End-to-end Learning for Broad Coverage Semantics

Luke Zettlemoyer[†*]

Joint work with **Luheng He**[†], **Kenton Lee**[†], **Matthew Peters***, Christopher Clark[†], Matthew Gardner*, Mohit Iyyer***,** Mandar Joshi[†], Mike Lewis[‡], Julian Michael[†], Mark Neumann*

[†] Paul G. Allen School of Computer Science & Engineering, University of Washington,
[‡] Facebook AI Research
[*] Allen Institute for Artificial Intelligence

PAUL G. ALLEN SCHOOL OF COMPUTER SCIENCE & ENGINEERING

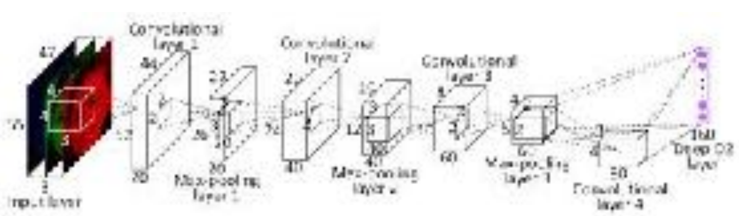
ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

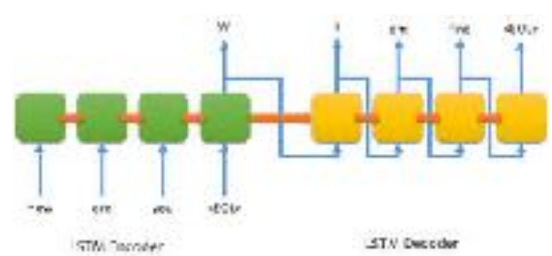
Three Simple Steps that will Revolutionize Your ML Research

Step 1: Gather lots of training data!

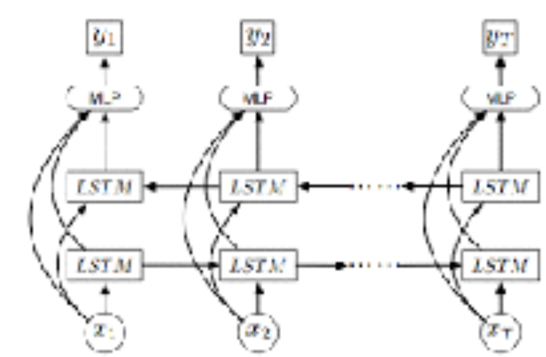
IMAGENET European Parliament twitter . . .

Step 2: Apply Deep Learning!!





. . .



Step 3: Observe Impressive Gains!!!

Broad Coverage Semantics

Example Tasks:

Coreference: clustering NPs

A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building.

Semantic Role Labeling: who did what, etc.

ARG0	NASA
PRED	<u>*observe*</u>
ARG1	an X-ray flare 400 times brighter than usual
TMP	On January 5, 2015

Many applications:

Question Answering

Information Extraction

Machine Translation

Does the Recipe Work for Broad Coverage Semantics?

Step 1: Gather lots of training data!

Challenge 1: Data is costly and limited (e.g. linguists required to label PennTreebank / OntoNotes)

Step 2: Apply Deep Learning!!

Challenge 2: Pipeline of structured prediction problems with cascading errors (e.g. POS->Parsing->SRL->Coref)

Step 3: Observe Impressive Gains!!!

New Learning Approaches

New state-of-the-art results for two tasks:

Coreference:

A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building.

Semantic Role Labeling:

ARG0	NASA
PRED	*observe*
ARG1	an X-ray flare 400 times brighter than usual
TMP	On January 5, 2015

Common themes:

- End-to-end training of deep neural networks
- No preprocessing (e.g., no POS, no parser, etc.)
- Large gains in accuracy with simpler models and no extra training data (*up to 40% relative error reductions!*)

Coreference Resolution

Input document
A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building.

Coreference Resolution

Input document
A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building.

Cluster #1	A fire in a Bangladeshi garment factory	the blaze in the four-story building

Coreference Resolution

Input document
A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building.

Cluster #1	A fire in a Bangladeshi garment factory	the blaze in the four-story building
Cluster #2	a Bangladeshi garment factory	the four-story building

Coreference Resolution

Input document
A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building.

Cluster #1	A fire in a Bangladeshi garment factory	the blaze in the four-story building
Cluster #2	a Bangladeshi garment factory	the four-story building
Cluster #3	at least 37 people	the deceased

Two Subproblems

Input document
A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building.

Mention detection

A fire in a Bangladeshi garment factory
at least 37 people
…
the four-story building

Mention clustering

Cluster #1	A fire in a Bangladeshi garment factory	the blaze in the four-story building
Cluster #2	a Bangladeshi garment factory	the four-story building
Cluster #3	at least 37 people	the deceased

Previous Approach: Rule-based pipeline

Input document

A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized.

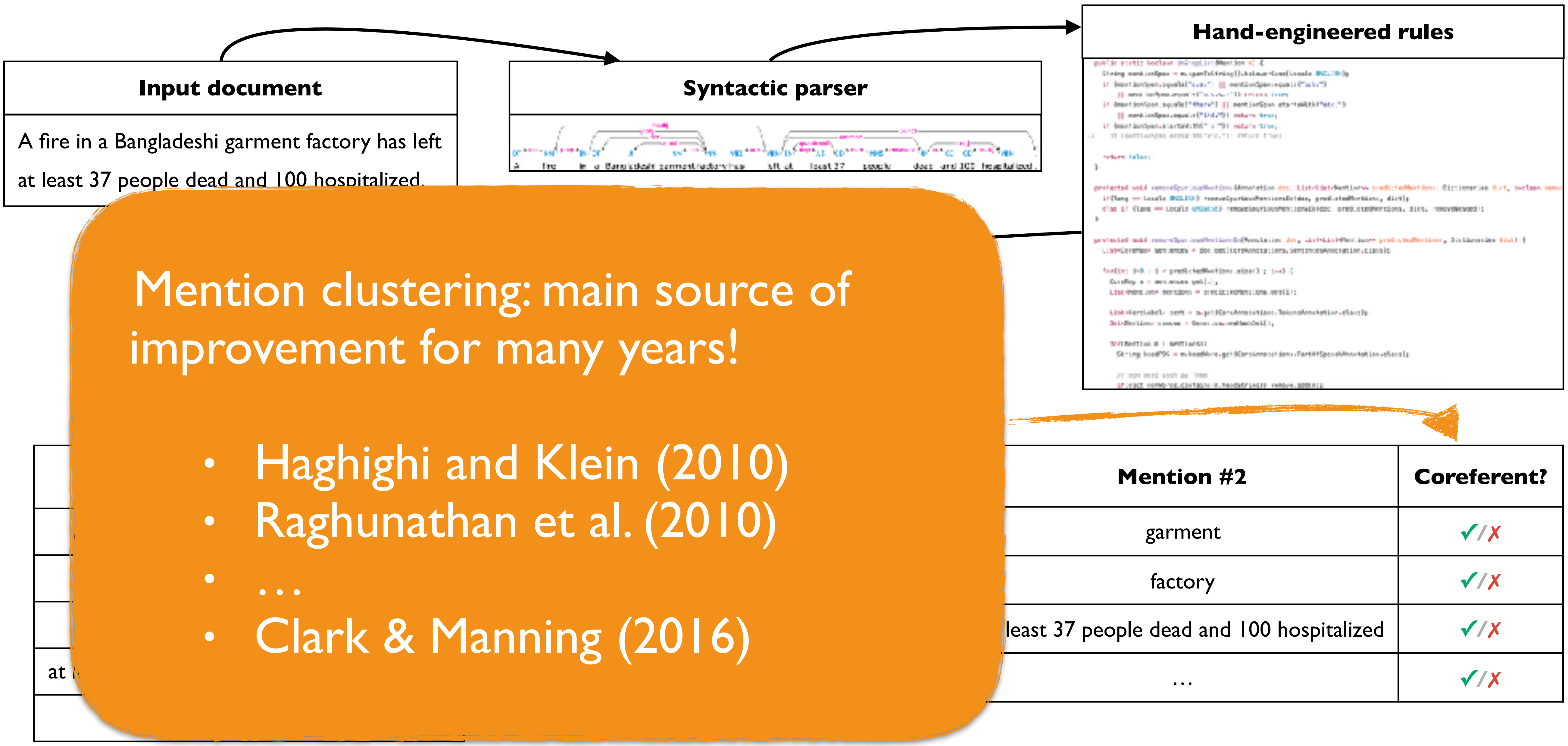
Syntactic parser

Hand-engineered rules

Candidate mentions
A fire in a Bangladeshi garment factory
garment
factory
at least 37 people dead and 100 hospitalized
…

Mention #1	Mention #2	Coreferent?
A fire in a Bangladeshi garment factory	garment	✓/✗
garment	factory	✓/✗
factory	at least 37 people dead and 100 hospitalized	✓/✗
…	…	✓/✗

Previous Approach: Rule-based pipeline



Mention clustering: main source of improvement for many years!

- Haghighi and Klein (2010)
- Raghunathan et al. (2010)
- …
- Clark & Manning (2016)

Previous Approach: Rule-based pipeline

Input document

A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized.

Syntactic parser

Hand-engineered rules

Relies on parser for:

- mention detection
- syntactic features for clustering (e.g. head words)

A fire in a Bangladeshi garment factory
garment
factory
at least 37 people dead and 100 hospitalized
…

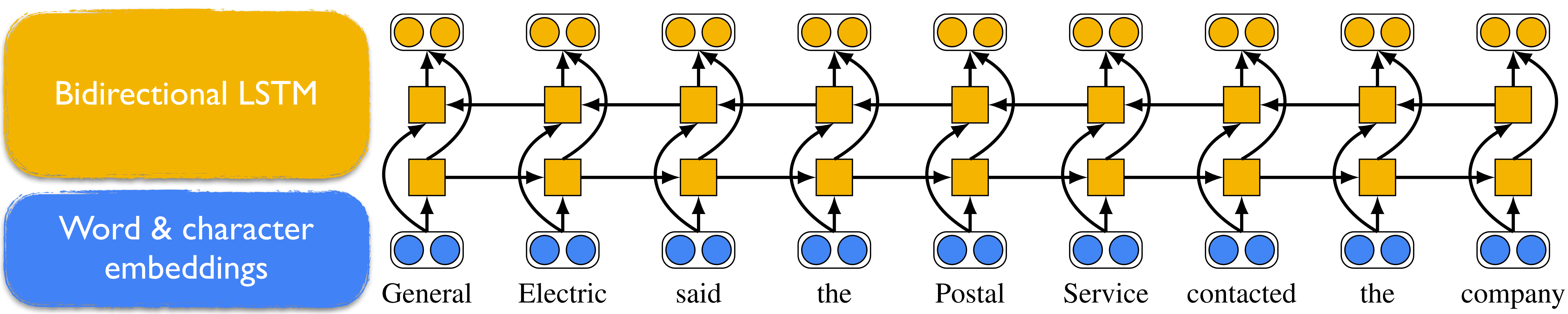
		Coreferent?
A fire in a Bangladeshi garment factory	garment	✓/✗
garment	factory	✓/✗
factory	at least 37 people dead and 100 hospitalized	✓/✗
…	…	✓/✗

End-to-end Approach

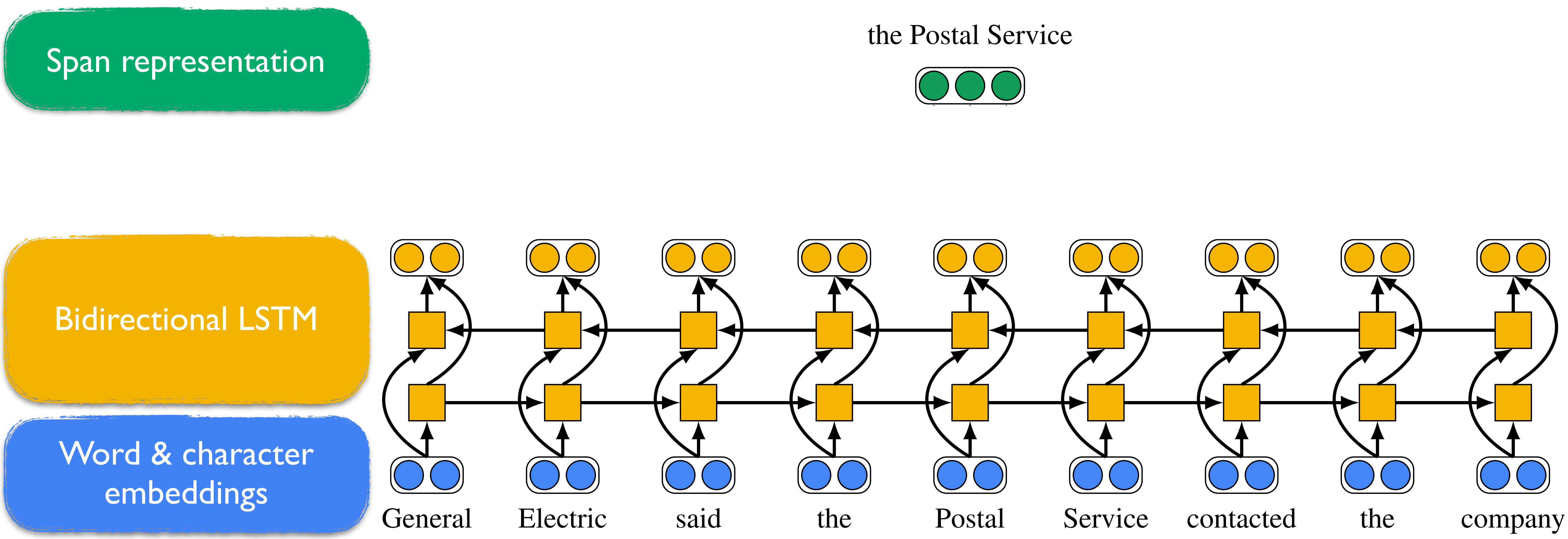
- Consider all possible text spans
- Learn to rank antecedent spans
- Factored model to prune search space

[Lee et al, 2017]

