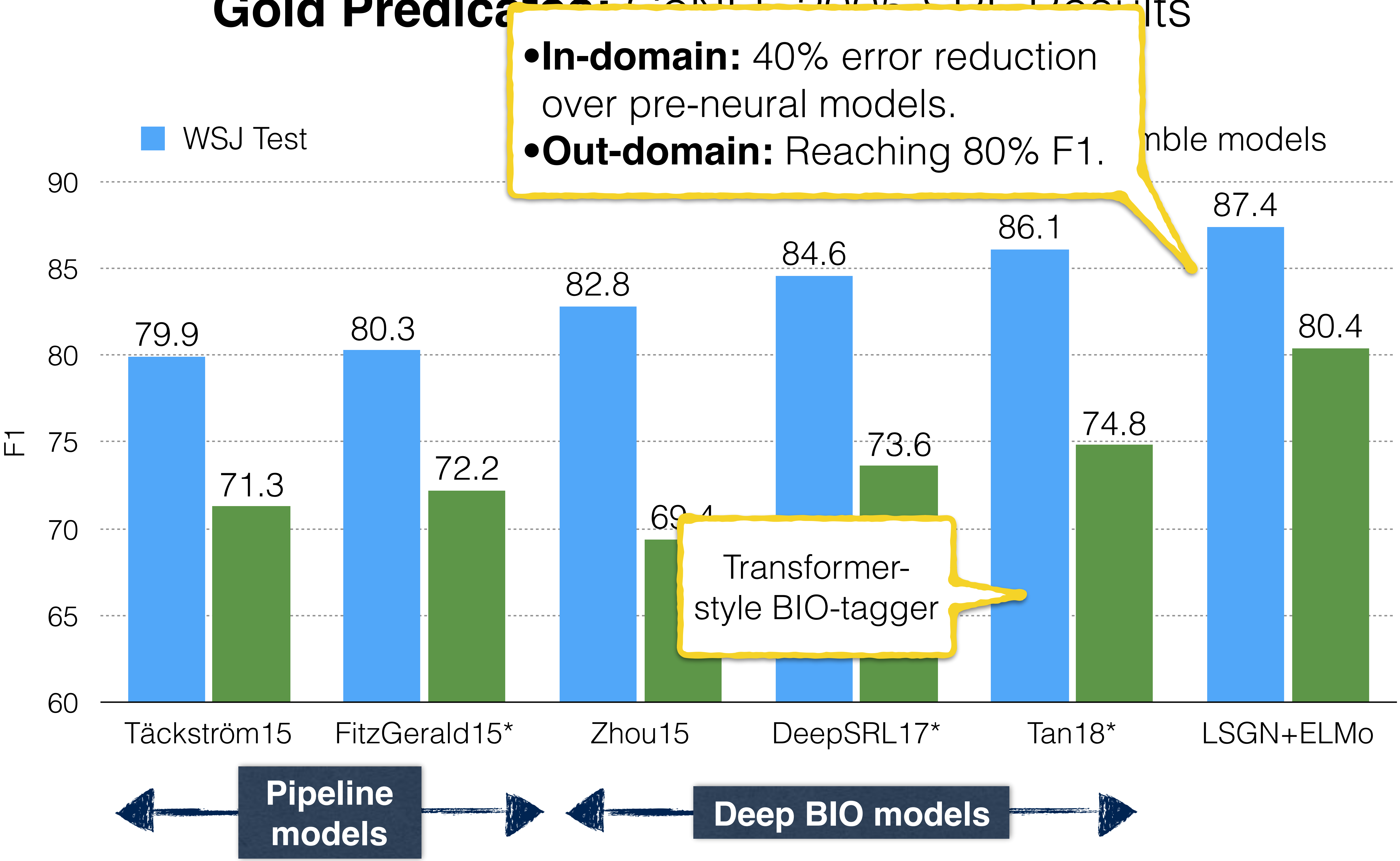
Gold Predicates: CoNLL 2005 SRL Results



Gold Predicates: CoNLL 2005 SRL Results

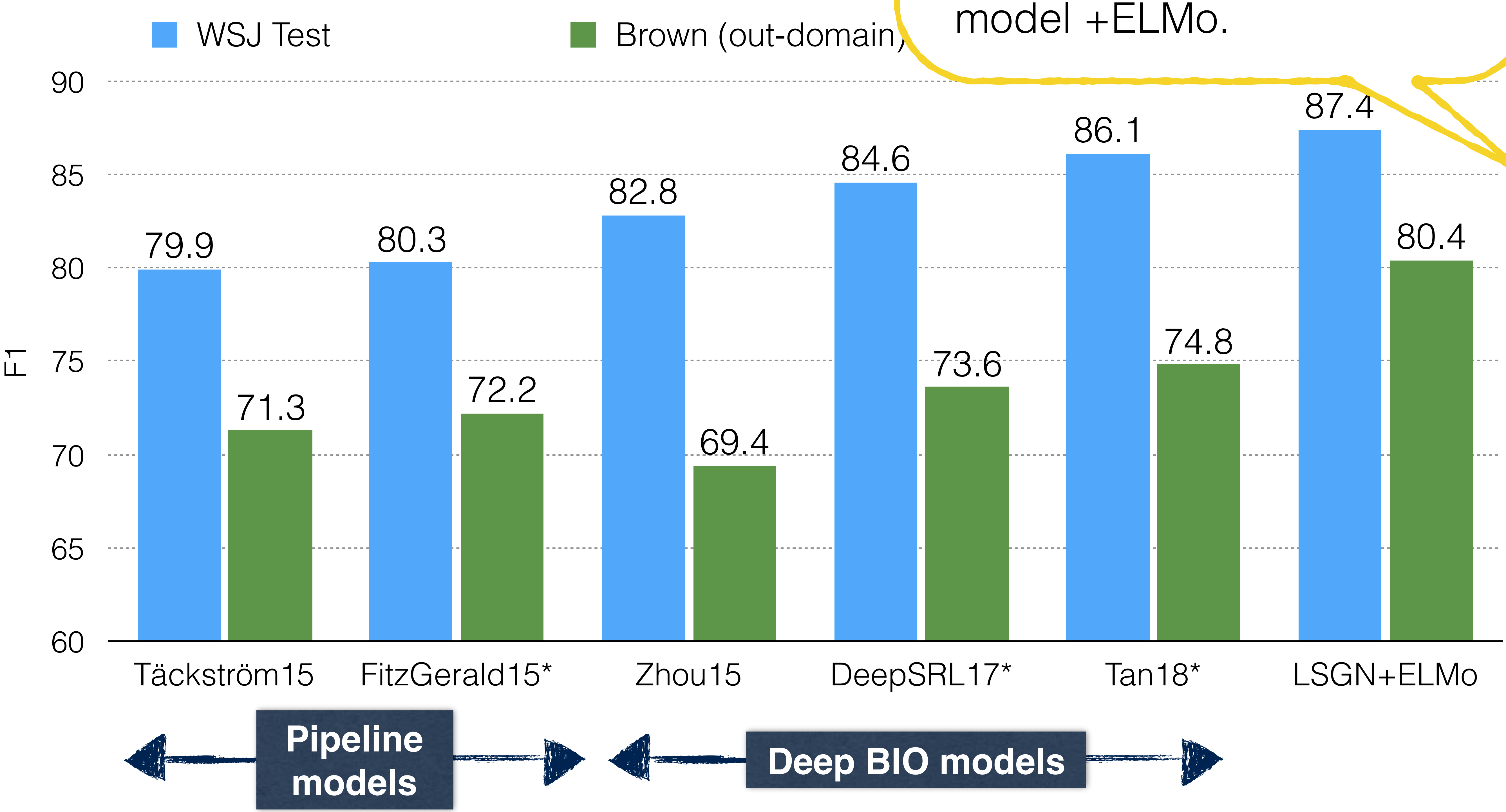


Gold Predicates: CoNLL 2005 SRL Results



Gold Predicates: CoNLL 20

- New results (Ouchi. et al, 2018) with span-selection model +ELMo.



Related Work on Span-based Models

	Nesting Spans	Span Feature
BIO-Taggers (Collobert et al., 2010, Chiu and Nichols, 2016, DeepSRL)	No	No
Semi-Markov Models (Kong et al., 2016)	No	Yes
LSGN	Yes	Yes

Span-based vs. BIO

	DeepSRL (BIO)	**LSGN (Span-based)**
Inputs	(Sentence, Predicate)	**Sentence**
Predicate Identification	Pipelined	**Joint**
Global Consistency		
Long-range Dependency		

Span-based vs. BIO

	DeepSRL (BIO)	**LSGN (Span-based)**
Inputs	(Sentence, Predicate)	**Sentence**
P	Pipelined	**Joint**
Global Consistency		
Long-range Dependency		

Due to the strong independence assumption LSGN makes

Span-based vs. BIO

	DeepSRL (BIO)	**LSGN (Span-based)**
Inputs	(Sentence, Predicate)	**Sentence**
P	ipelined	**Joint**
Global Con		
Long-range Dependency		