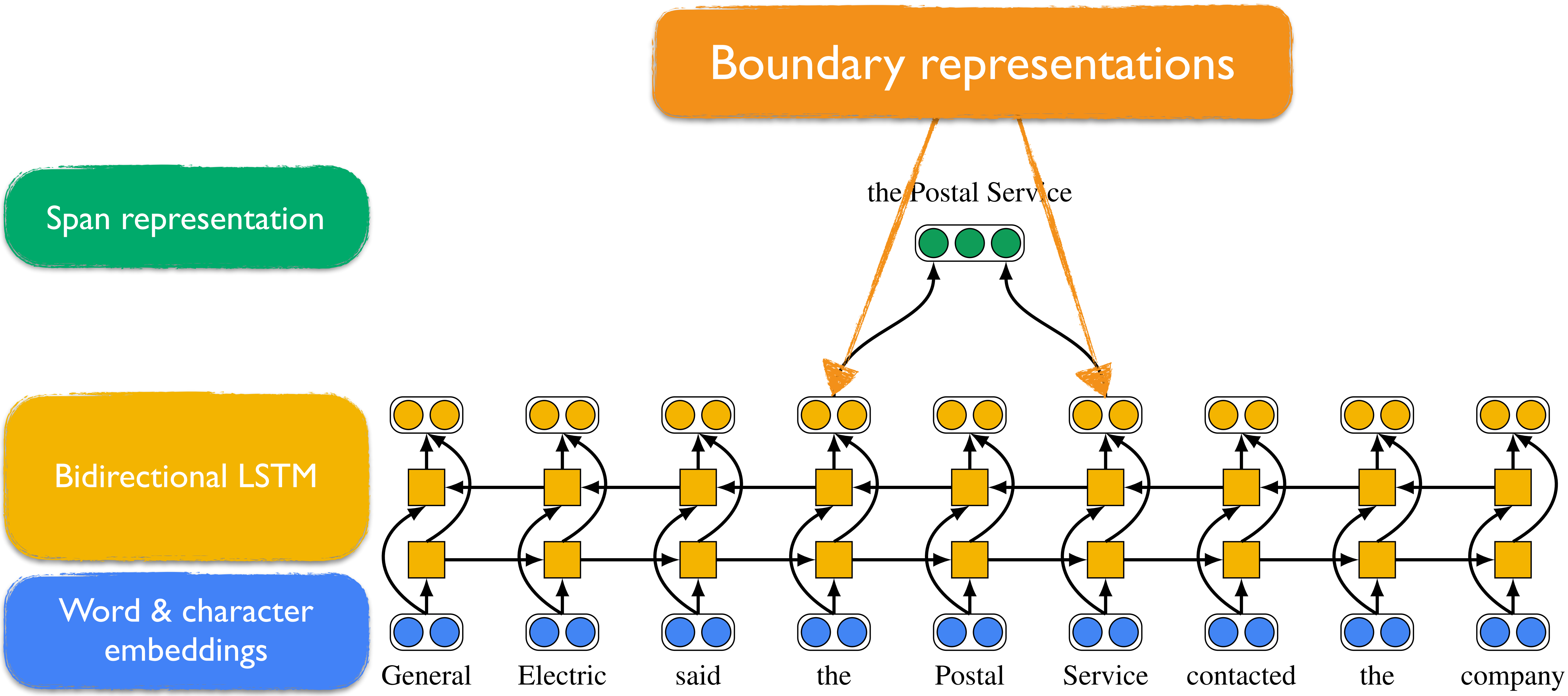
Key Idea: Span Representations



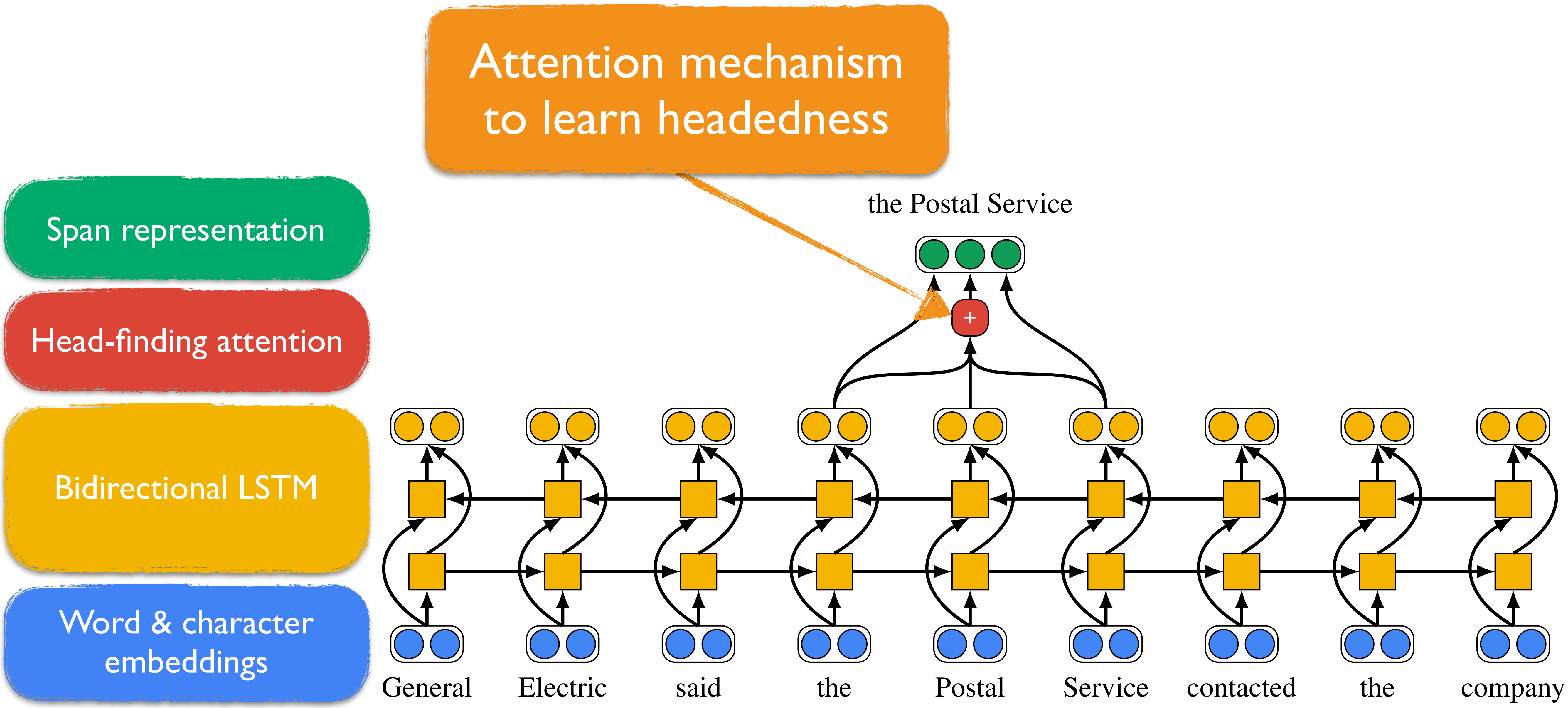
Key Idea: Span Representations



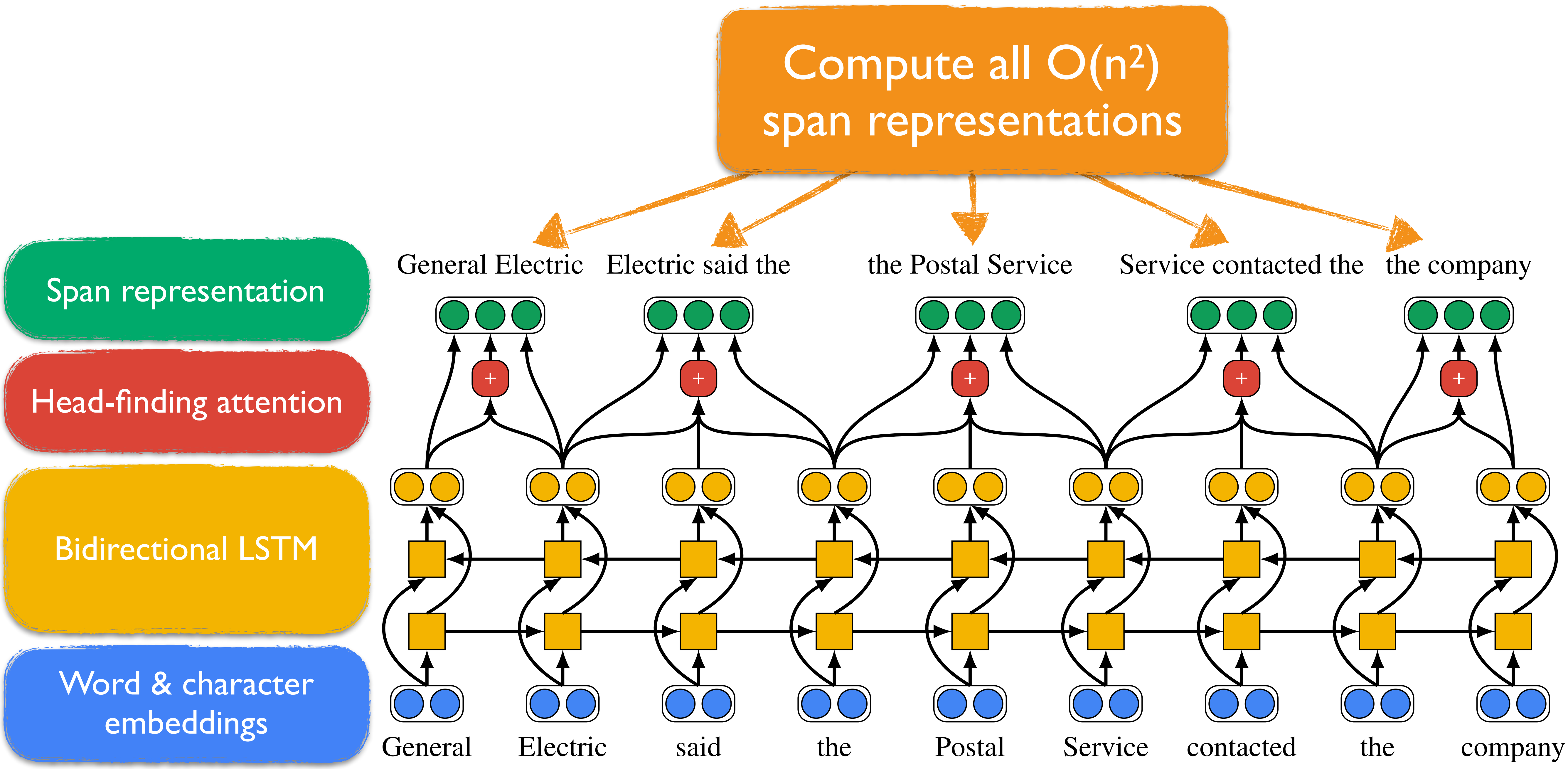
Key Idea: Span Representations



Key Idea: Span Representations



Key Idea: Span Representations



Mention Ranking

Every span independently chooses an antecedent

Input document

A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building. Witnesses say the only exit door was on the ground floor, and that it was locked when the fire broke out.

Mention Ranking

- Reason over all possible spans

- Assign an antecedent to every span

$$y_3 \in \{\epsilon, 1, 2\}$$

	Span	**Antecedent**
1	A	y_1
2	A fire	y_2
3	A fire in	y_3
…	…	…
M	out	y_M

Mention Ranking

- Reason over all possible spans
- Assign an antecedent to every span

$$y_3 \in \{\epsilon, 1, 2\}$$

ϵ : no coreference link

	Span	Antecedent
1	A	y_1
2	A fire	y_2
3	A fire in	y_3
…	…	…
M	out	y_M

Mention Ranking

- Reason over all possible spans
- Assign an antecedent to every span

	Span	Antecedent
1	A	y_1
2	A fire	y_2
3	A fire in	y_3
…	…	…
M	out	y_M

$$y_3 \in \{\epsilon, 1, 2\}$$

Coreference link from span 1 to span 3

Mention Ranking

- Reason over all possible spans
- Assign an antecedent to every span

	Span	Antecedent
1	A	y_1
2	A fire	y_2
3	A fire in	y_3
…	…	…
M	out	y_M

$$y_3 \in \{\epsilon, 1, 2\}$$

Coreference link from span 2 to span 3

Example Clustering

Input document
A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building. Witnesses say the only exit door was on the ground floor, and that it was locked when the fire broke out.

Span	**Antecedent (y_i)**
A	ϵ
A fire	ϵ
…	…
a Bangladeshi garment factory	ϵ
…	…
the four-story building	a Bangladeshi garment factory
…	…
out	ϵ

Example Clustering

Input document
A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building. Witnesses … d floor, and that it was locked when the fire broke out.

Not a mention

Span	Antecedent (y_i)
A	ϵ
A fire	ϵ
…	…
a Bangladeshi garment factory	ϵ
…	…
the four-story building	a Bangladeshi garment factory
…	…
out	ϵ

Example Clustering

Input document
A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building. Witnesses say the only exit door was on the ground floor, and that it was locked when the fire broke out.

No link with previously occurring span

Span	**Antecedent (y_i)**
…	…
a Bangladeshi garment factory	ϵ
…	…
the four-story building	a Bangladeshi garment factory
…	…
out	ϵ

Example Clustering

Input document
A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building. Witnesses say the only exit door was on the ground floor, and that it was locked when the fire broke out.

Span	Antecedent (y_i)
A	ϵ
A fire	ϵ
	$\dots$
	ϵ
$\dots$	$\dots$
the four-story building	a Bangladeshi garment factory
$\dots$	$\dots$
out	ϵ

Predicted coreference link

Span Ranking Model

$$P(y_1, \ldots, y_M \mid D) = \prod_{i=1}^{M} P(y_i \mid D)$$

$$= \prod_{i=1}^{M} \frac{e^{s(i, y_i)}}{\sum_{y' \in \mathcal{Y}(i)} e^{s(i, y')}}$$

Factor coreference score $s(i, j)$ to enable span pruning:

$$s(i,j) = \begin{cases} s_{\mathrm{m}}(i) + s_{\mathrm{m}}(j) + s_{\mathrm{a}}(i,j) & j \neq \epsilon \\ 0 & j = \epsilon \end{cases}$$

Span Ranking Model

$$P(y_1, \ldots, y_M \mid D) = \prod_{i=1}^{M} P(y_i \mid D)$$

$$= \prod_{i=1}^{M} \frac{e^{s(i,y_i)}}{\sum_{y' \in \mathcal{Y}(i)} e^{s(i,y')}}$$

Is this span a mention?

Factor coreference score $s(i, j)$ to enable span pruning:

$$s(i,j) = \begin{cases} s_{\mathrm{m}}(i) + s_{\mathrm{m}}(j) + s_{\mathrm{a}}(i,j) & j \neq \epsilon \\ 0 & j = \epsilon \end{cases}$$

Span Ranking Model

$$P(y_1, \ldots, y_M \mid D) = \prod_{i=1}^{M} P(y_i \mid D)$$

$$= \prod^{M} \frac{e^{s(i,y_i)}}{\sum}$$

Is span j an antecedent of span i?

Factor coreference score $s(i,j)$ to enable span pruning:

$$s(i,j) = \begin{cases} s_{\mathrm{m}}(i) + s_{\mathrm{m}}(j) + s_{\mathrm{a}}(i,j) & j \neq \epsilon \\ 0 & j = \epsilon \end{cases}$$

Span Ranking Model

$$P(y_1, \ldots, y_M \mid D) = \prod_{i=1}^{M} P(y_i \mid D)$$

$$= \prod_{i=1}^{M} \frac{e^{s(i,y_i)}}{\sum_{y' \in \mathcal{Y}(i)} e^{s(i,y')}}$$

Factor coreference score $s(i, j)$ to enable span pruning:

$$s(i,j) = \begin{cases} s_{\mathrm{m}}(i) + s_{\mathrm{m}}(j) + s_{\mathrm{a}}(i,j) & j \neq \epsilon \\ 0 & j = \epsilon \end{cases}$$

Dummy antecedent
has a fixed zero score

Experimental Setup

Dataset: English OntoNotes (CoNLL-2012)

Genres: Telephone conversations, newswire, newsgroups, broadcast conversation, broadcast news, weblogs

Documents: 2802 training, 343 development, 348 test

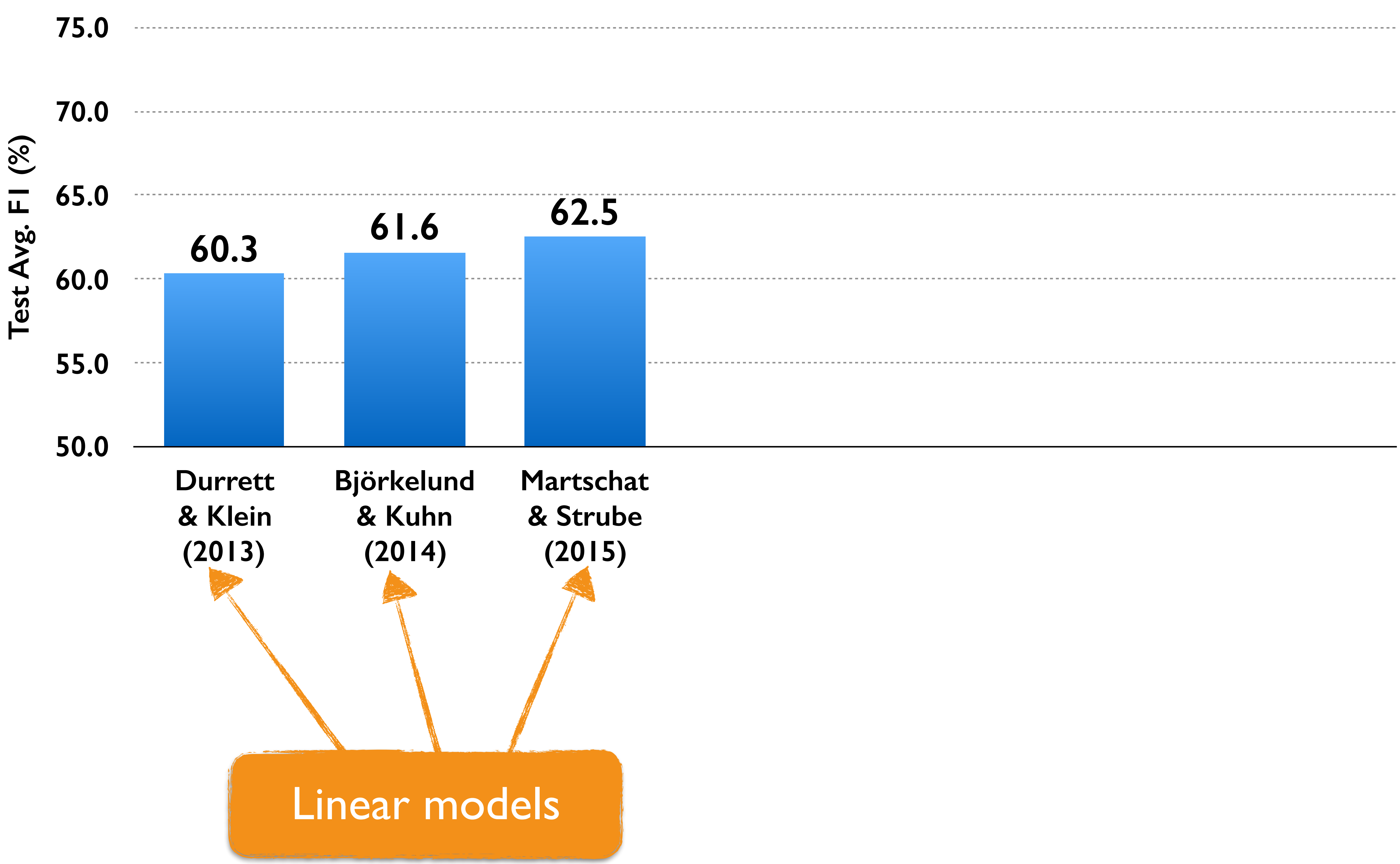
Longest document has 4009 words!