Due to the strong independence assumption LSGN makes

By allowing direct interaction between predicates and arguments

Outline

☑ Accurate
☑ No NLP pipeline
☑ Joint predicate ID
☐ Full-text Semantics

Predicting SRL with Deep BiLSTMs

— DeepSRL

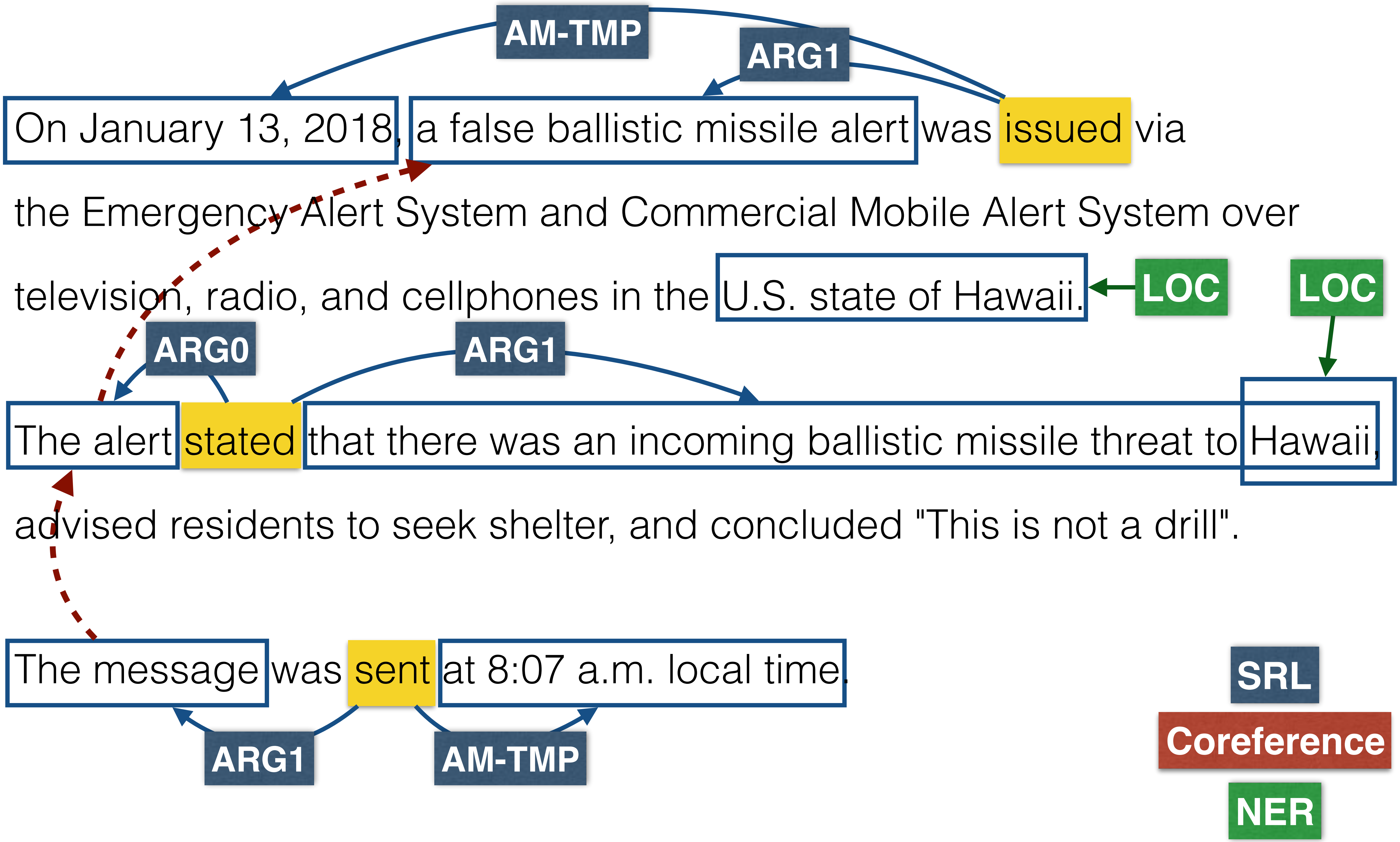
An End-to-End, Span-based SRL Model

— Labeled Span Graph Network (LSGN)

Towards Unified and Full-text Semantic Analysis

— Multi-task learning with LSGN; ScienceIE (Luan et al., 2018)

Document-level Understanding



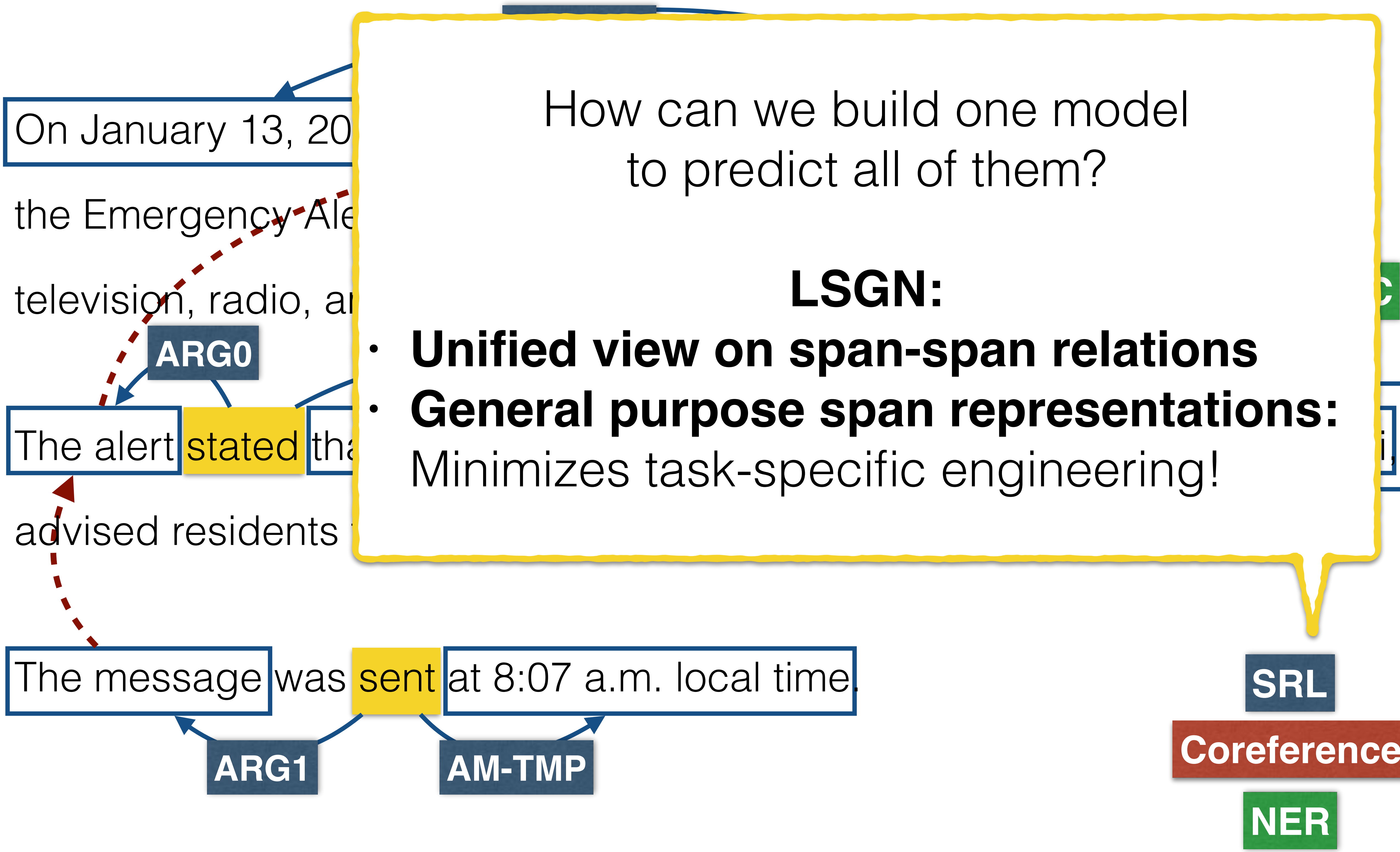
From Wikipedia: 2018 Hawaii false missile alert. Only part of the structures are visualized.

Document-level Understanding

How can we build one model to predict all of them?

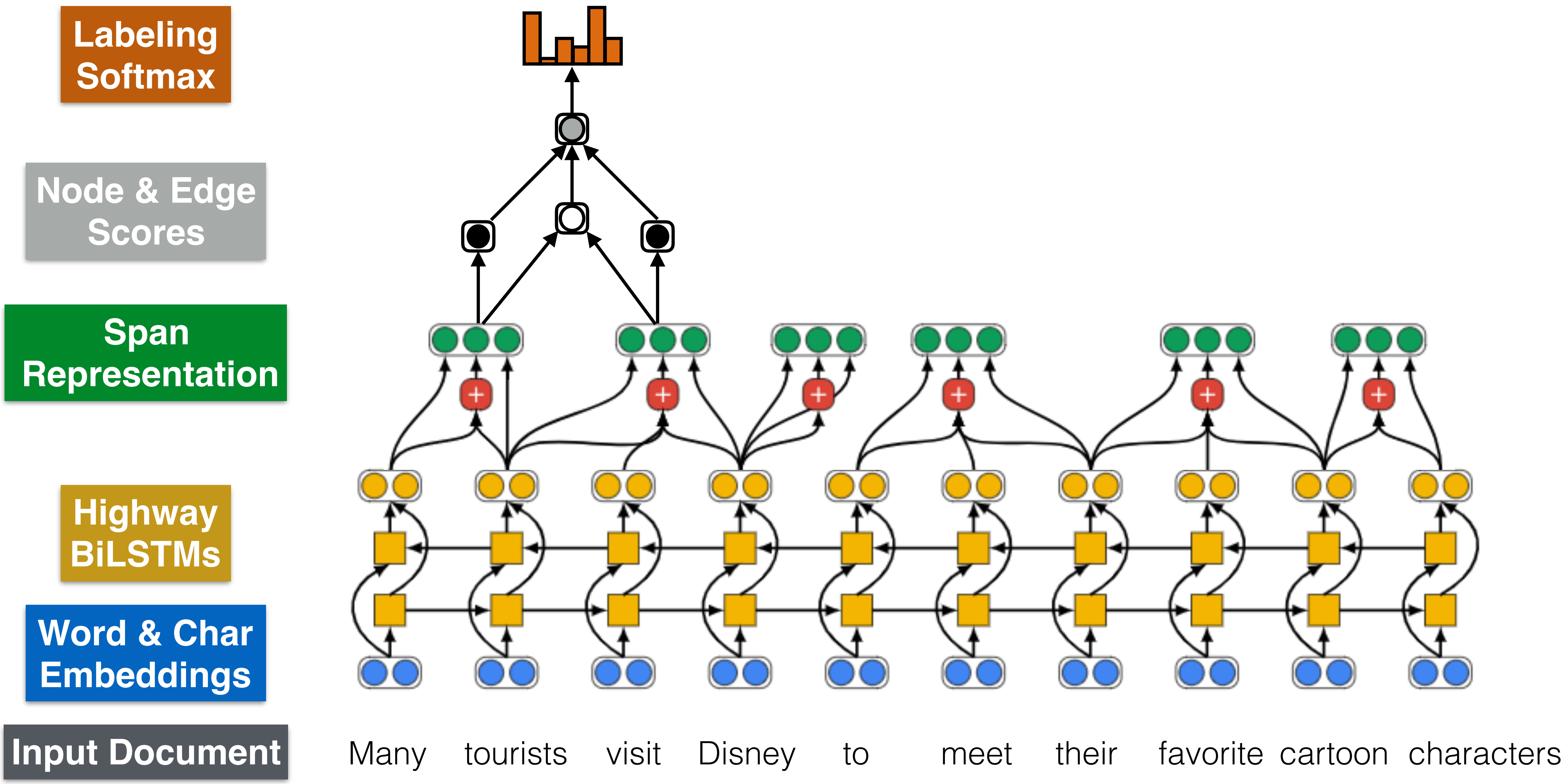
LSGN:

- **Unified view on span-span relations**
- **General purpose span representations:** Minimizes task-specific engineering!