Aggressive pruning: Maximum span width, maximum sentence training, suppress spans with inconsistent bracketing, maximum number of antecedents

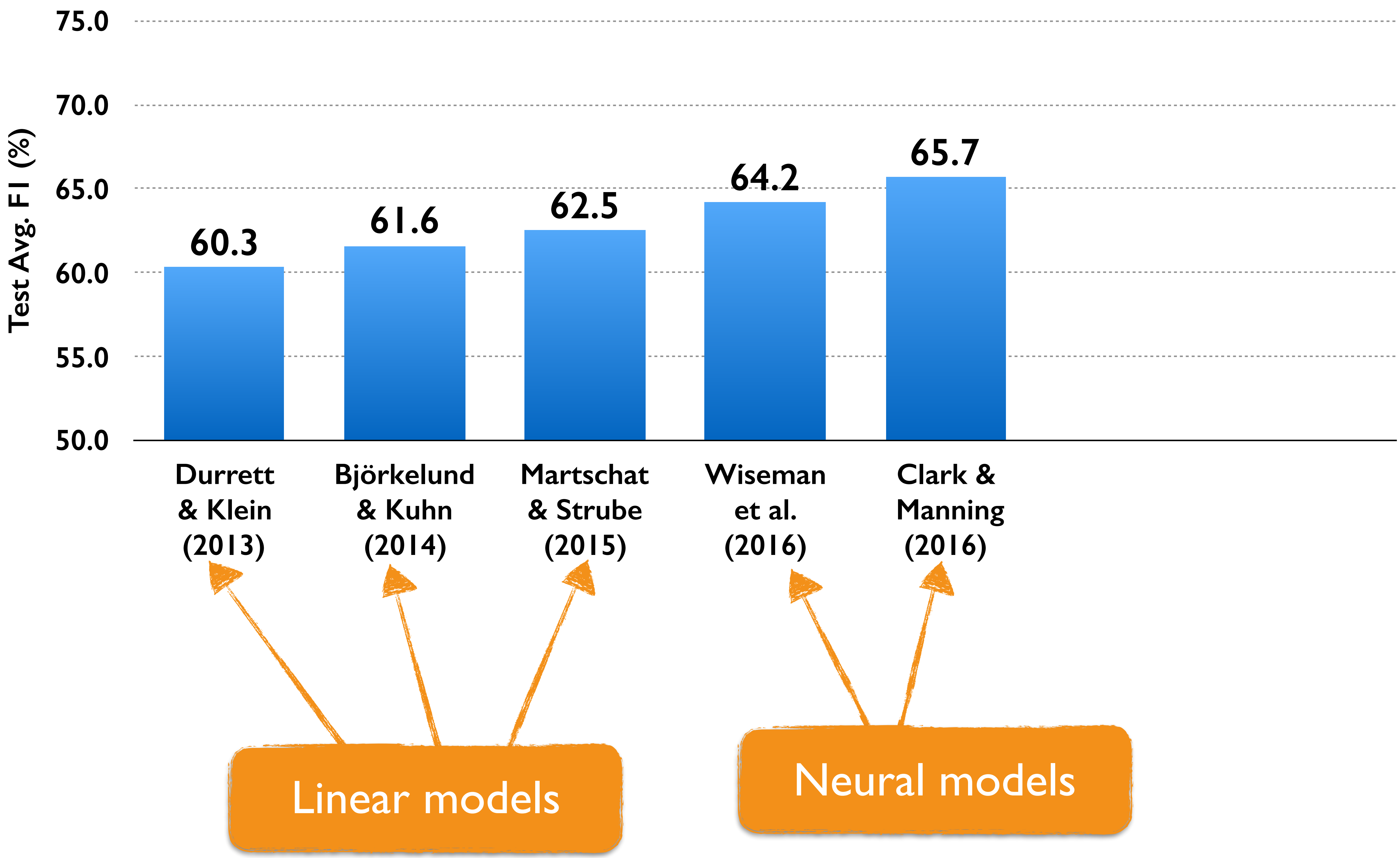
Features: distance between spans, span width

Metadata: speaker information, genre

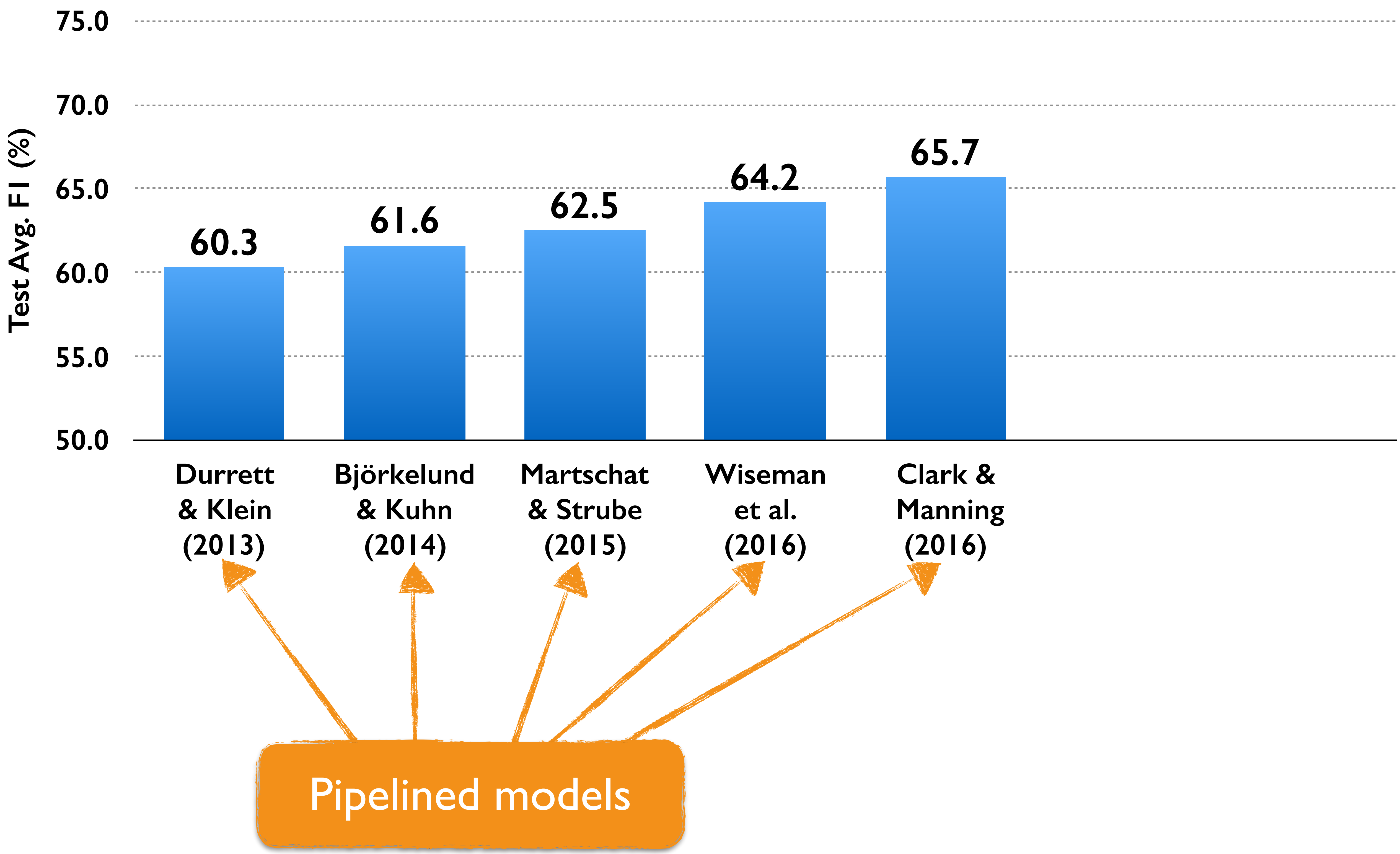
Coreference Results



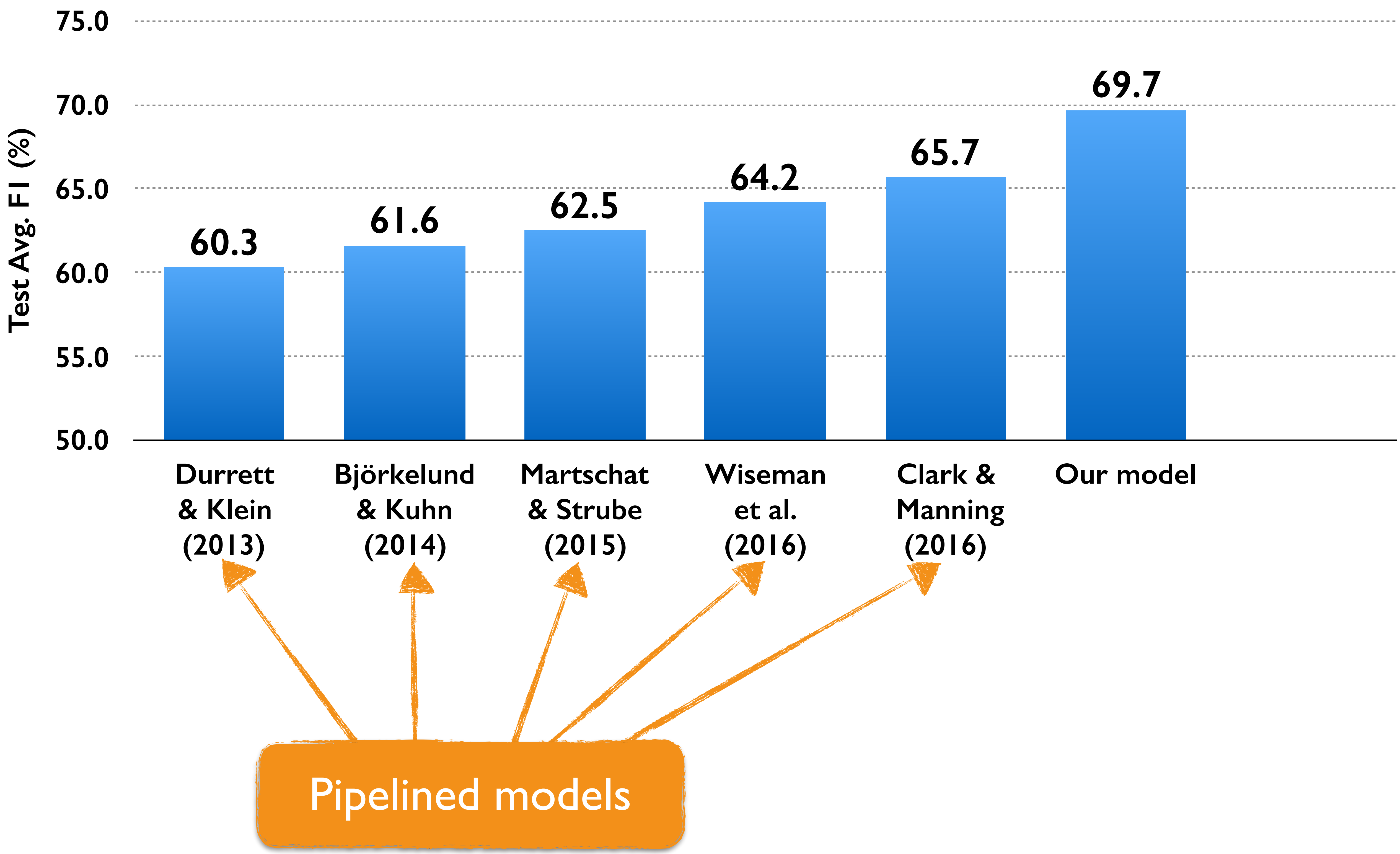
Coreference Results



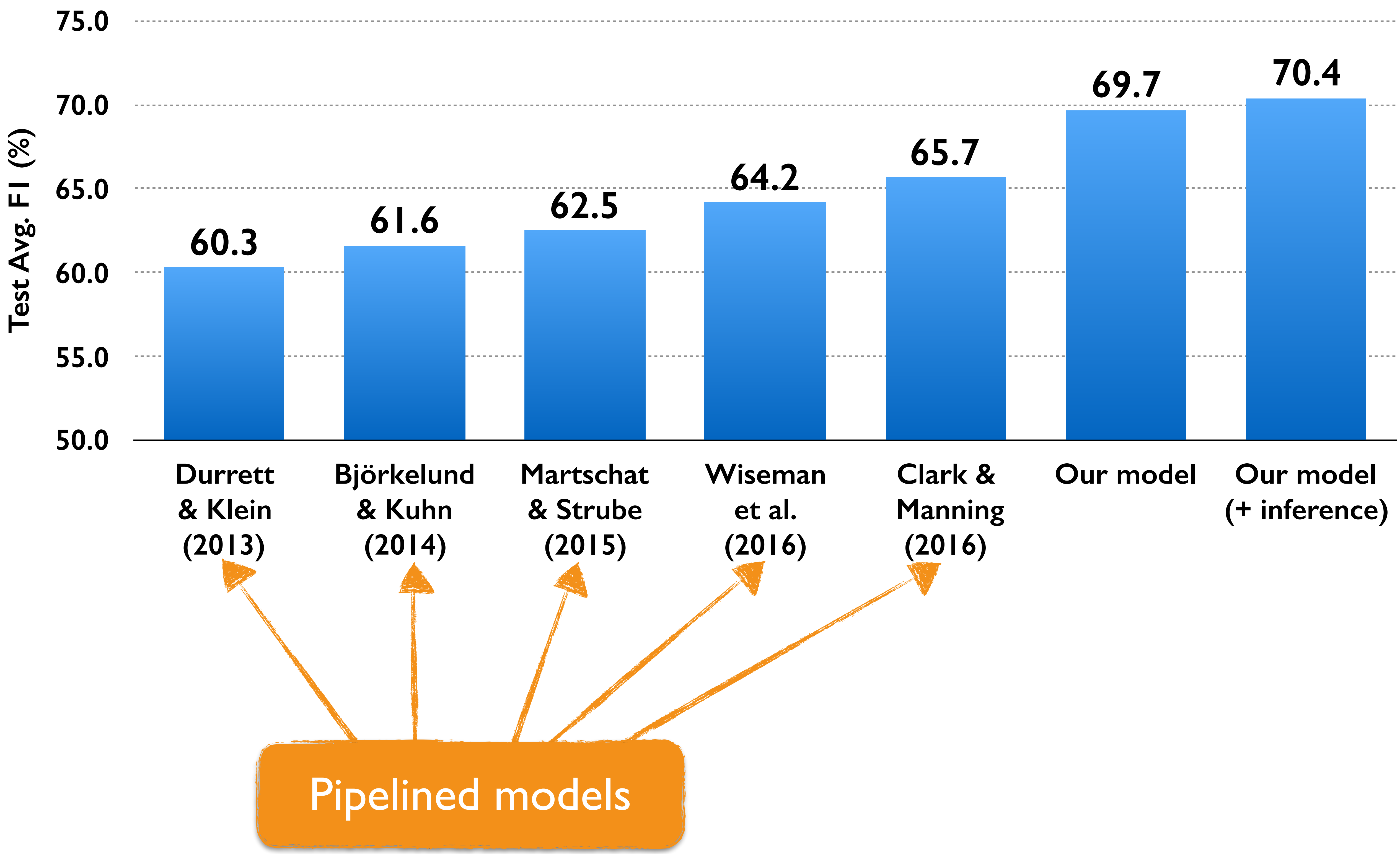
Coreference Results



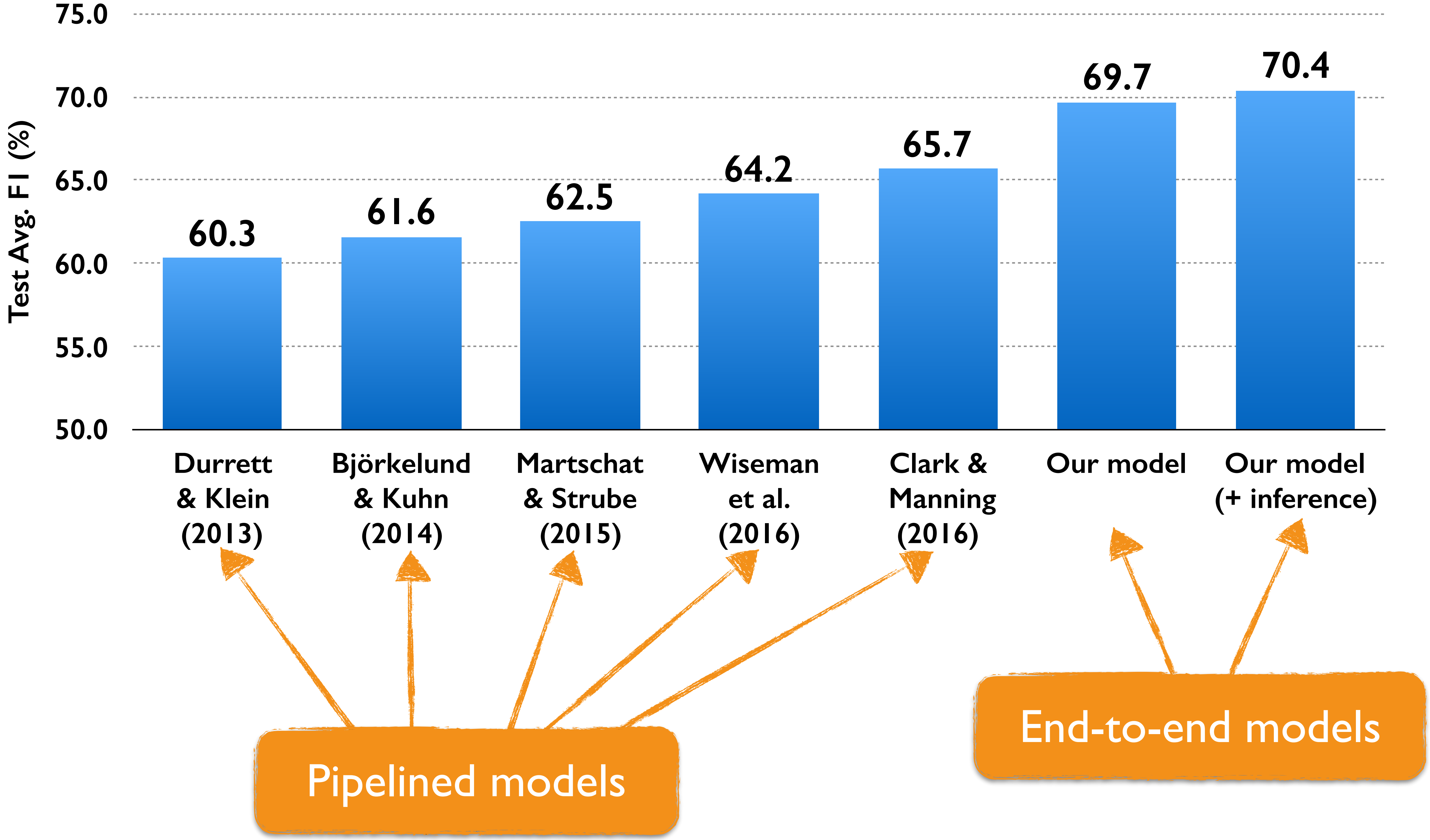
Coreference Results



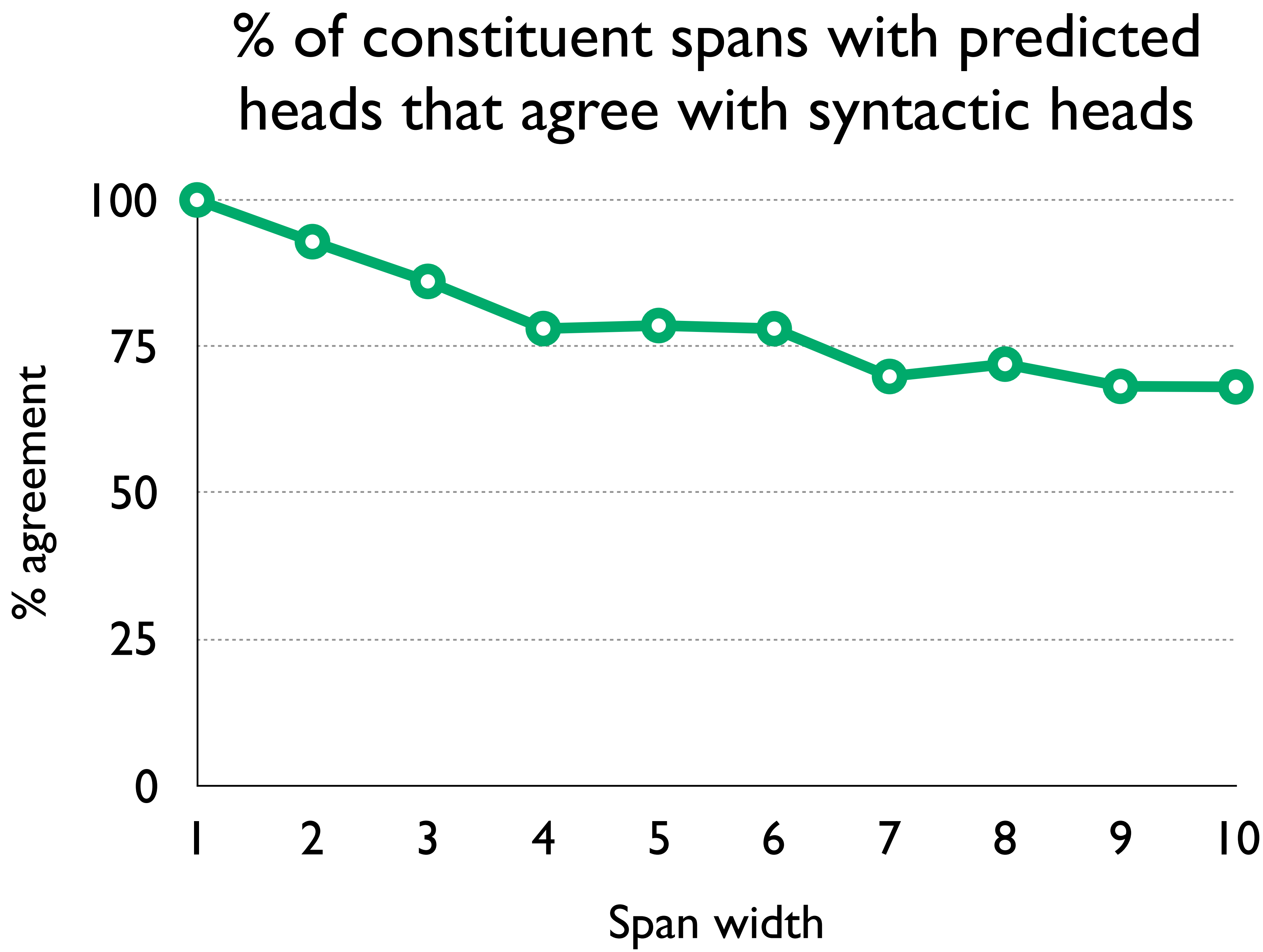
Coreference Results



Coreference Results



Head-finding Agreement



Qualitative Analysis

: Mention in a predicted cluster

: Head-finding attention weight

A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building.