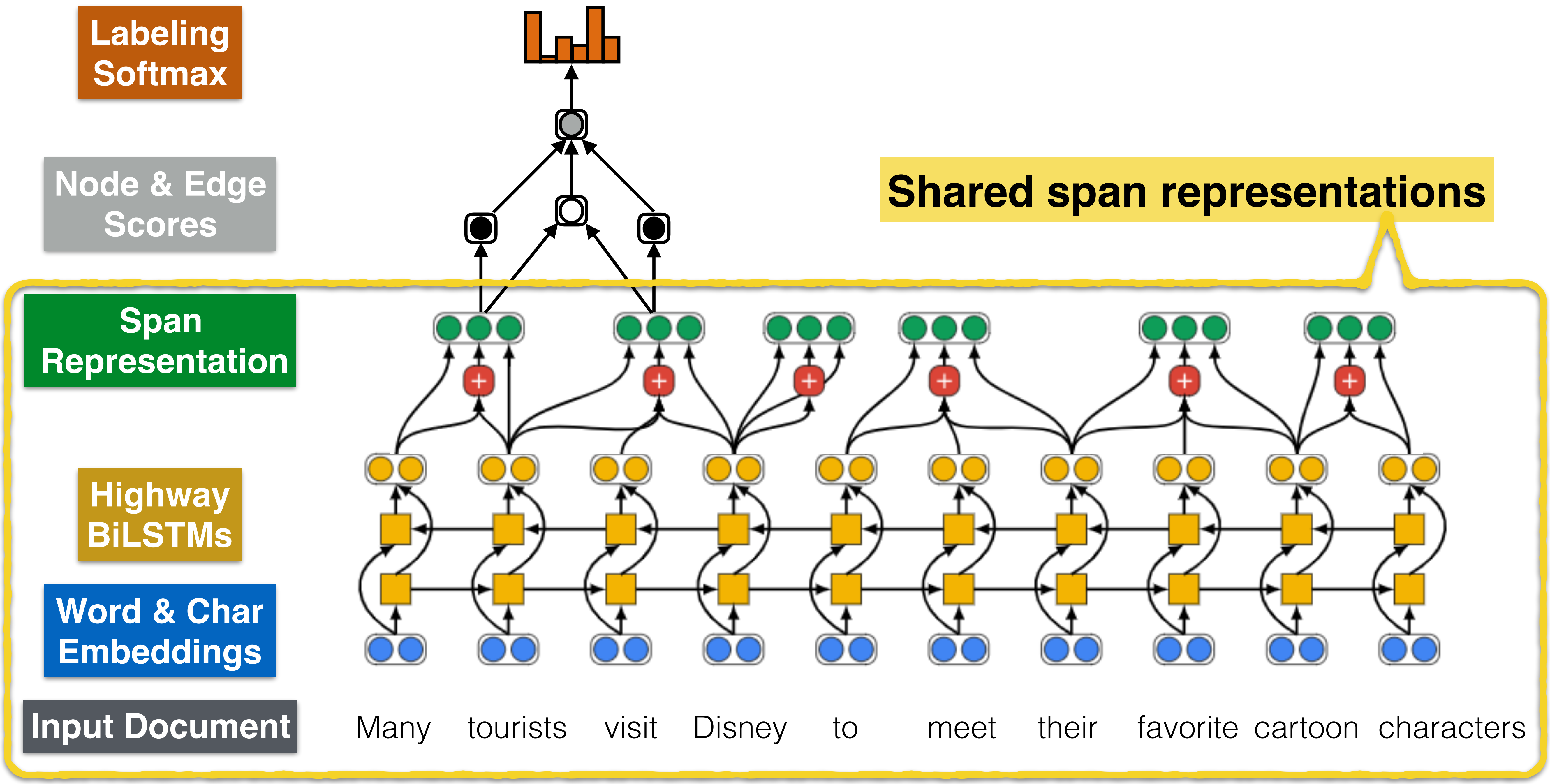


From Wikipedia: 2018 Hawaii false missile alert. Only part of the structures are visualized.