Qualitative Analysis

: Mention in

: Head-findin

Attention-based head finder facilitates soft similarity cues

A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building.

Qualitative Analysis

: Mention in a predicted cluster

: Hea

Good head-finding requires word-order information!

A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building.

Common Error Case

: Mention in a predicted cluster

: Head-finding attention weight

The flight attendants have until 6:00 today to ratify labor concessions. The pilots' union and ground crew did so yesterday.

Common Error Case

: Mention in a predicted cluster

: Head-finding attention weight

The flight attendants have until 6:00 today to ratify labor concessions. The pilots' union and ground crew did so yesterday.

Conflating **relatedness** with **paraphrasing**

New Learning Approaches

New state-of-the-art results for two tasks:

Coreference:

> A fire in a Bangladeshi garment factory has left at least 37 people dead and 100 hospitalized. Most of the deceased were killed in the crush as workers tried to flee the blaze in the four-story building.

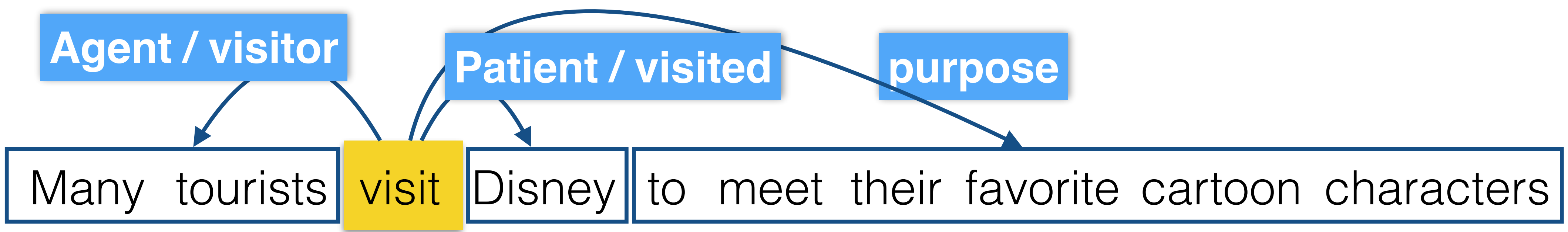
Semantic Role Labeling:

ARG0	NASA
PRED	*observe*
ARG1	an X-ray flare 400 times brighter than usual
TMP	On January 5, 2015

Common themes:

- End-to-end training of deep neural networks
- No preprocessing (e.g., no POS, no parser, etc.)
- Large gains in accuracy with simpler models and no extra training data

Semantic Role Labeling (SRL)



Predicate | **Arguments**

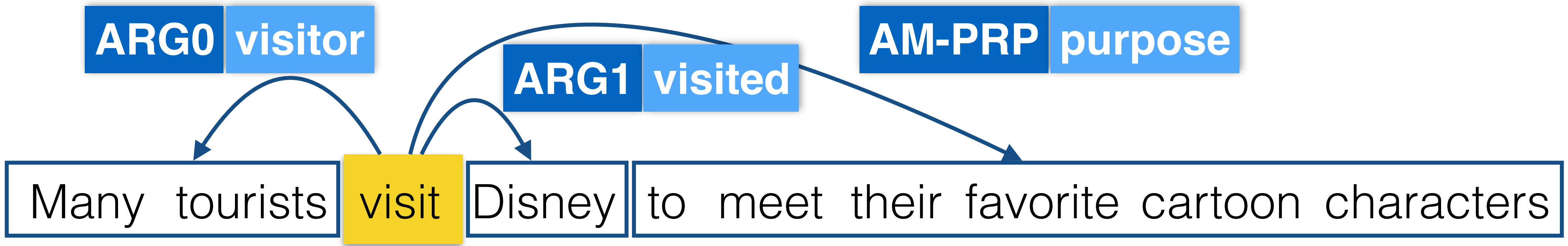
visit

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]

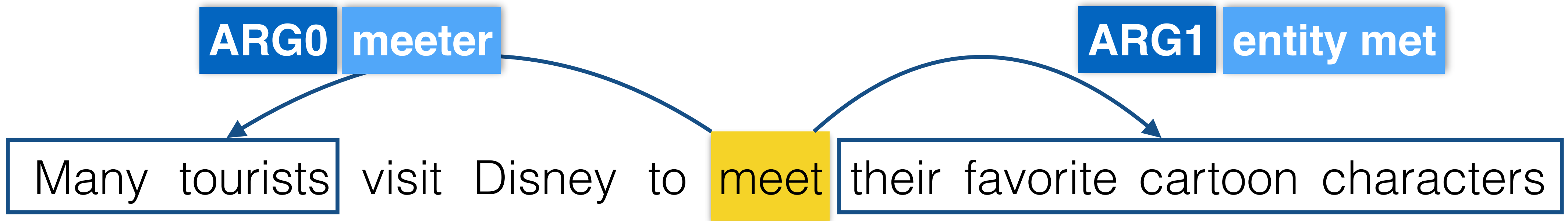
Frame: *visit.01*

role	description
ARG0	visitor
ARG1	visited

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

Semantic Role Labeling (SRL)



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 SRL Tasks: give gold **predicates**, predict the **argument spans** and **labels**.

Challenges of SRL

Over 10 years, F1 on the PropBank test set: 79.4 (Punyakanok 2005) — 80.3 (FitzGerald 2015)

Annotation Challenge:

- PropBank has ~100 pages annotation guideline

Modeling Challenges:

- Syntactic Alternation
- Prepositional Phrase (PP) Attachment
- Long-range Dependencies and Common Sense

Syntactic Alternation

PP Attachment

Long-range Dependencies

ARG0 causer | ARG1 thing sinking

The alien invasion | sank | our city

ARG1 thing sinking

Ergative

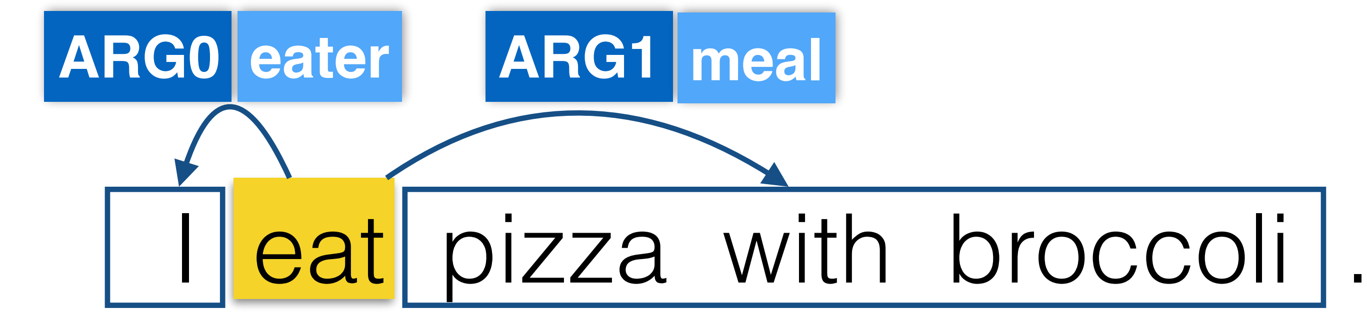
Our city | sank

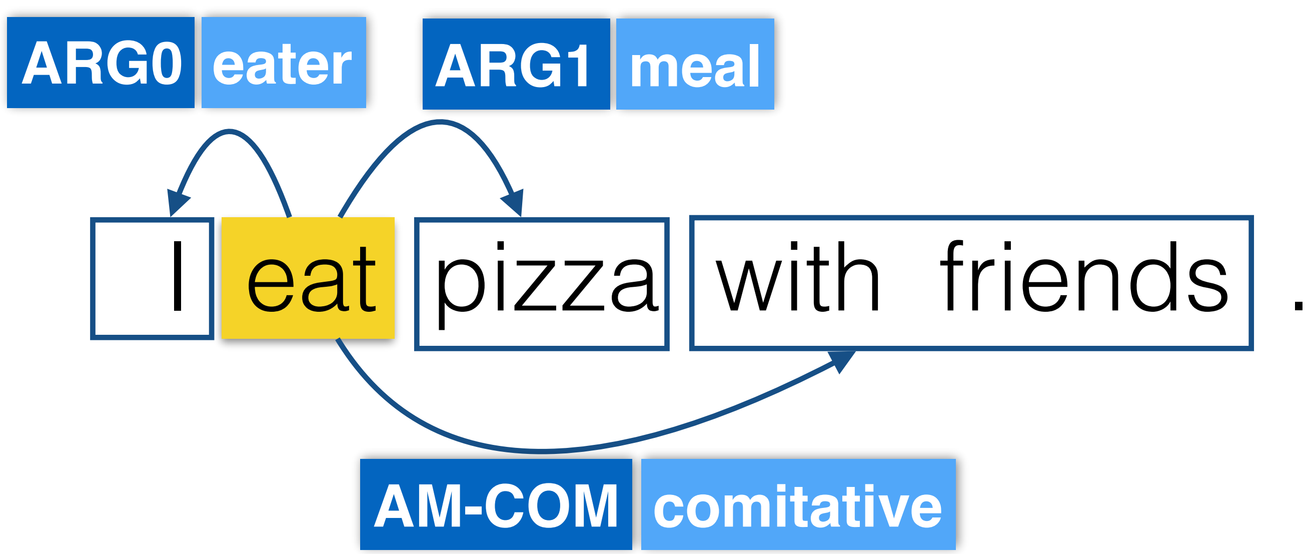
ARG1 thing sinking | ARG0 causer

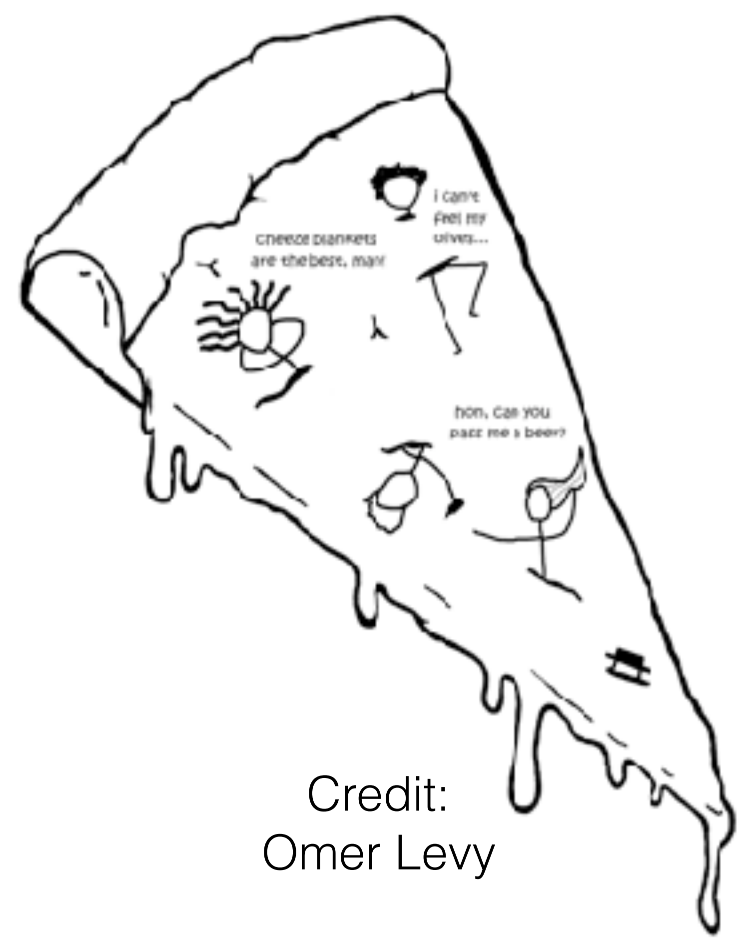
Nominal predicate (Nombank, OntoNotes …)

The | sinking | of | our city | by | alien invasions

Prepositional Phrase (PP) Attachment





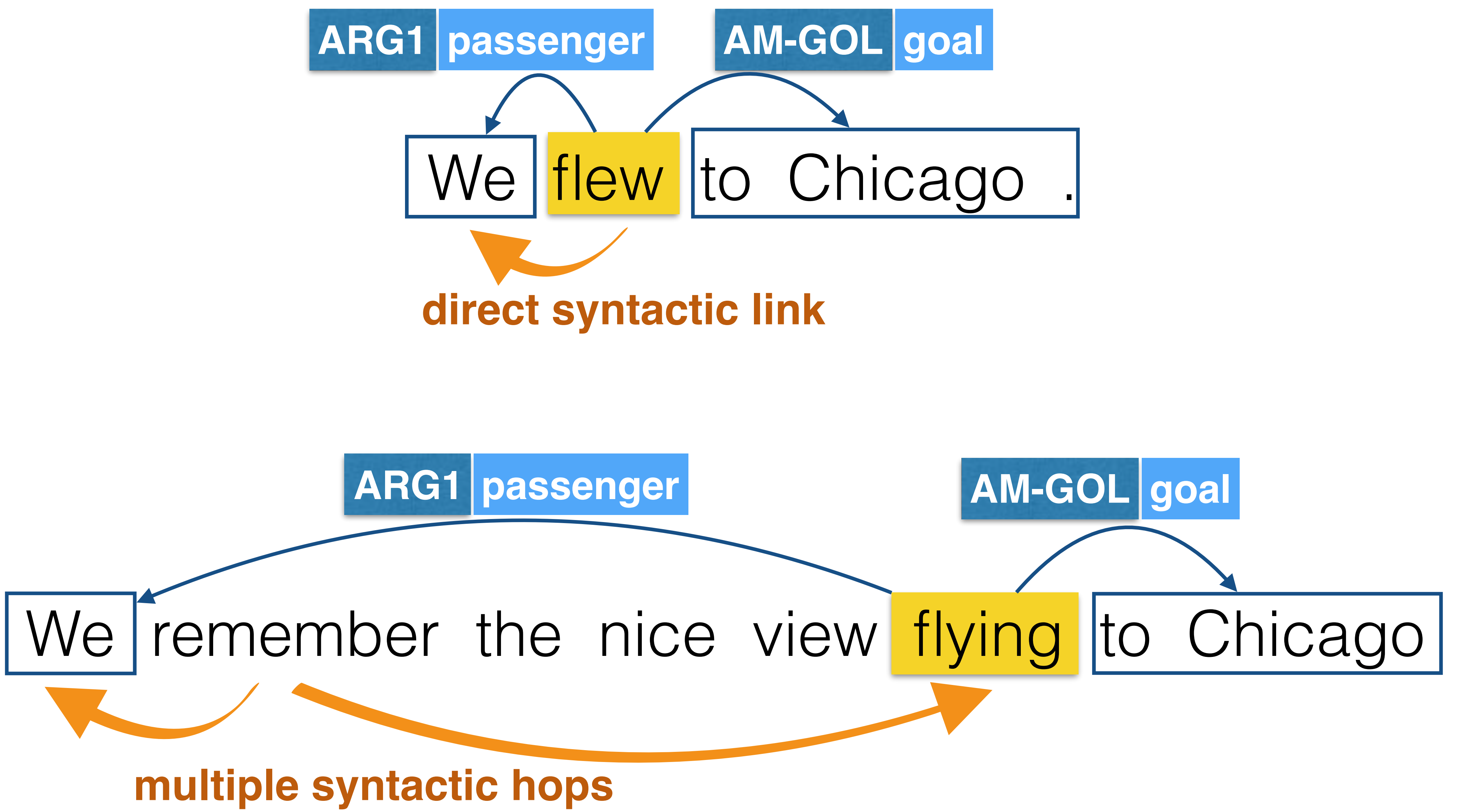


Credit:
Omer Levy

Syntactic Alternation

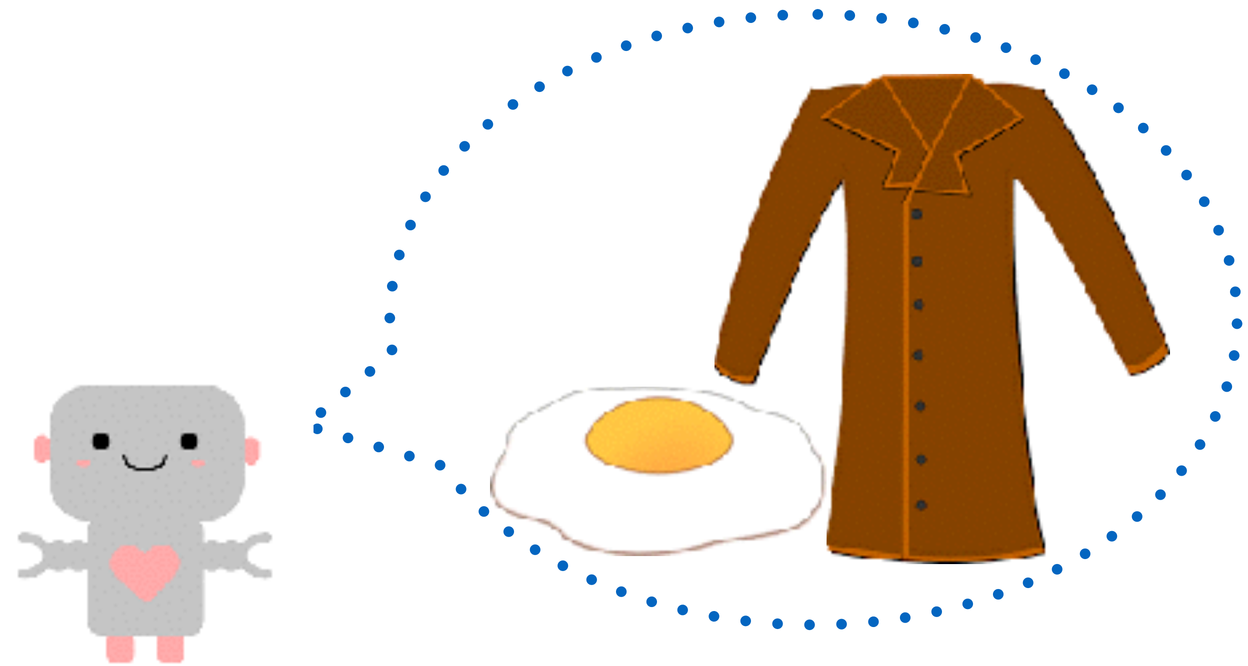
PP Attachment

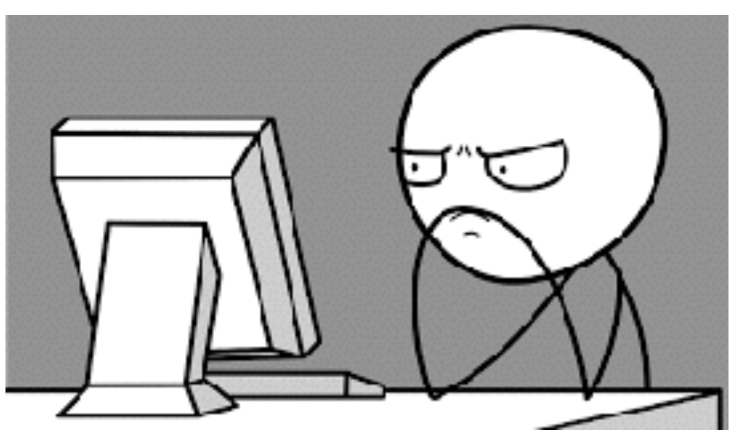
Long-range Dependencies



With out-of-domain data, all these difficulties will be further amplified …

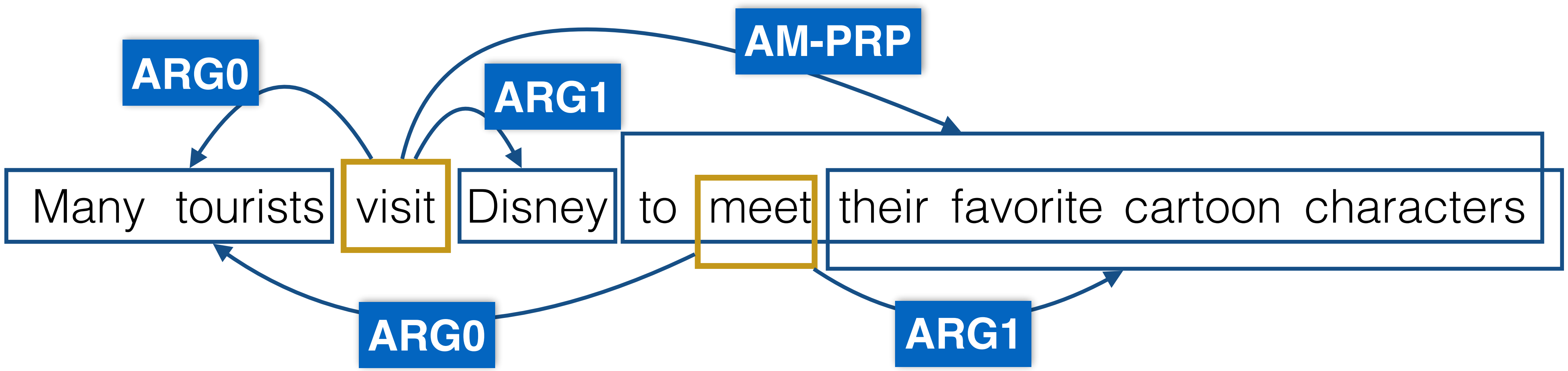
“Dip chicken breasts into eggs to ***coat***”





Active, Ser133-phosphorylated CREB ***effects*** transcription of CRE-dependent genes via interaction with the 265-kDa …

Intuition: SRL as Span-Span Relations

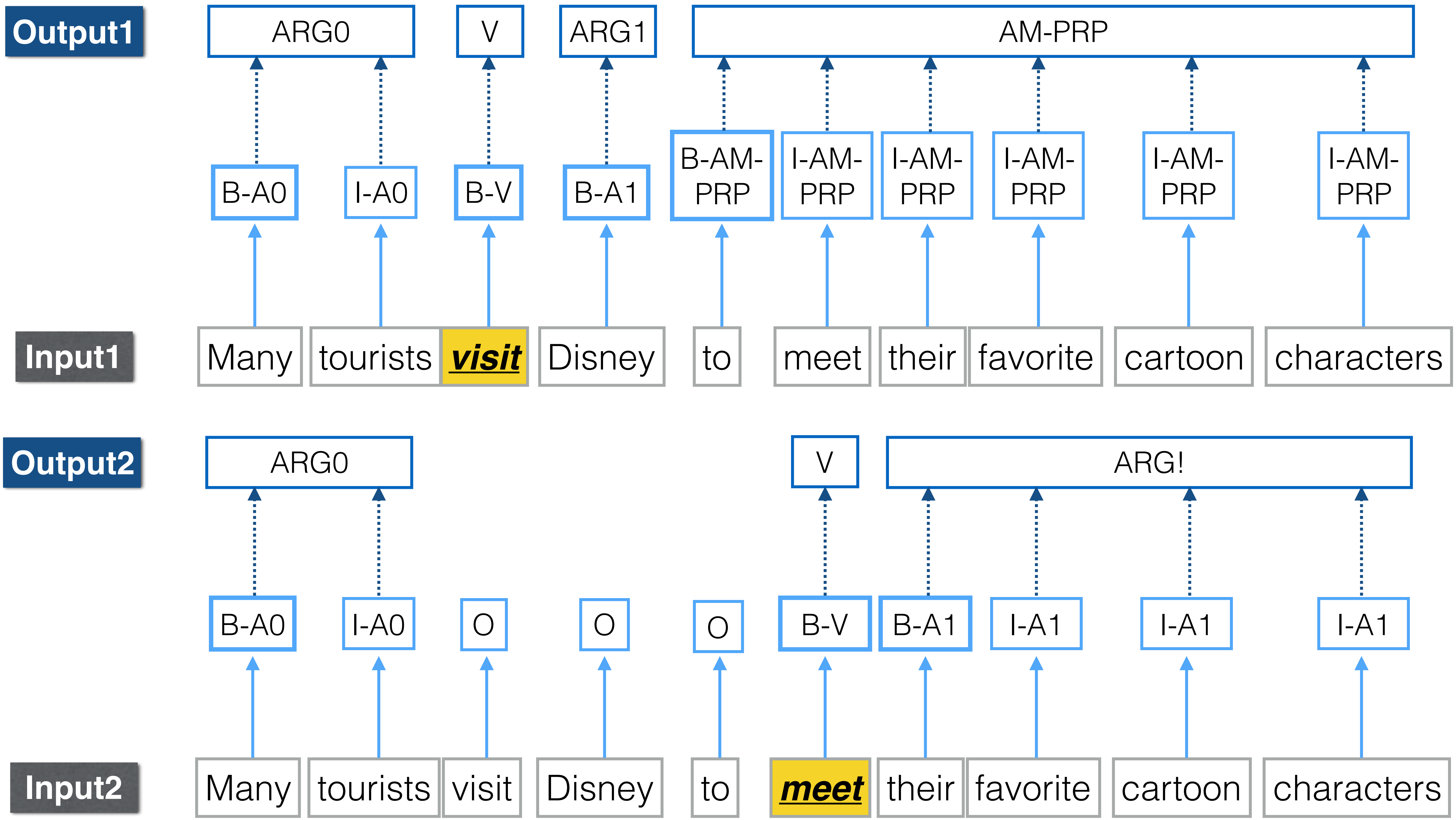


Goal: Predict complete graph for all verbs at once

Challenges:

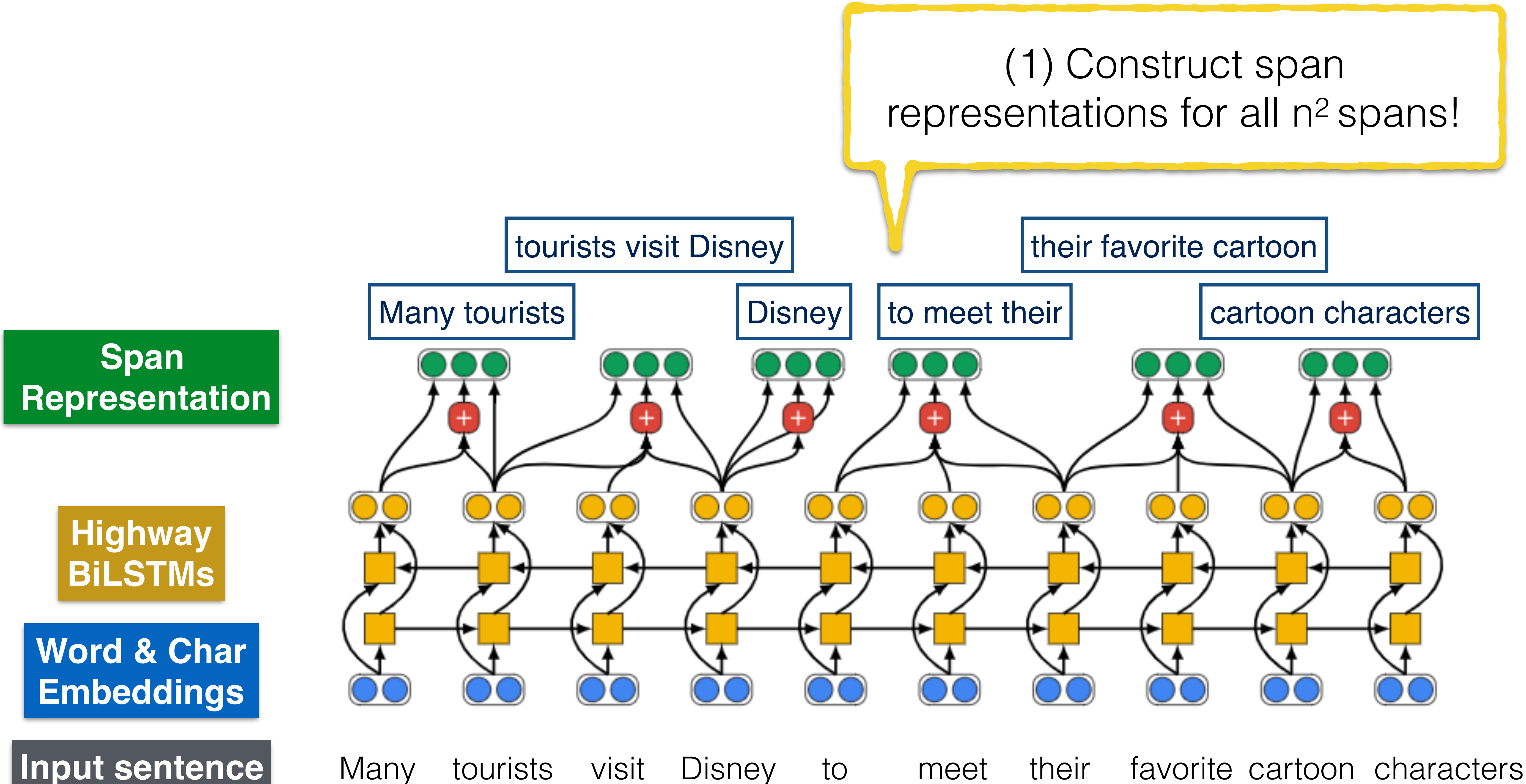
1. Span can nest within each other.
2. Many possible edges (n^2 argument spans & n predicates).

Previous Work: SRL as BIO Tagging



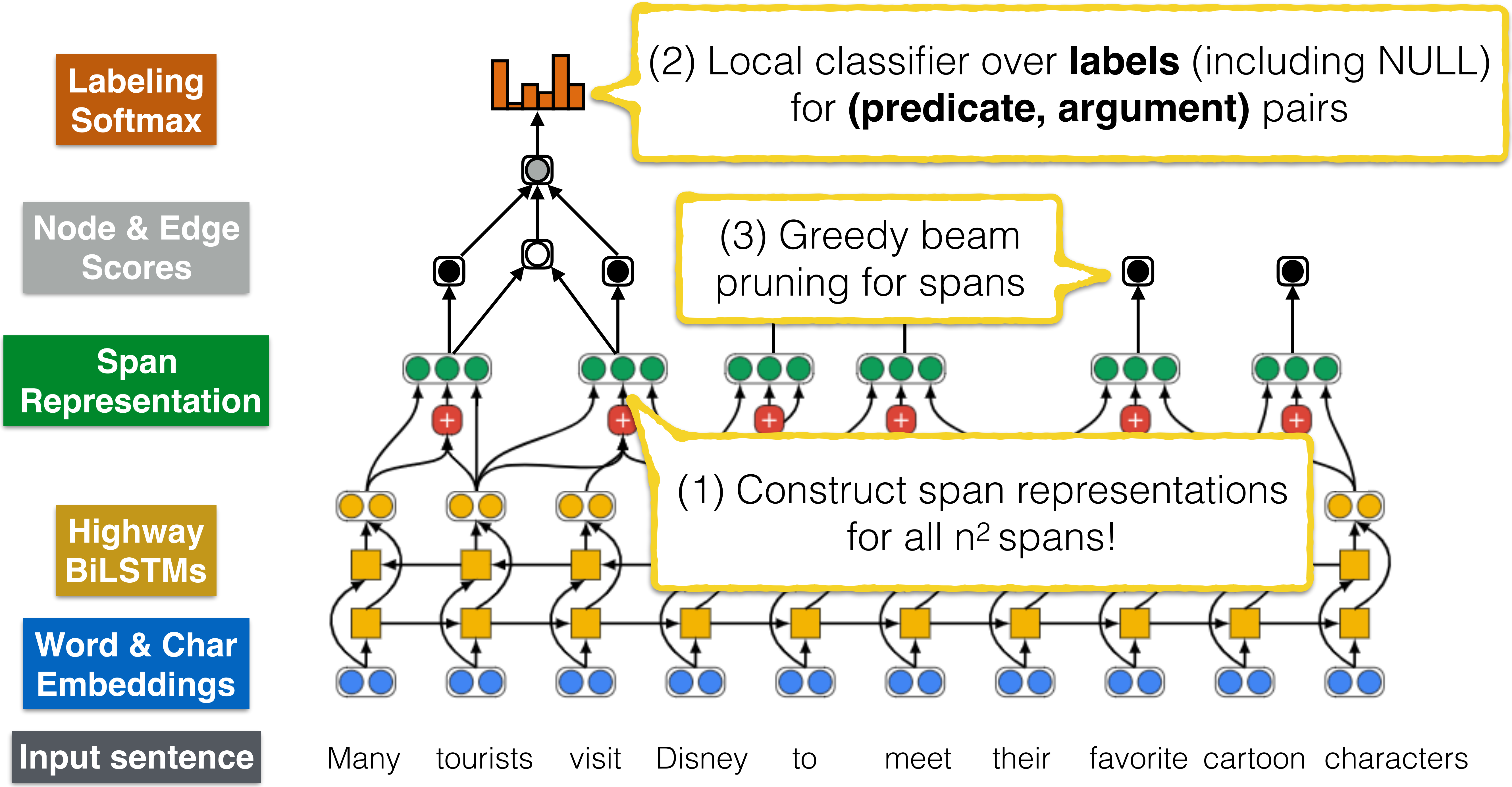
[Zhou and Xu 2015; He et al 2017]