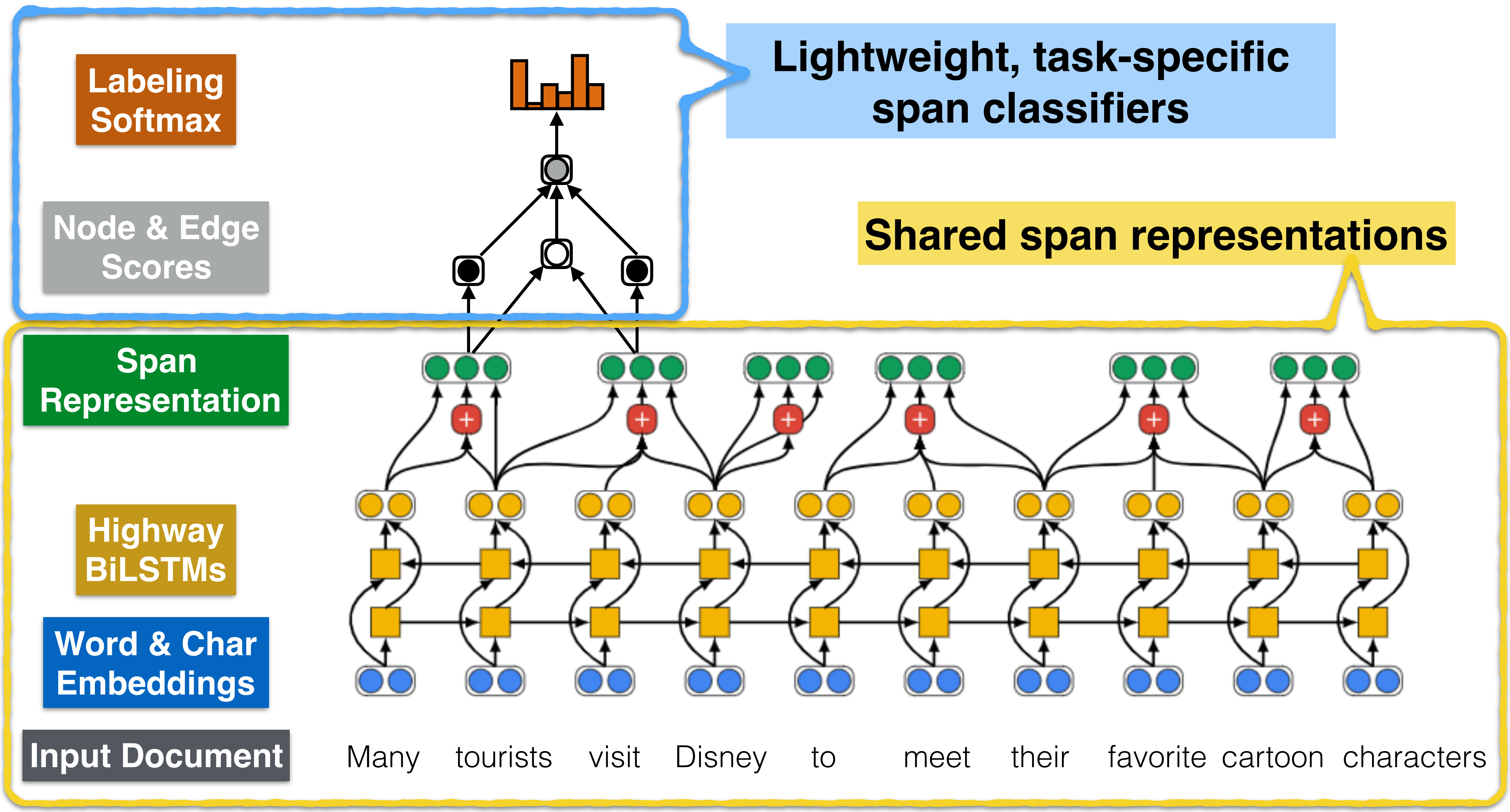
Multi-task LSGN Architecture



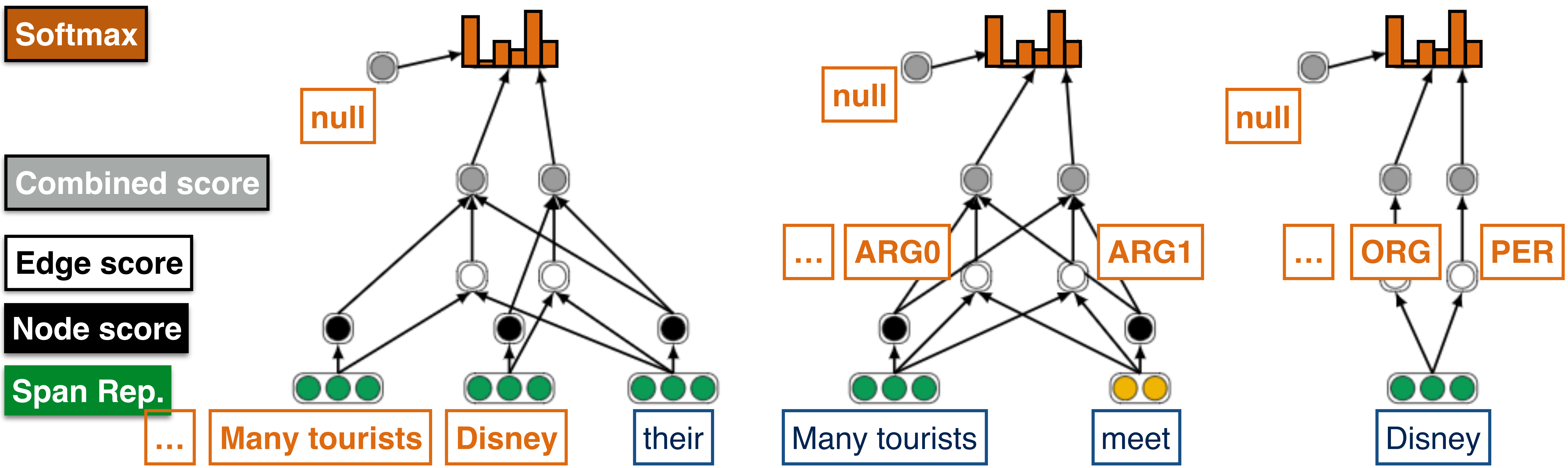
Multi-task LSGN Architecture



Multi-task LSGN Architecture



Task-specific Span Classifiers

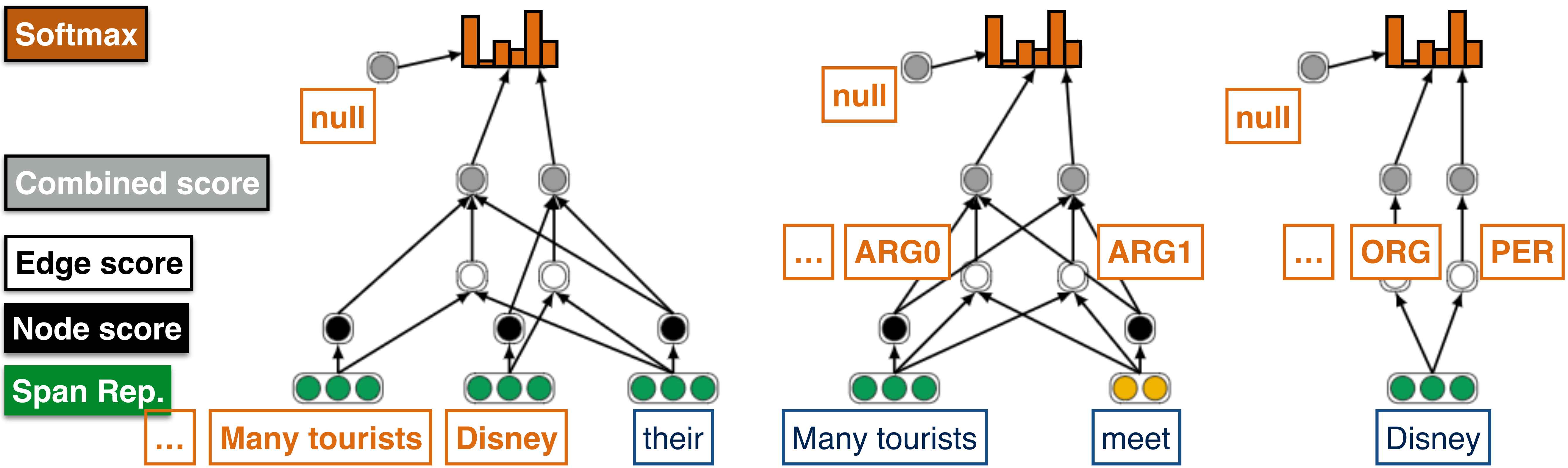


Coreference
(Lee et al., 2017)

SRL/Relation Extraction

NER

Task-specific Span Classifiers



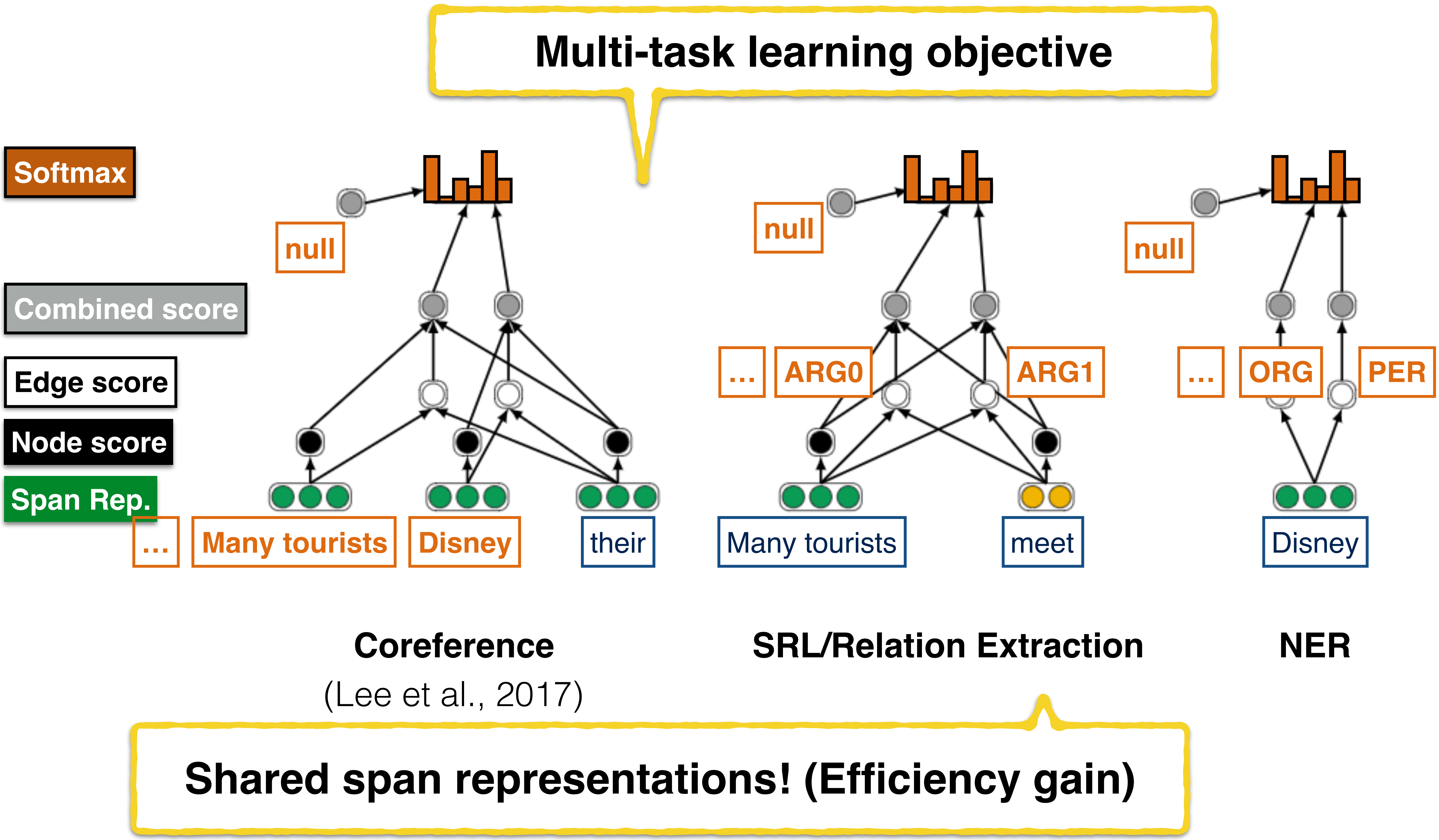
Coreference (Lee et al., 2017)

SRL/Relation Extraction

NER

Shared span representations! (Efficiency gain)

Task-specific Span Classifiers



Two multi-tasking setups

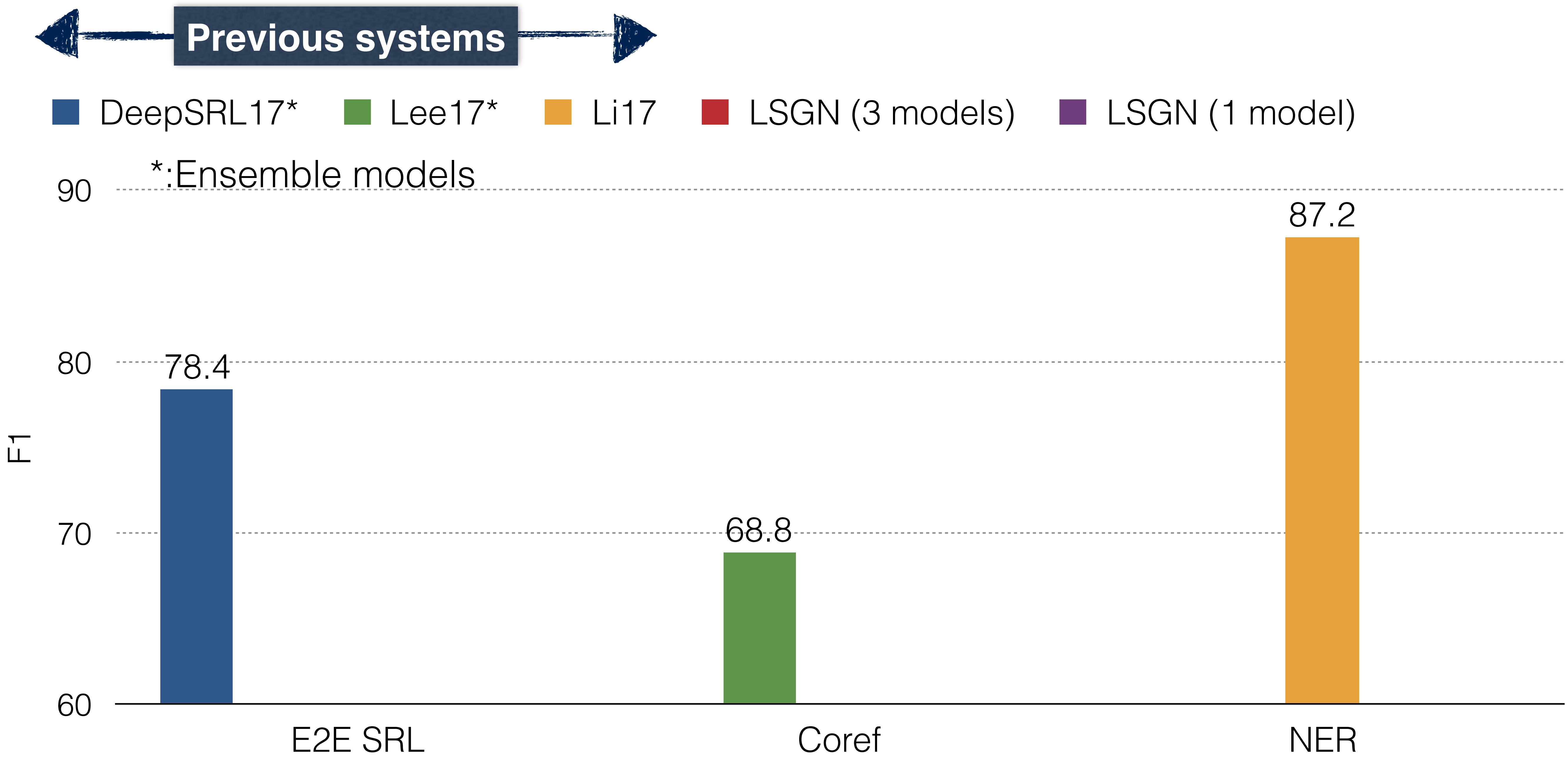
1. "***One model for everything***":
   Train ***one*** model to predict all the ***n*** tasks. Performance is tuned on the average of the n metrics …