Span Pair Classification



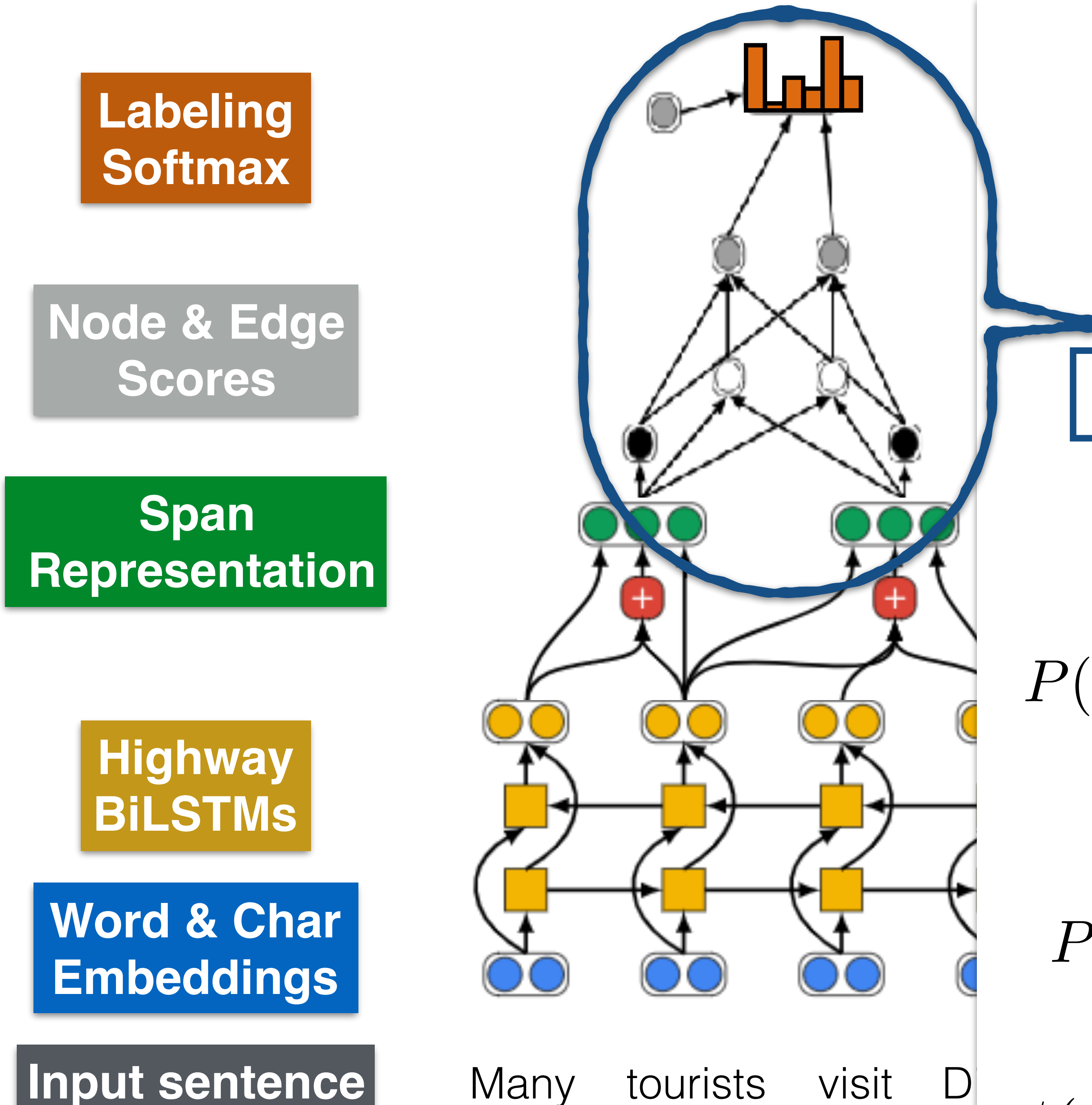
[He et al, in prep]

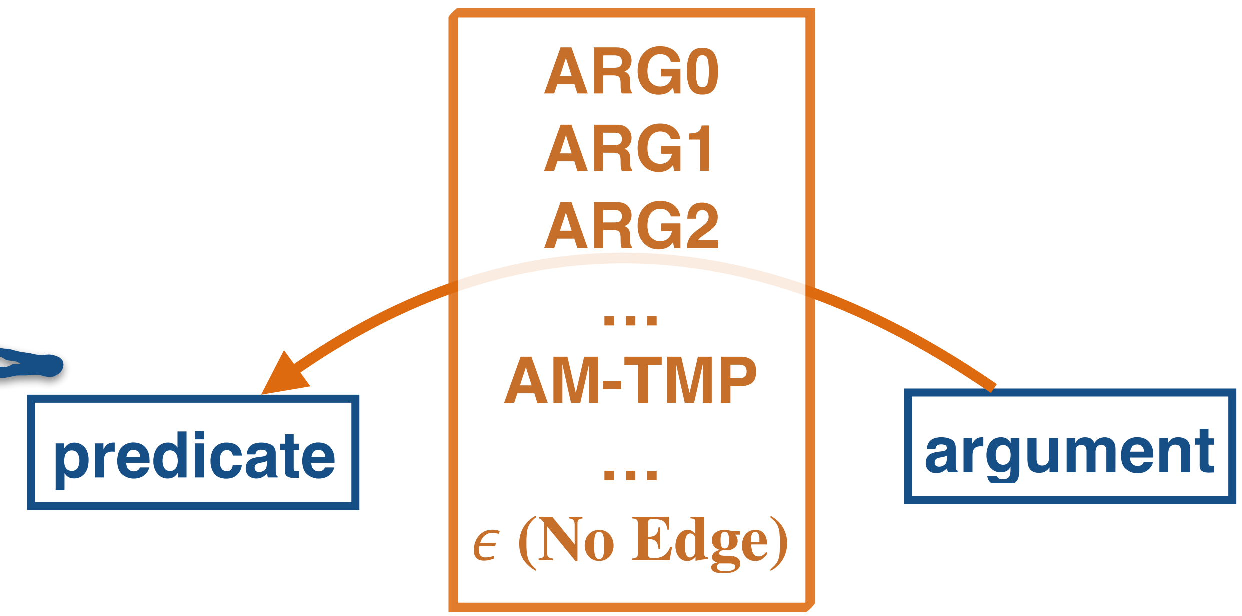
Span Pair Classification



[He et al, in prep]

Local Span Classifier



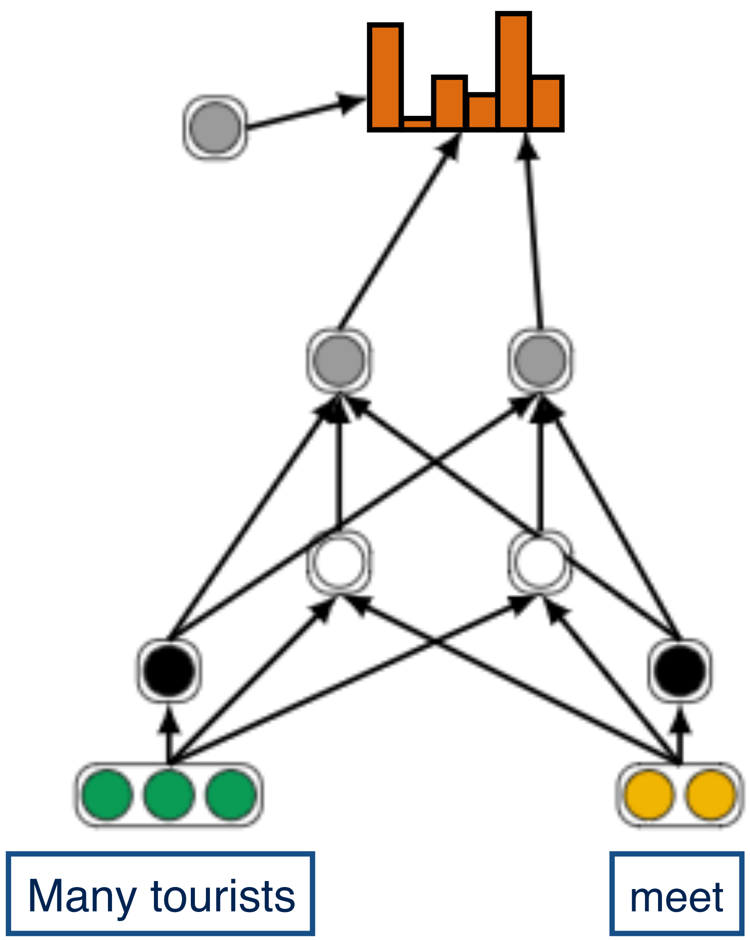


$$P(Y \mid X) = \prod_{\substack{\text{arg}\in\text{Spans} \\ \text{pred}\in\text{Words}}} P(y_{\text{pred,arg}} = l \mid X)$$

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

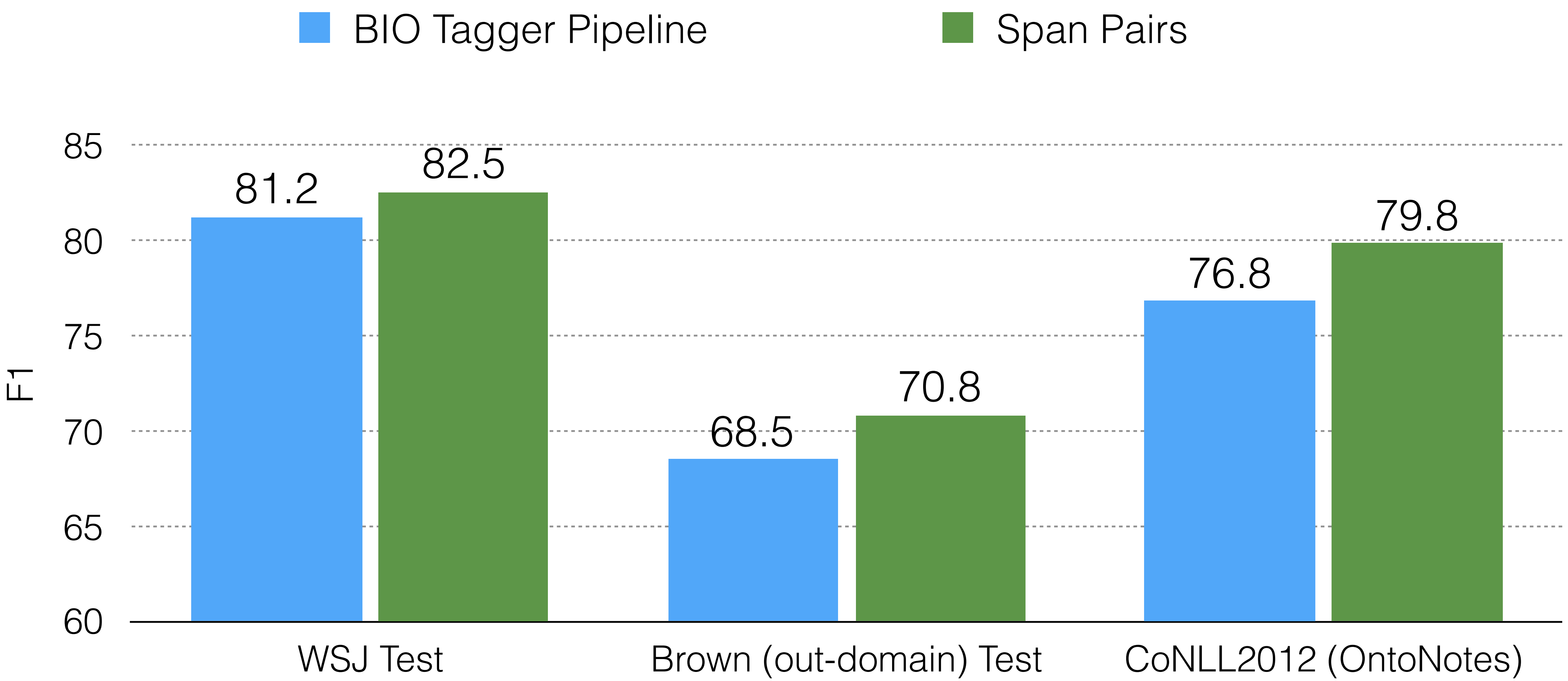
$$\phi(\text{pred}, \text{arg}, l) = \Phi_{\text{c}}(s_{\text{arg}}) + \Phi_{\text{p}}(s_{\text{pred}}) + \Phi_{\text{rel}}^{(l)}(s_{\text{arg}}, s_{\text{pred}})$$

Other Implementation Details …



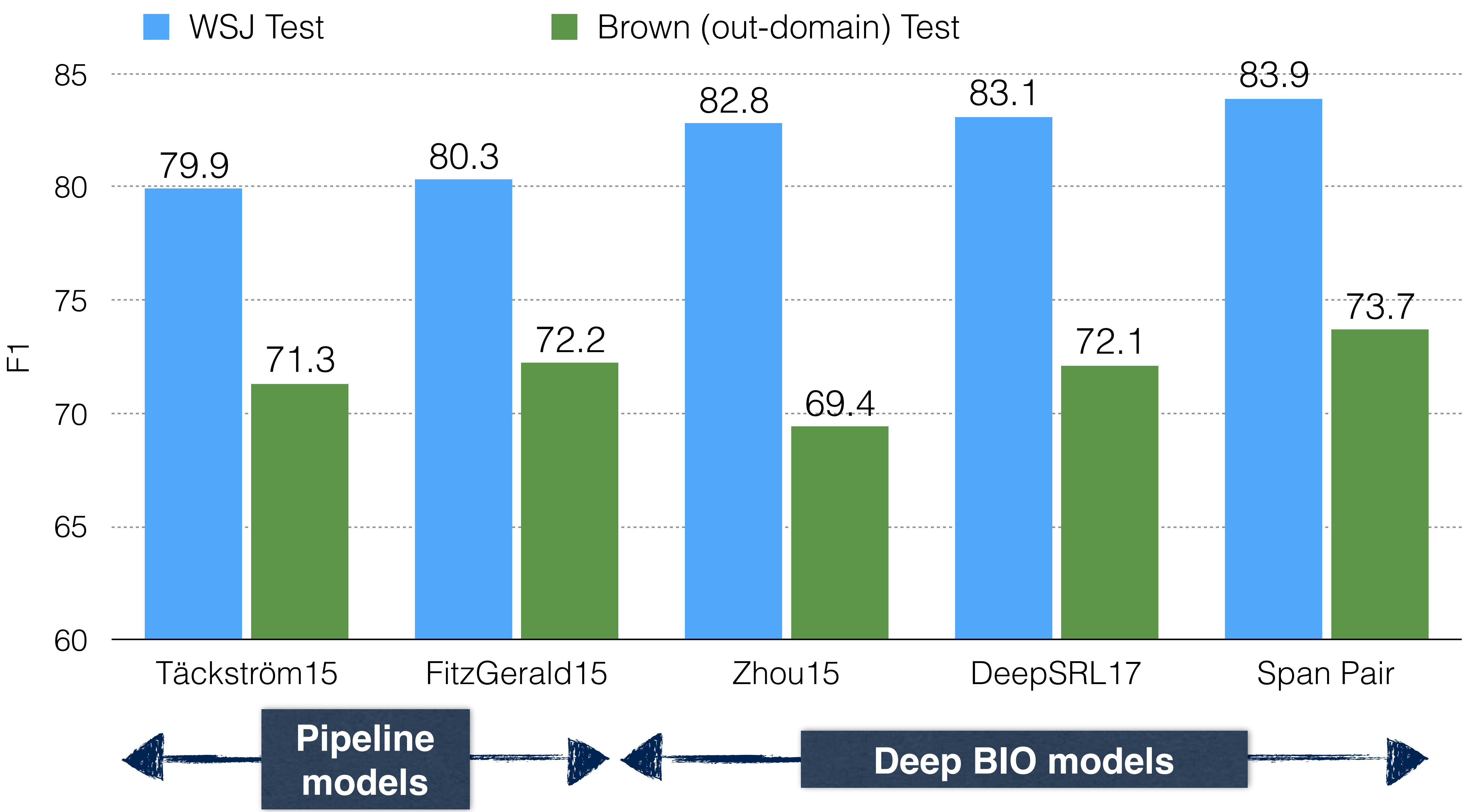
- Predicates are single words.
- Max. length of argument spans = 30.
- Trained for 300 epochs (Dev cycle < 48 hours)
- Reuse all other hyperparameters from coreference model