Two multi-tasking setups

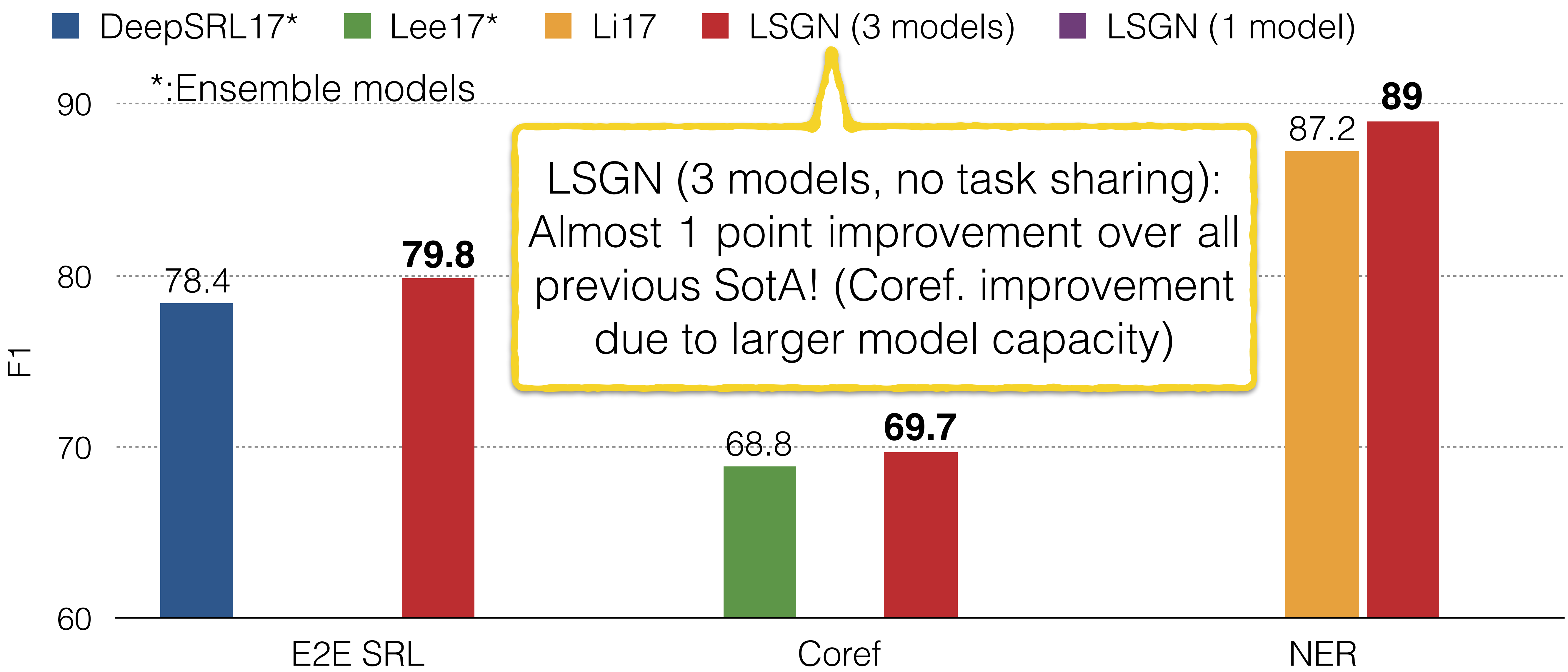
1. "***One model for everything***":
   Train ***one*** model to predict all the ***n*** tasks. Performance is tuned on the average of the n metrics …

2. "***Let the tasks help each other***":
   Train ***n*** models, each predicts a ***target*** task, and treat the rest n-1 tasks as ***auxiliary***. (Swayamdipta et al., 2018 …)

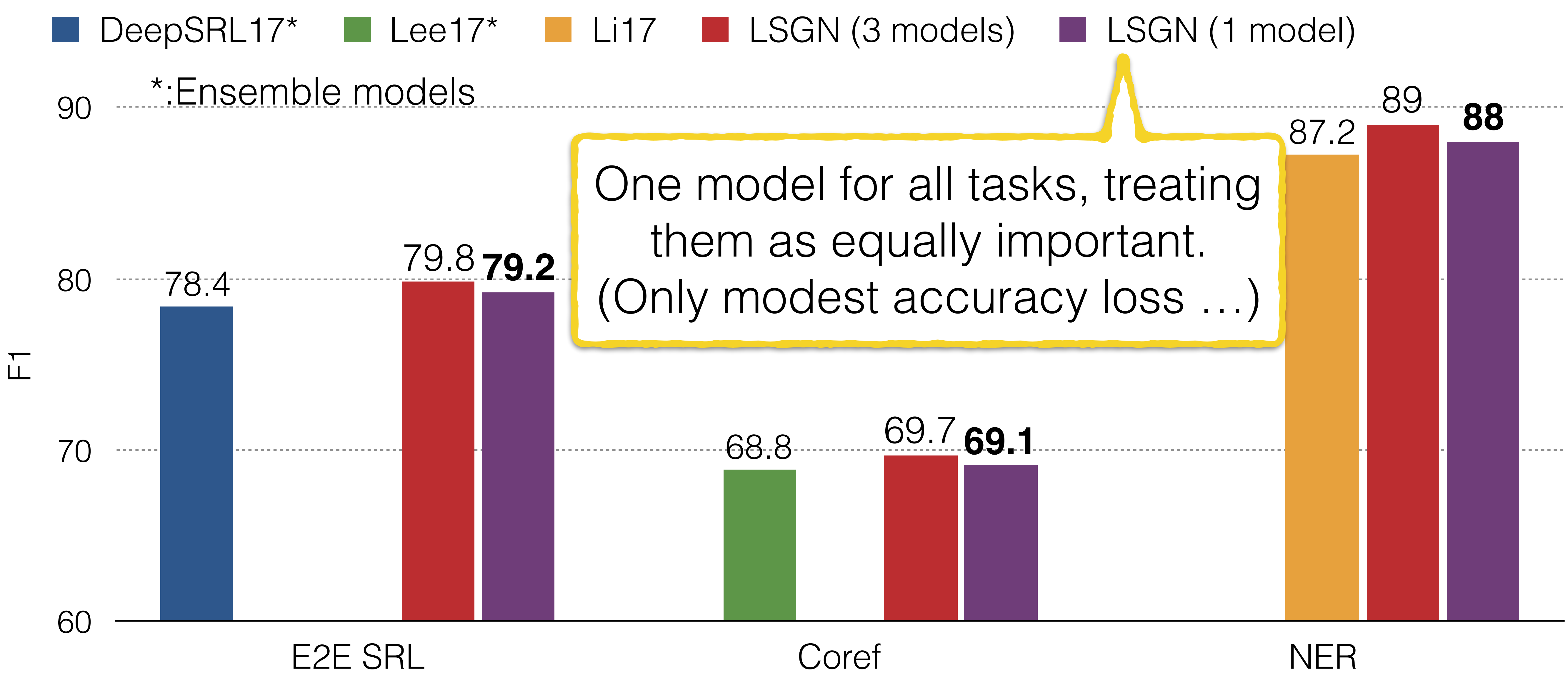
OntoNotes: Is it possible to have one model to do them all?



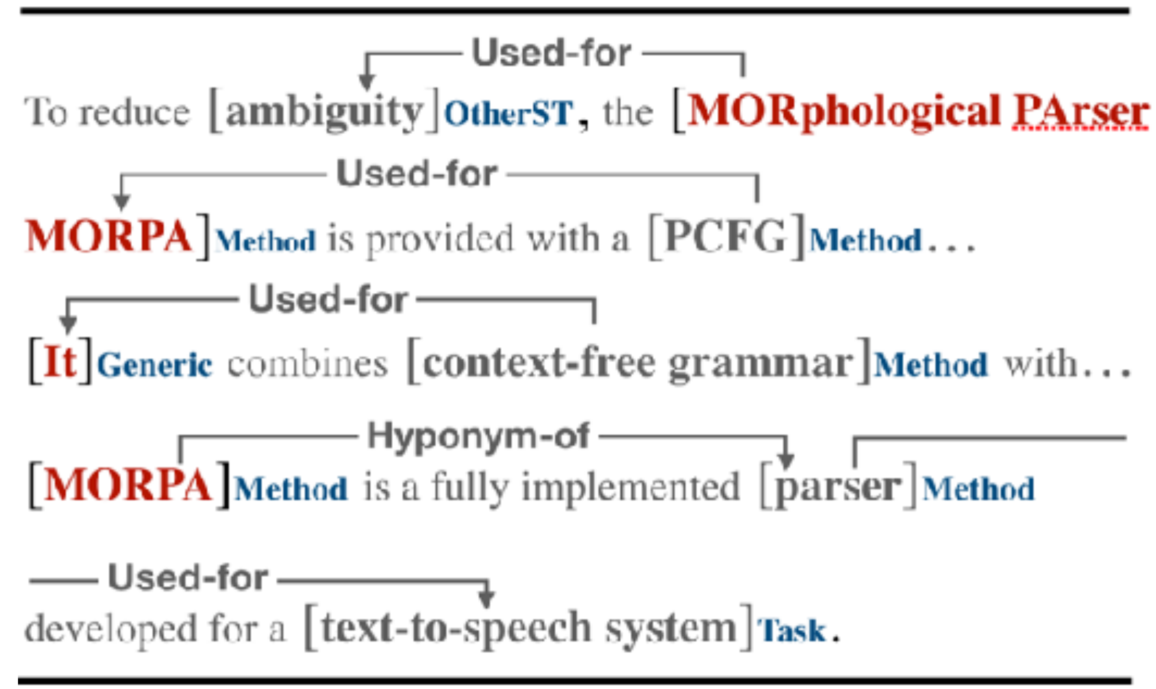
OntoNotes: Is it possible to have one model to do them all?

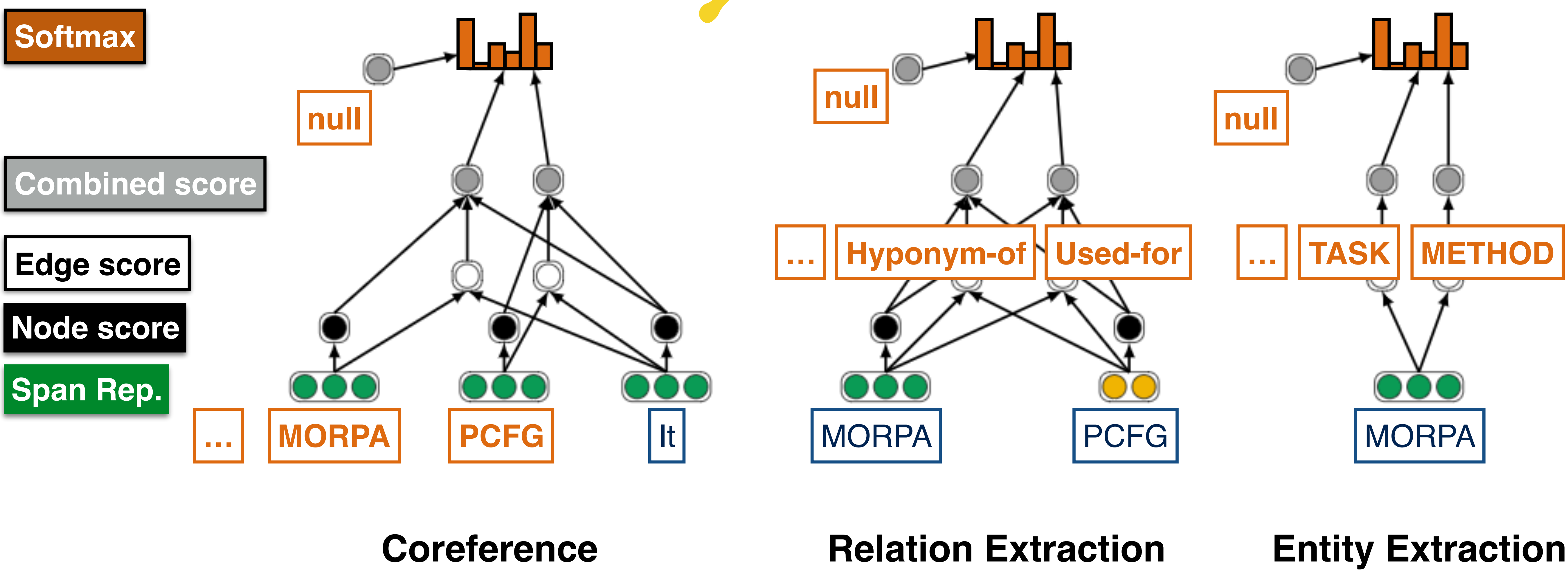


OntoNotes: Is it possible to have one model to do them all?

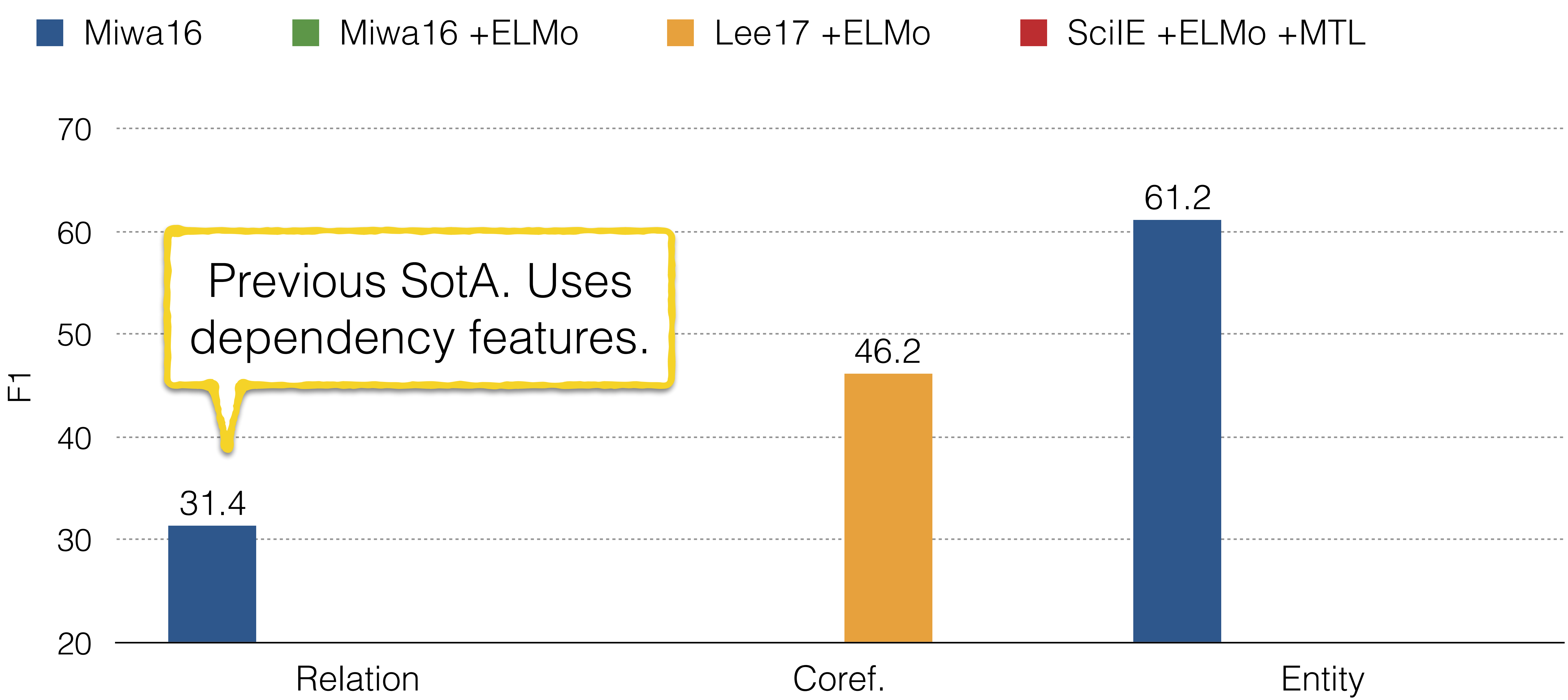


ScienceERC Data (Luan et al., 2018): Can the tasks help each other?

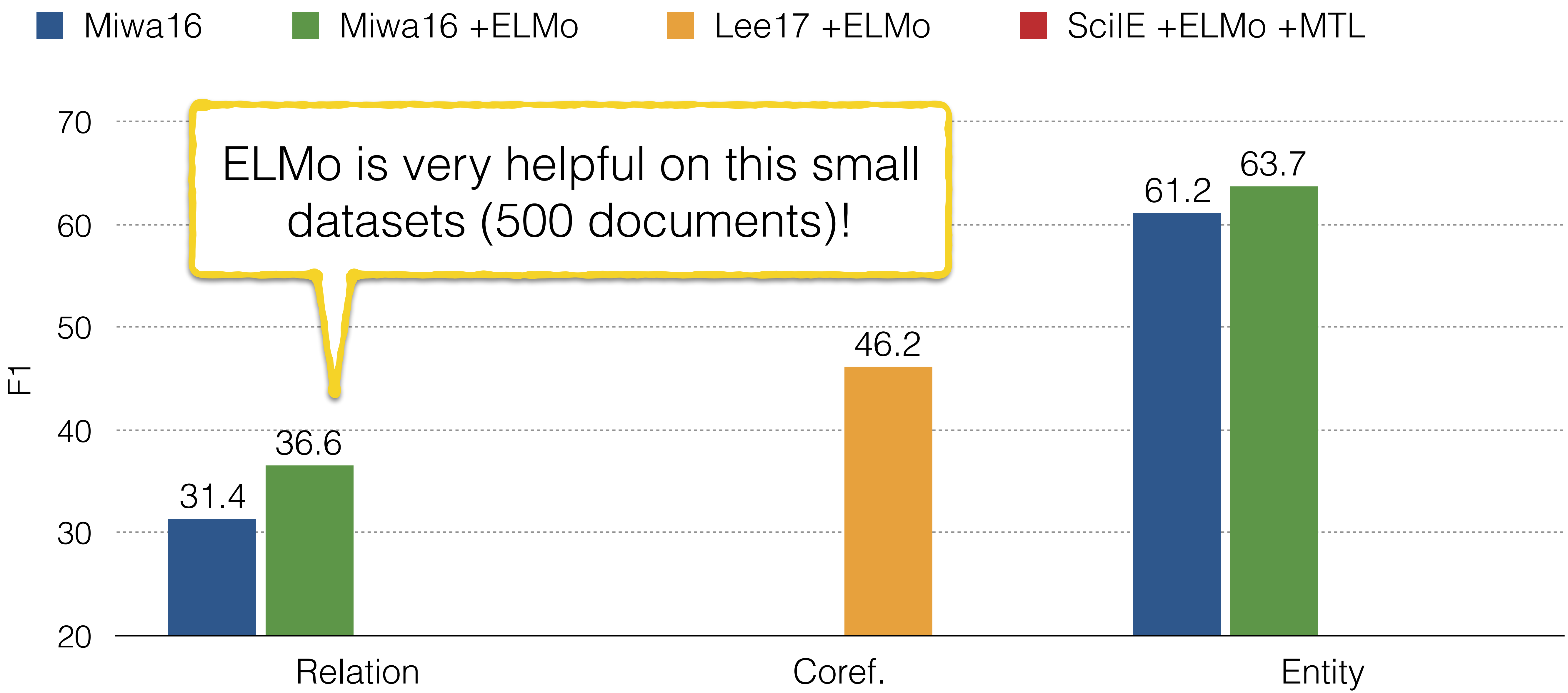




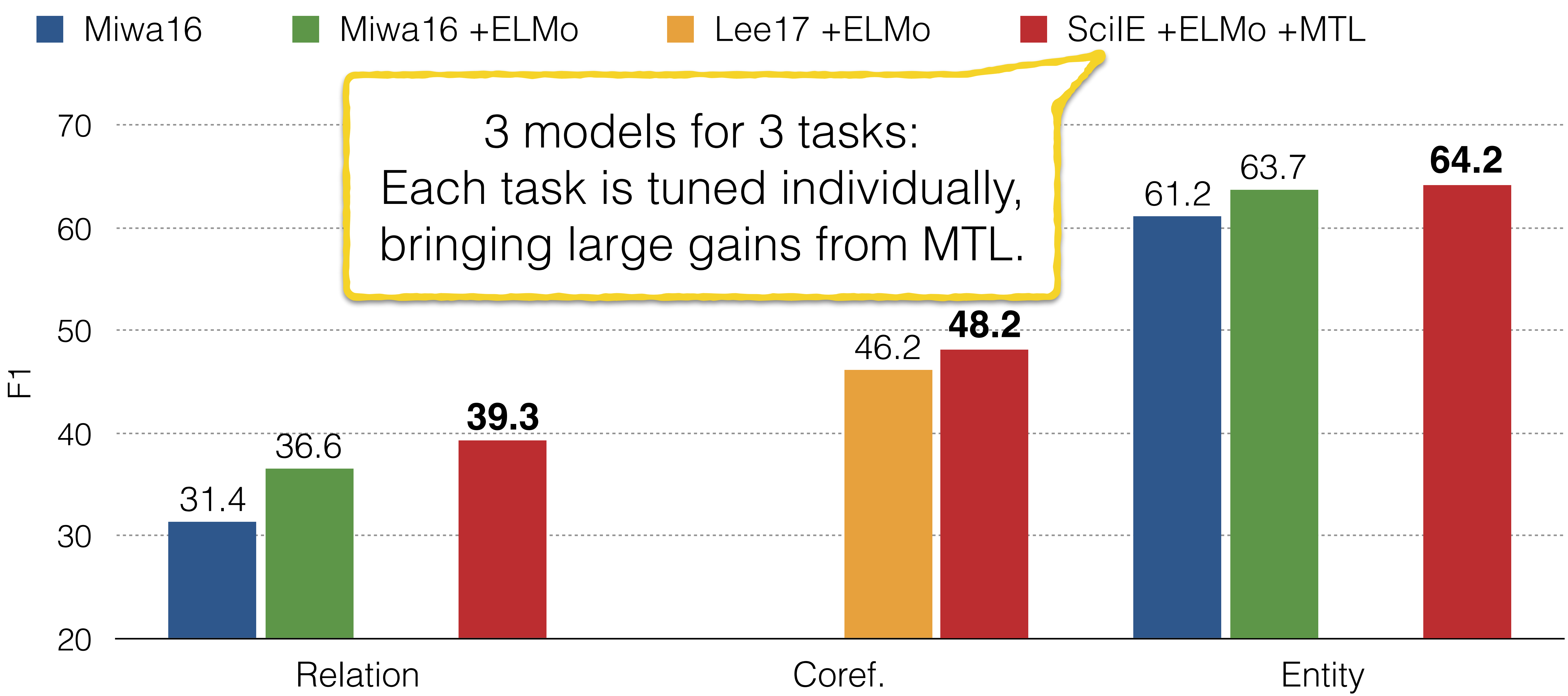
ScienceERC: Can the tasks help each other?