End-to-End SRL Results

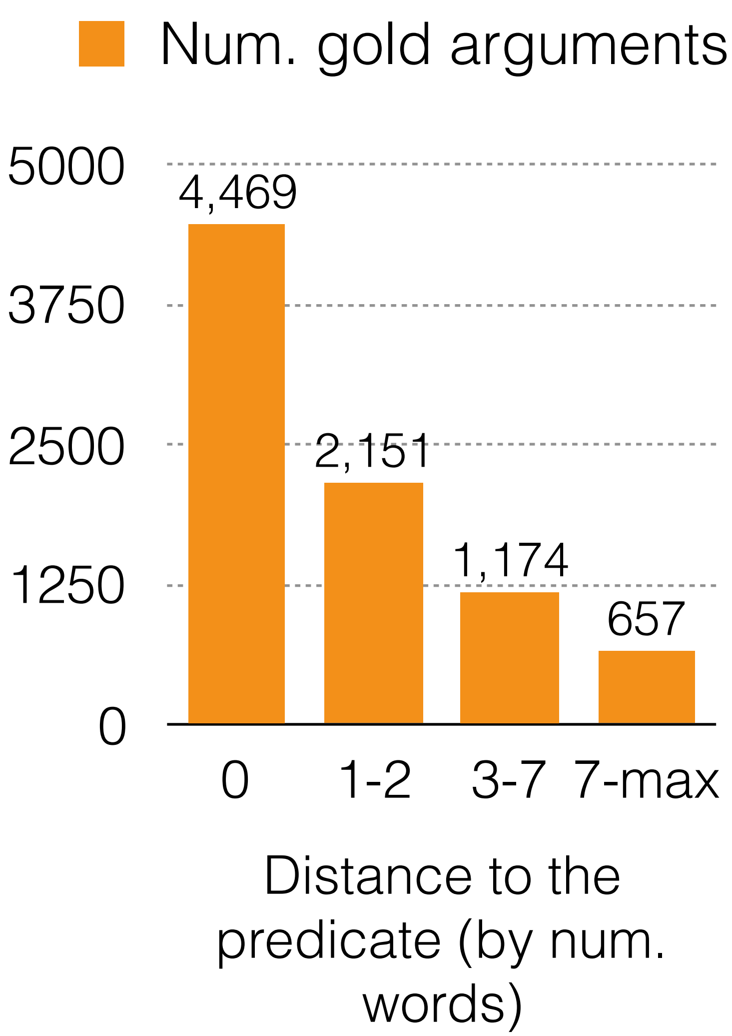


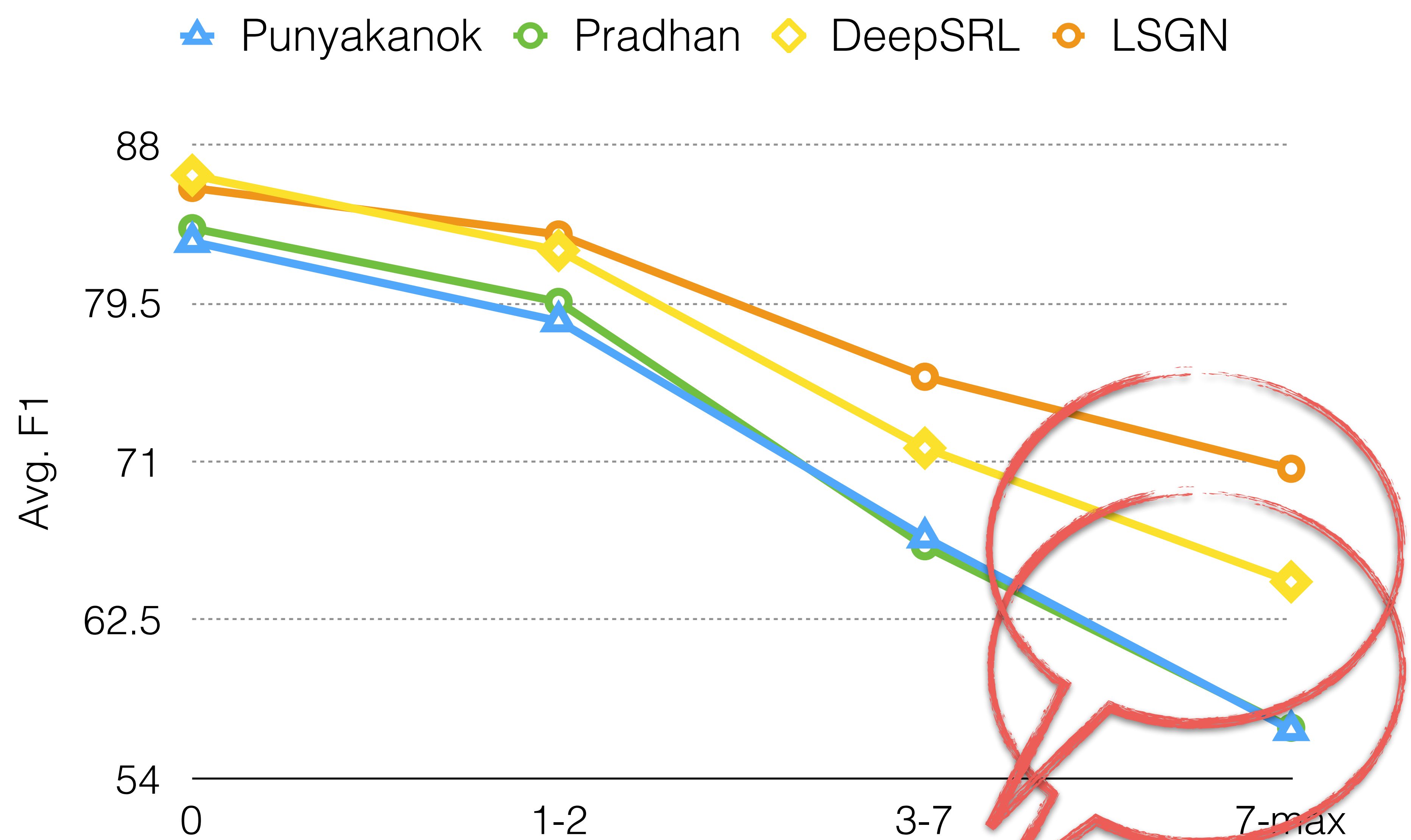
- More improvements on Brown (out-domain) & OntoNotes (with nominal predicates)

Gold Predicates: CoNLL 2005 SRL Results



Quantitative Analysis: Long-range Dependencies





LSGN out-performs **DeepSRL**
on long-range dependencies
with more direct interaction between
predicates and arguments.

Does the Recipe Work for Broad Coverage Semantics?

Step 1: Gather lots of training data!

Challenge 1: Data is costly and limited (e.g. linguists required to label PennTreebank / OntoNotes)

Step 2: Apply Deep Learning!!

Challenge 2: Pipeline of structured prediction problems with cascading errors (e.g. POS->Parsing->SRL->Coref)

Step 3: Observe Impressive Gains!!!

Where Will the Data Come From???

Option 1: Semi-supervised learning

- E.g. word2vec and GloVe are in wide use
  [Mikolov et al., 2013; Pennington et al., 2014]
- Can we learn better word representations?

Option 2: Supervised learning

- Can we gather more direct forms of supervision?

Learning Better Word Representations

Goal: Model contextualized syntax and semantics

$$R(w_i, w_1 \dots w_n) \in \mathbb{R}^n$$

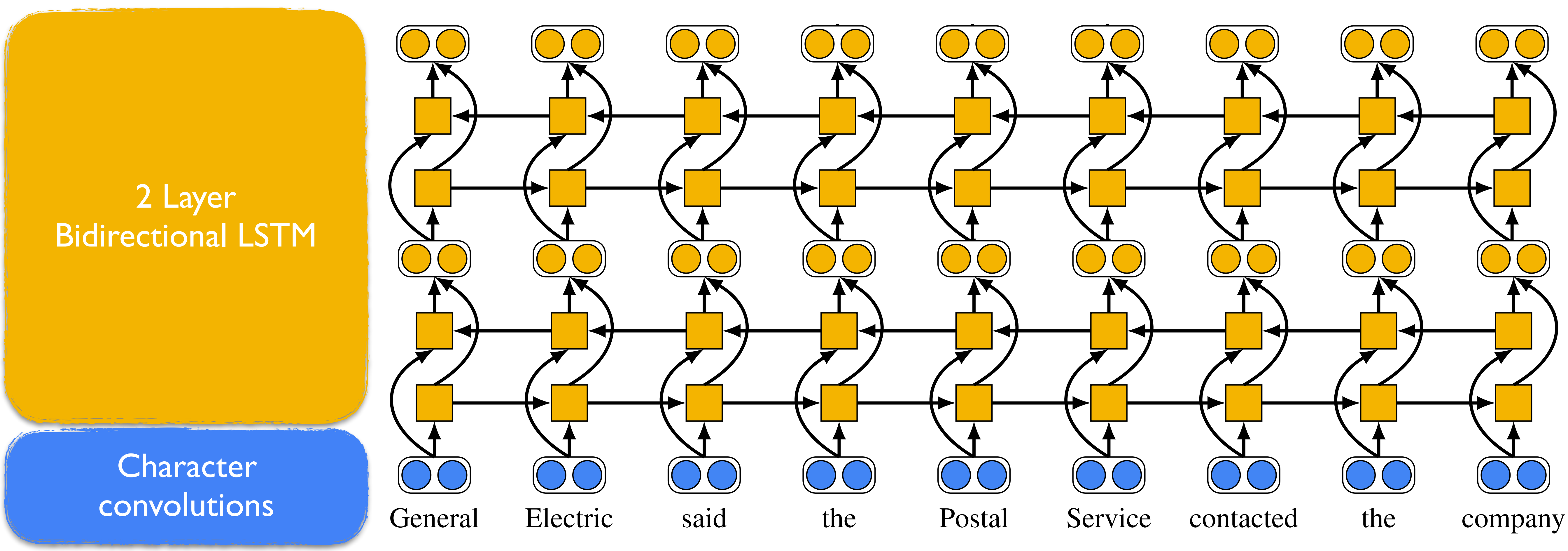
R(plays, “The robot plays piano.”)

$$\neq$$

R(plays, “The robot starred in many plays.”)

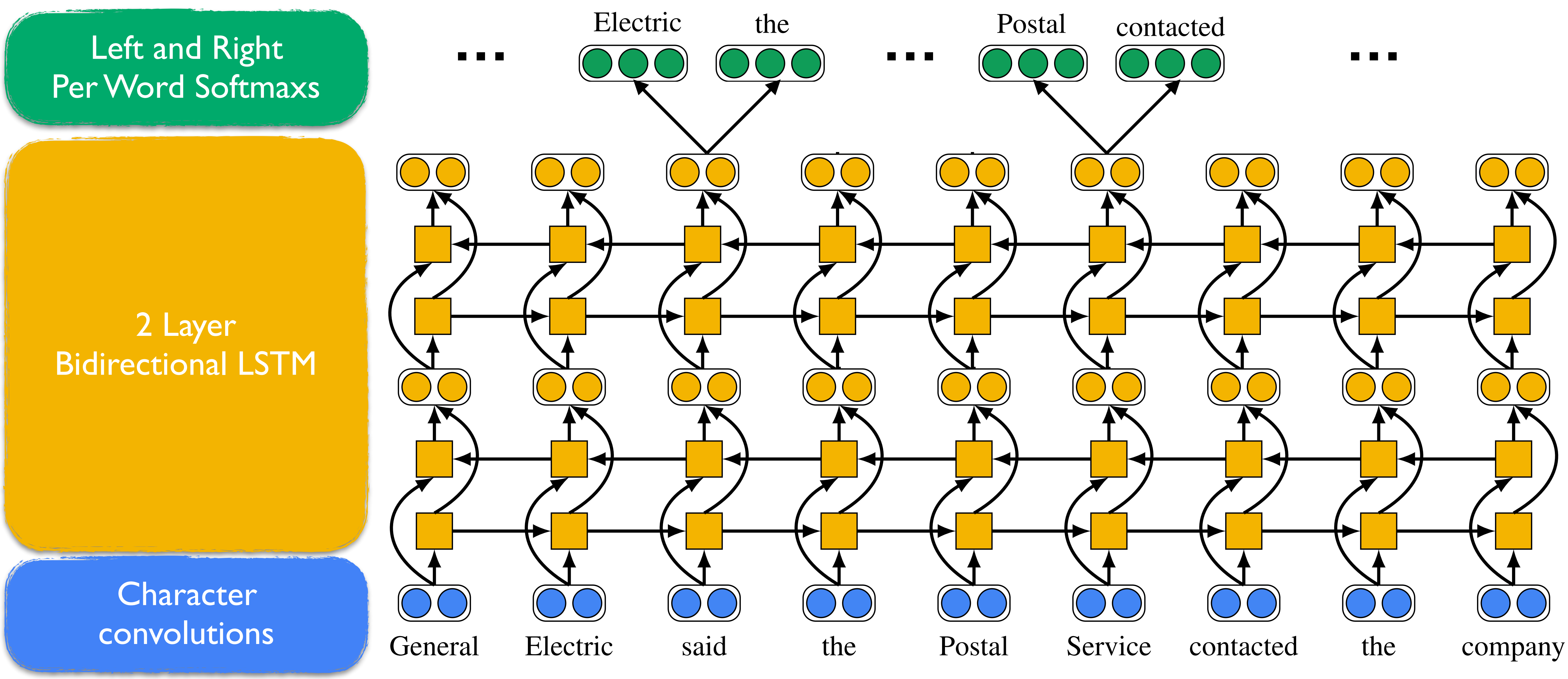
ELMo: Embeddings from a Language Model

Step 1: Train a large BiLM on unlabeled data



ELMo: Embeddings from a Language Model

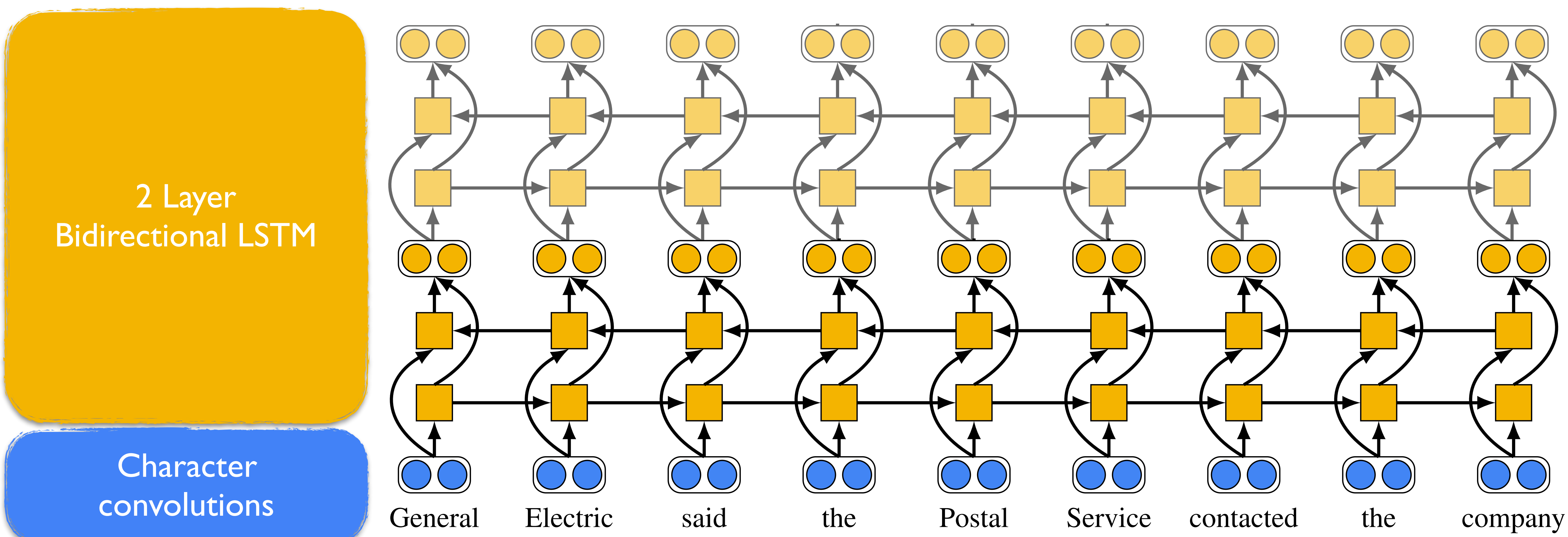
Step 1: Train a large BiLM on unlabeled data



ELMo: Embeddings from a Language Model

Step 1: Train a large BiLM on unlabeled data

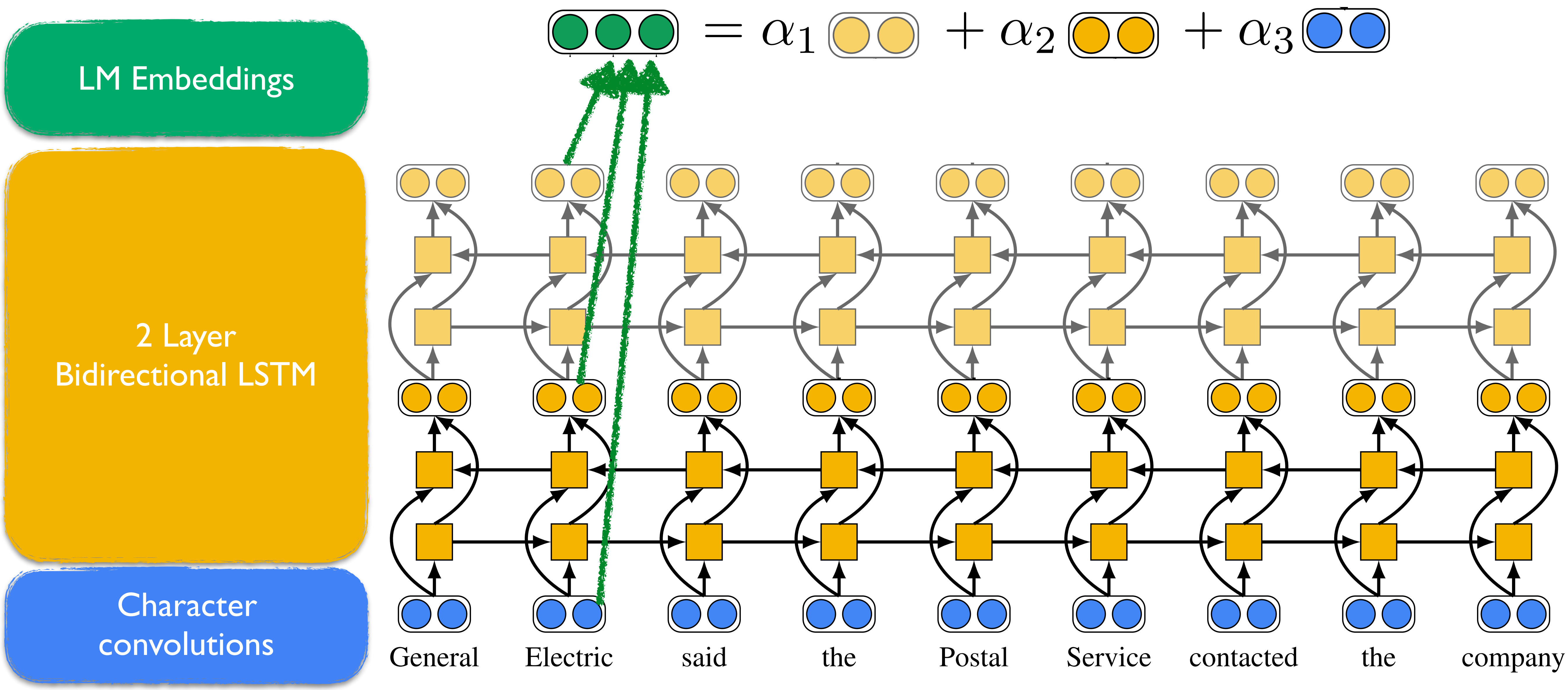
Step 2: Compute linear function of pre-trained model