ScienceERC: Can the tasks help each other?

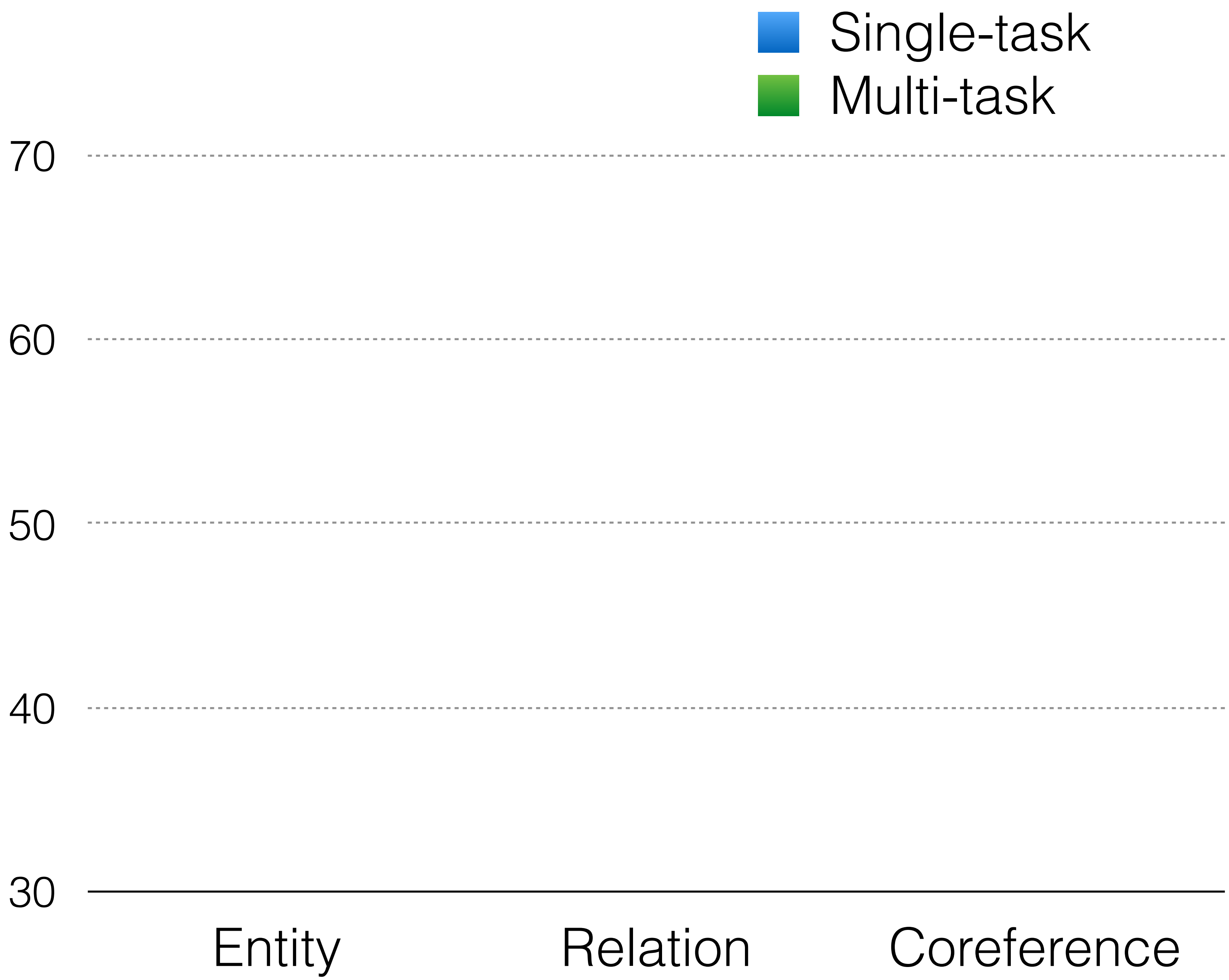


ScienceERC: Can the tasks help each other?

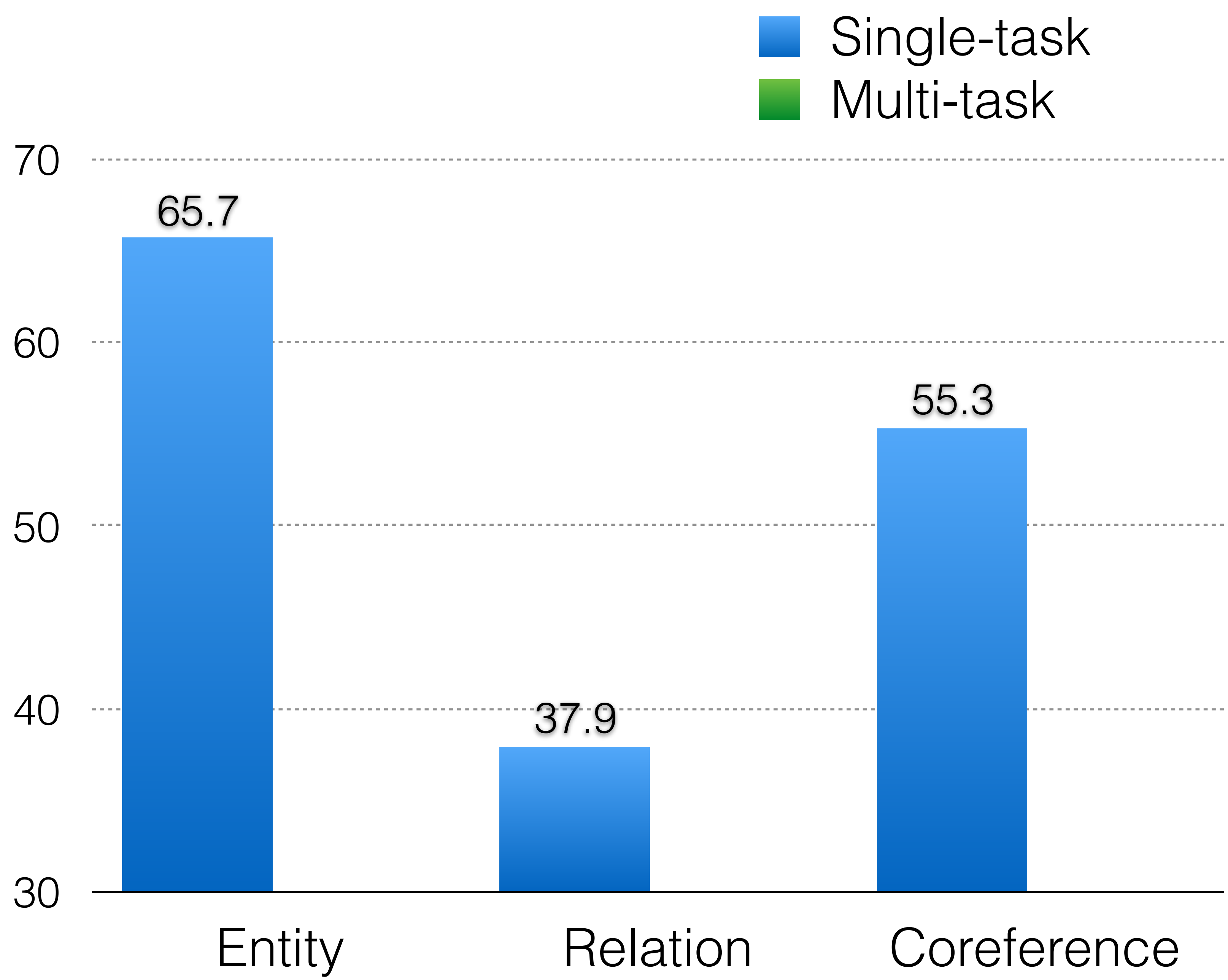


ScienceERC: Ablations

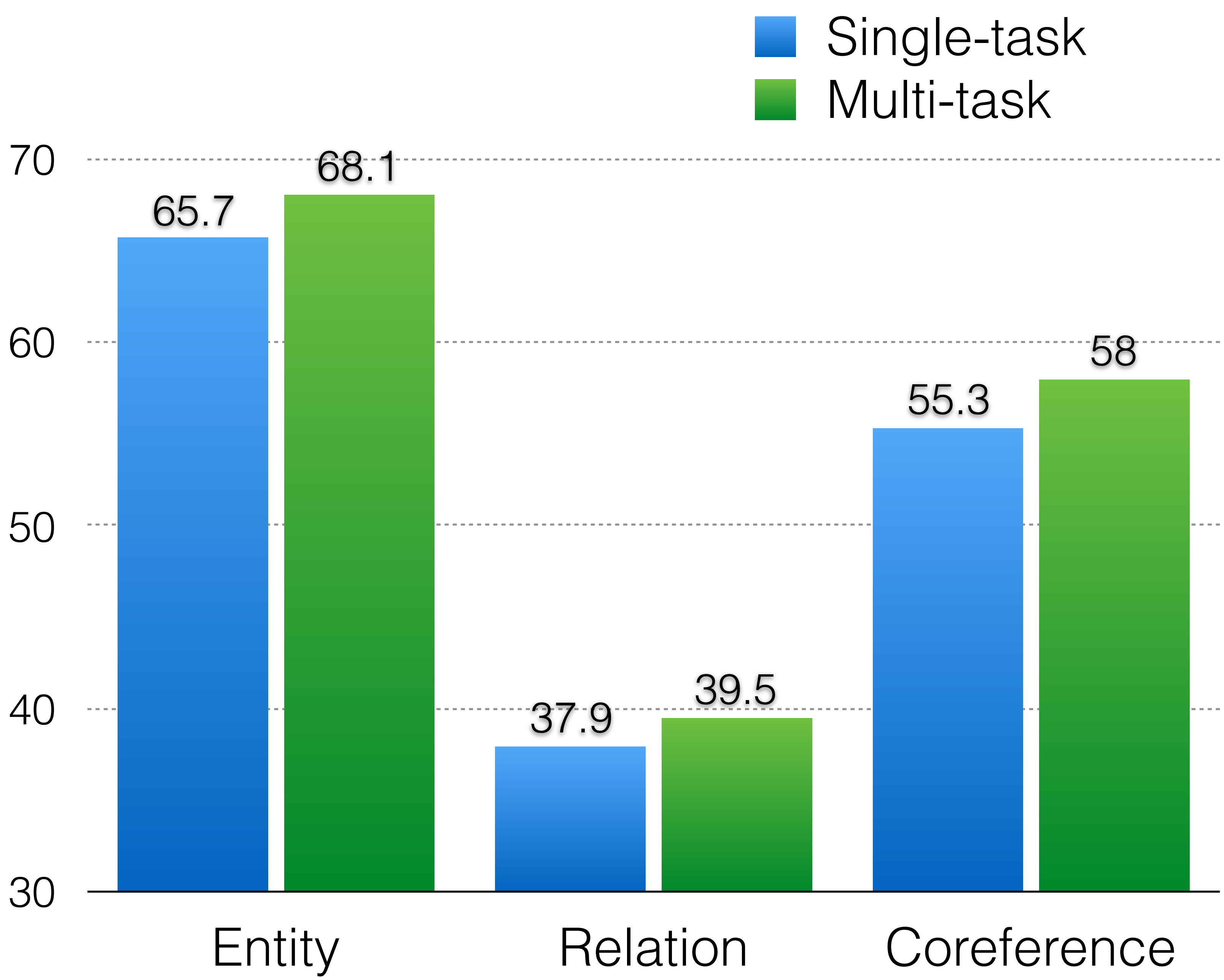
ScienceERC: Ablations



ScienceERC: Ablations

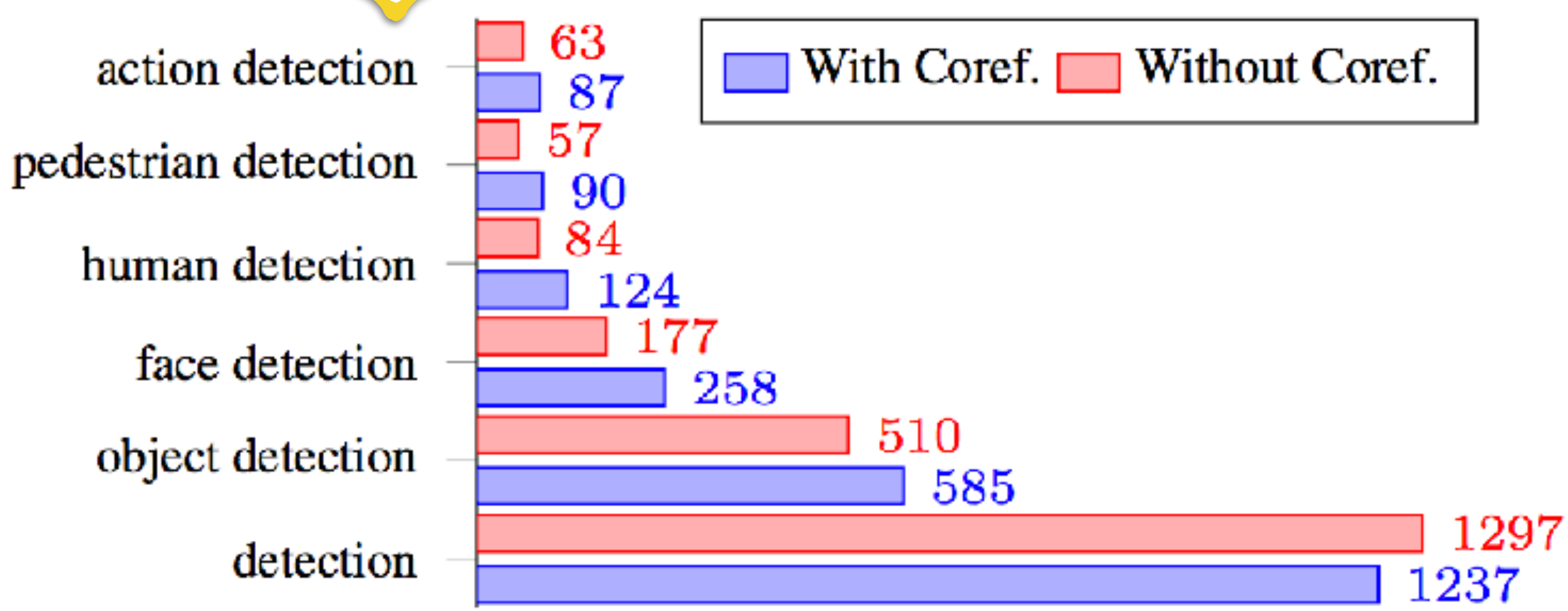


ScienceERC: Ablations



ScienceERC: Qualitative Analysis

With predicted coreference links, the system extracted less generic terms and more specific ones!

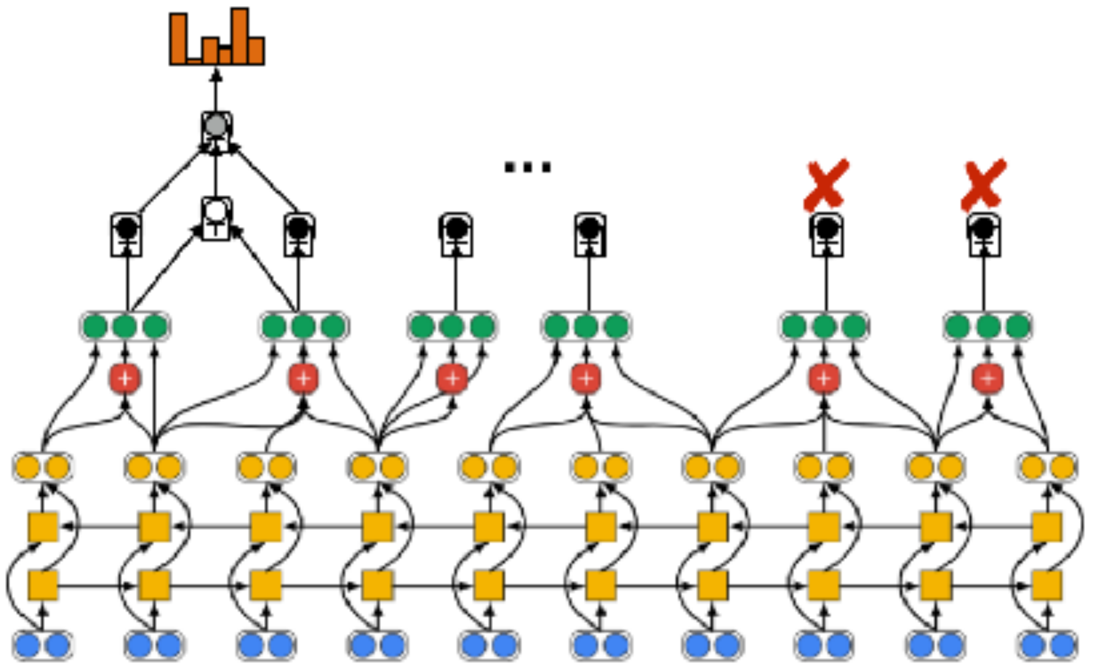


Conclusion

- A general framework for a variety of tasks.

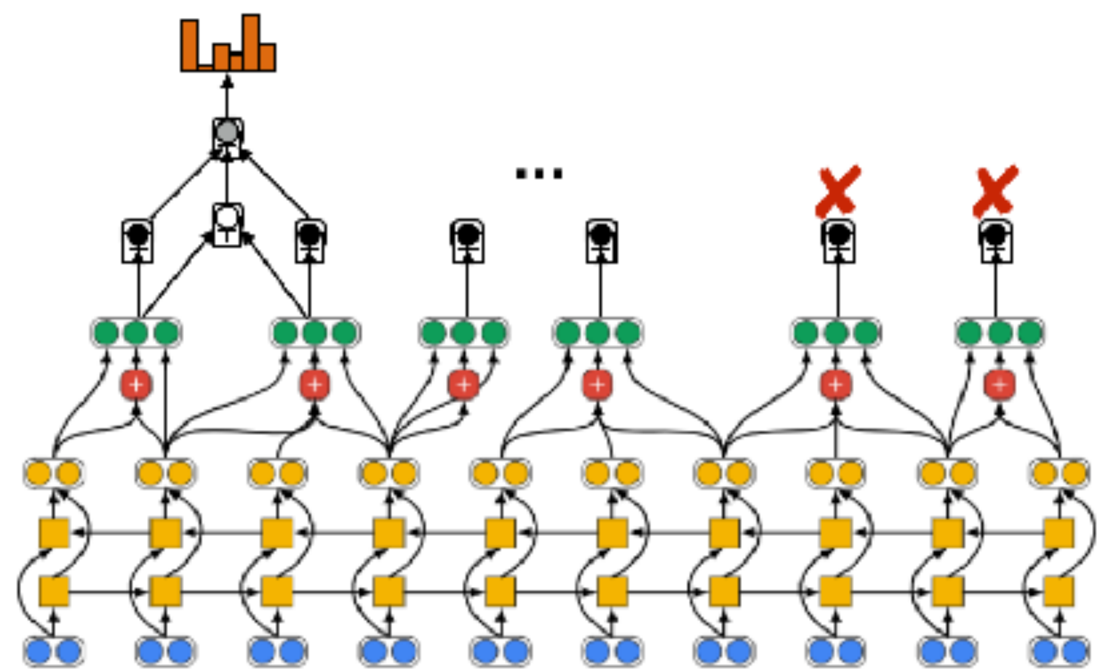
Conclusion

- A general framework for a variety of tasks.
- Our recipe:
  1. Contextualized span representations
  2. Local label classifiers
  3. Greedy span pruning



Conclusion

- A general framework for a variety of tasks.
- Our recipe:
  1. Contextualized span representations
  2. Local label classifiers
  3. Greedy span pruning



- Multi-task learning works (sometimes)

Future Work

1. LSGNs for more NLP tasks.
2. Improve global consistency of the LSGN outputs with joint inference (e.g. Singh et al., 2013).
3. Pre-train transferrable span embeddings.

Links to Code and Data

1. DeepSRL:

   https://github.com/luheng/deep_srl

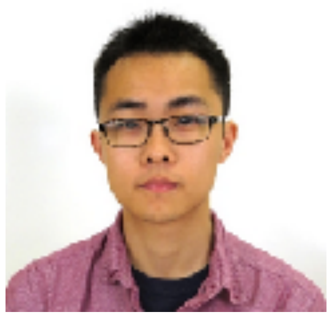
2. LSGN:

   https://github.com/luheng/lsgn

3. SciIE/ScienceERC (by Yi Luan):

   http://nlp.cs.washington.edu/sciIE/

Many Thanks to my Collaborators!





in collaboration with Kenton Lee, Mike Lewis, Omer Levy, Yi Luan, and Luke Zettlemoyer