ELMo: Embeddings from a Language Model

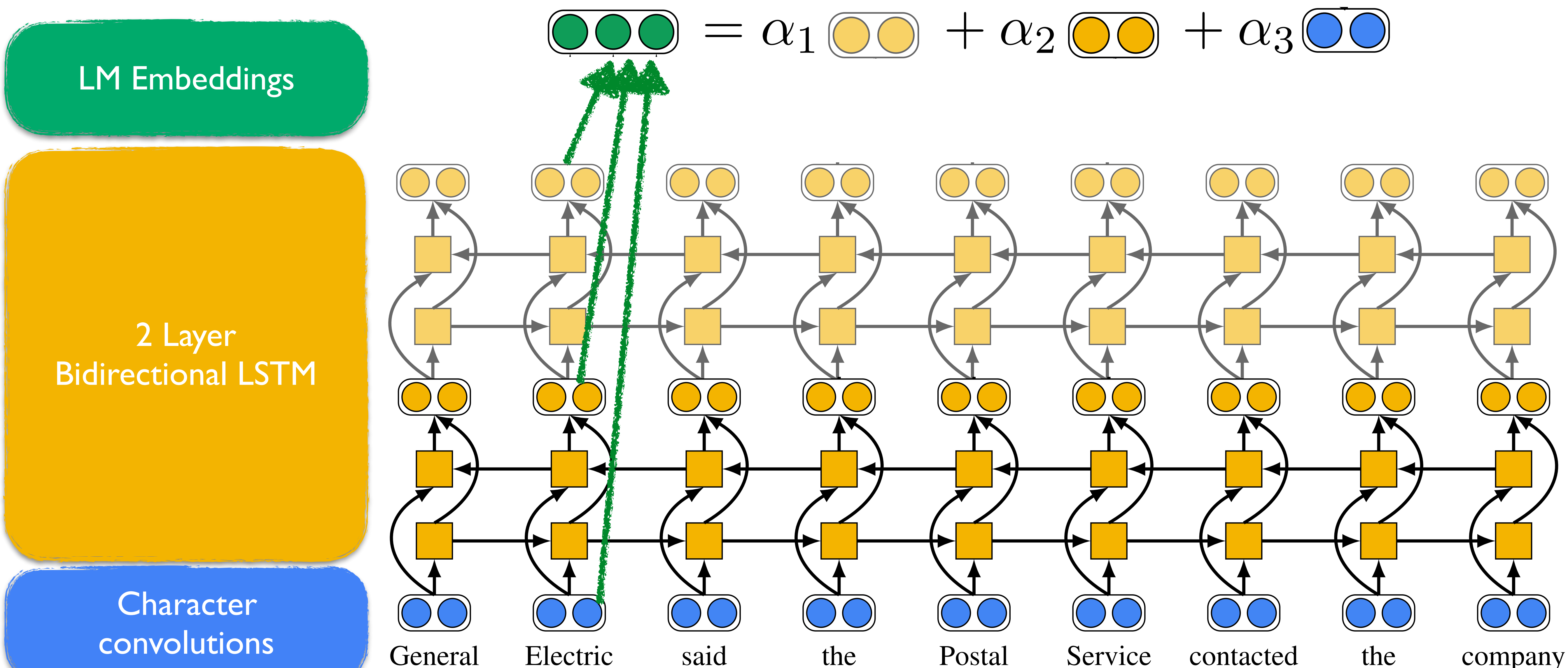
Step 1: Train a large BiLM on unlabeled data

Step 2: Compute linear function of pre-trained model

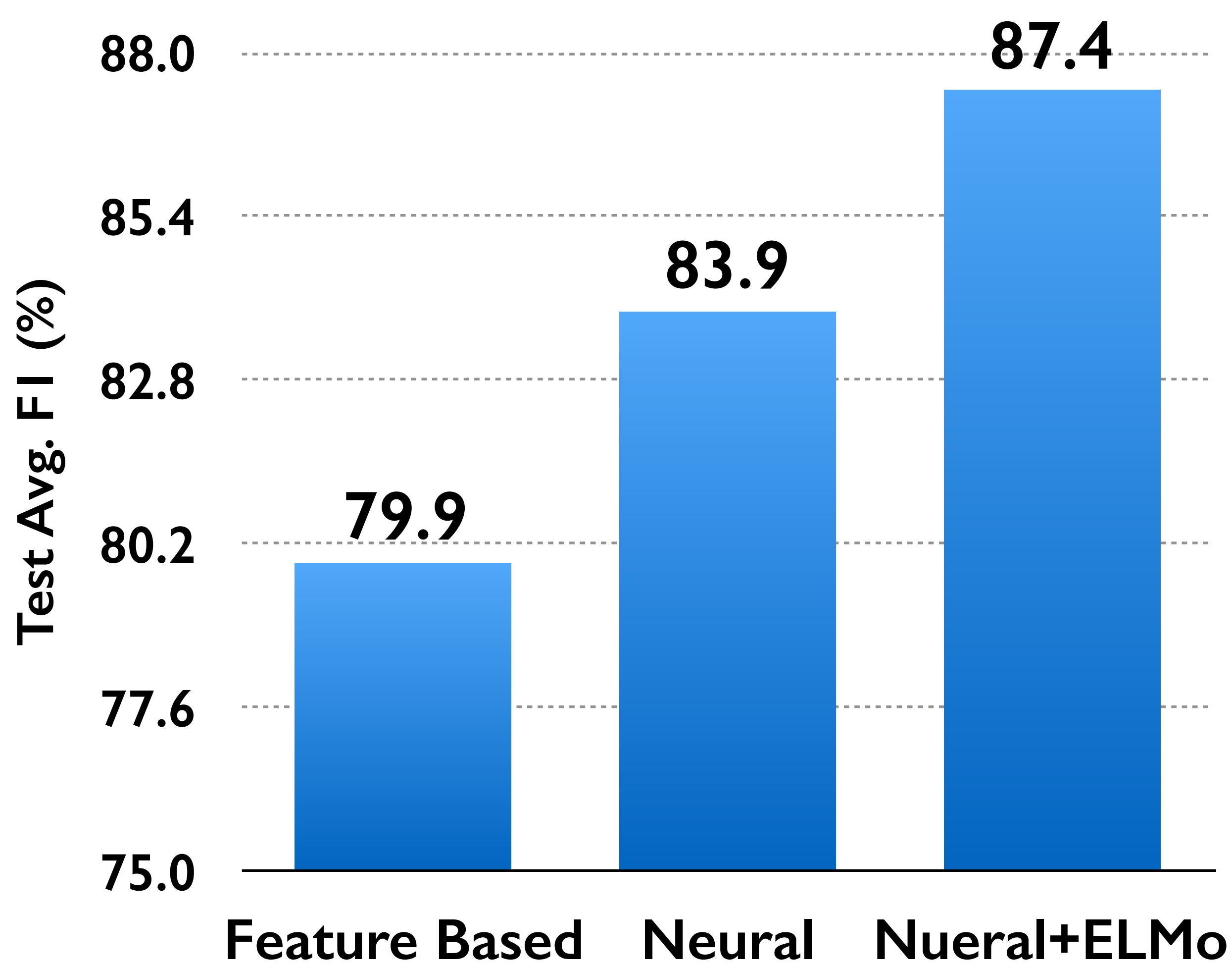


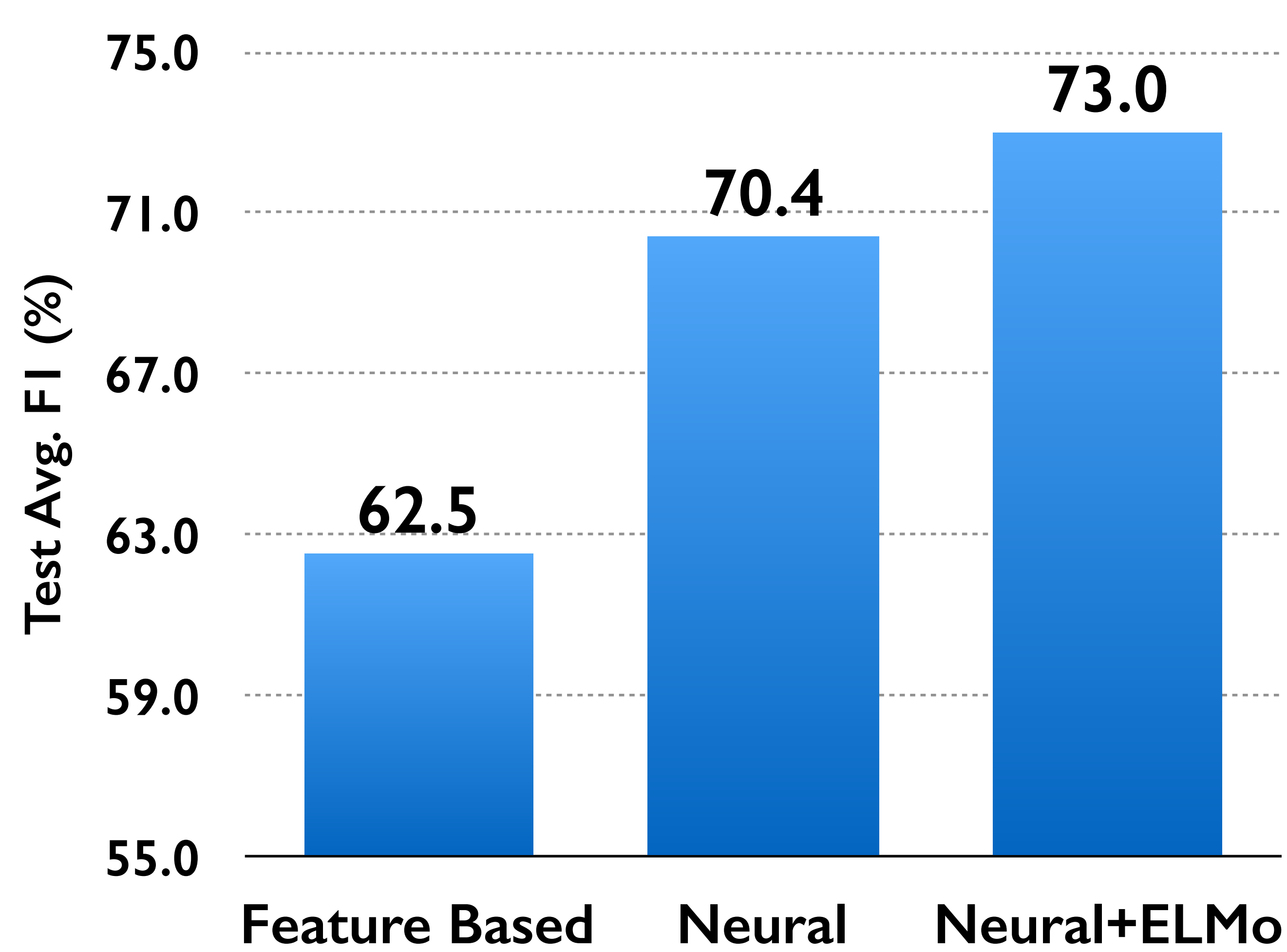
ELMo: Embeddings from a Language Model

Step 1: Train a large BiLM on unlabeled data
Step 2: Compute linear function of pre-trained model
Step 3: Learn weights for each end task



Best Single System Results

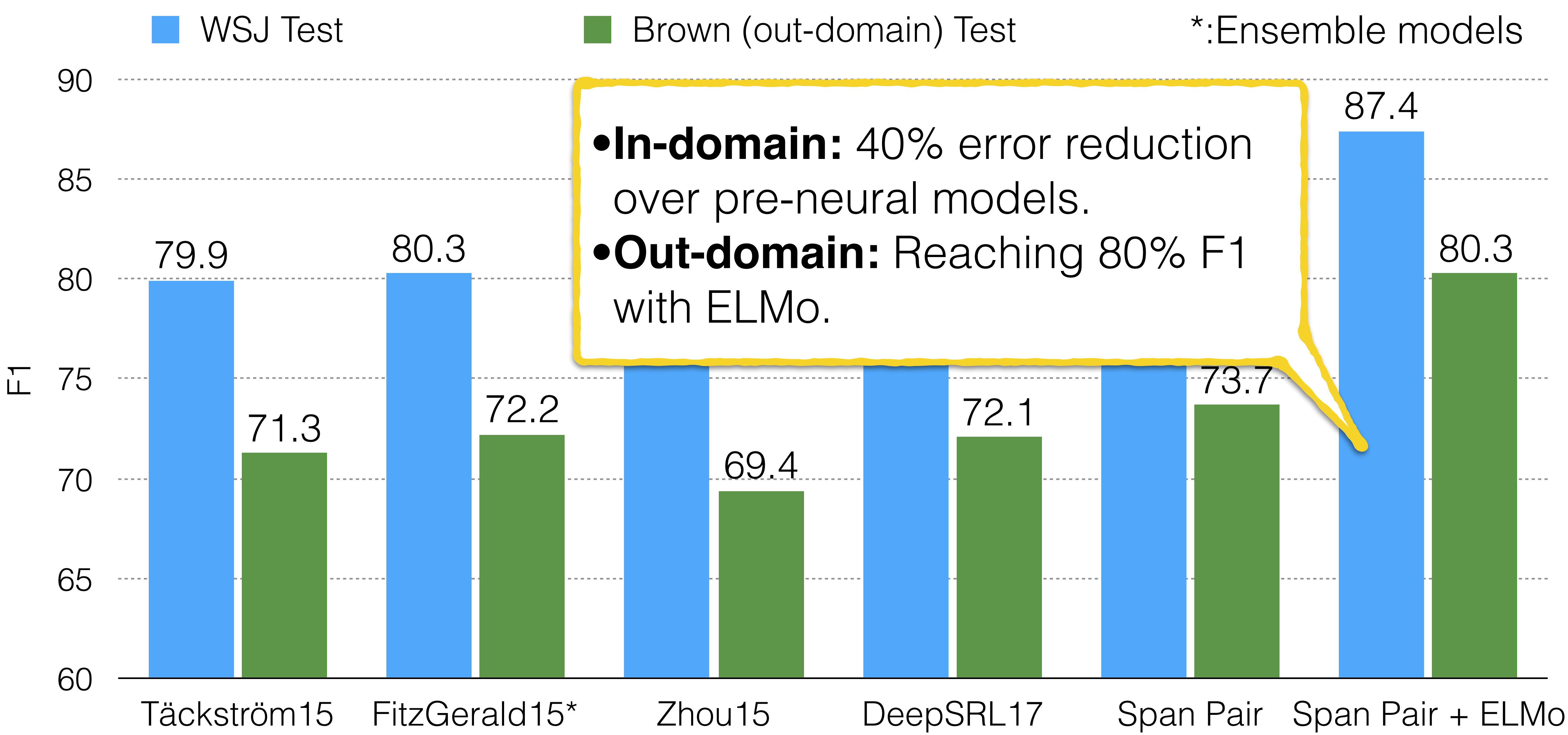




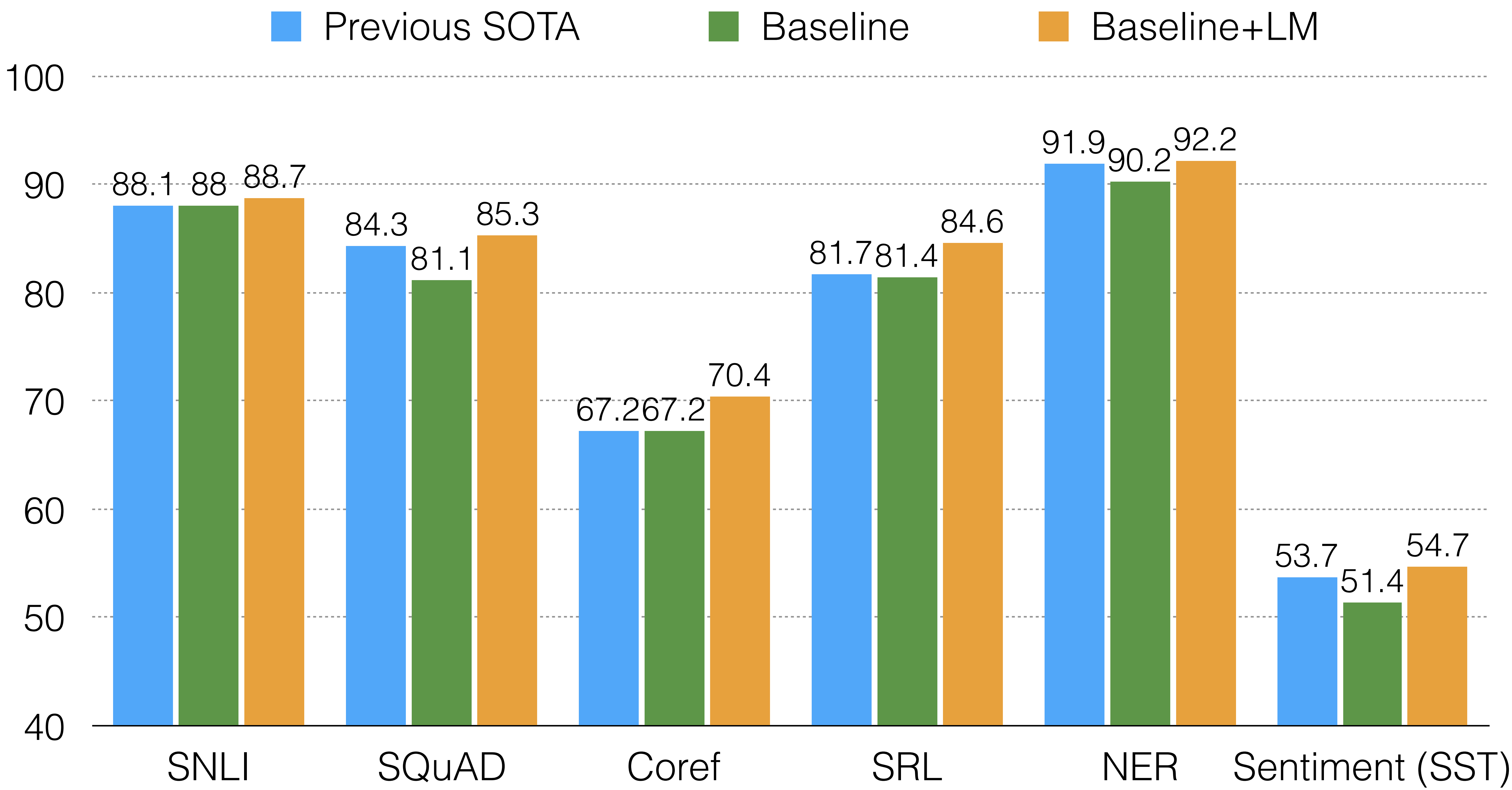
SRL
(+3.5 F1)

Coreference
(+2.6 F1)

Gold Predicates: CoNLL 2005 SRL Results



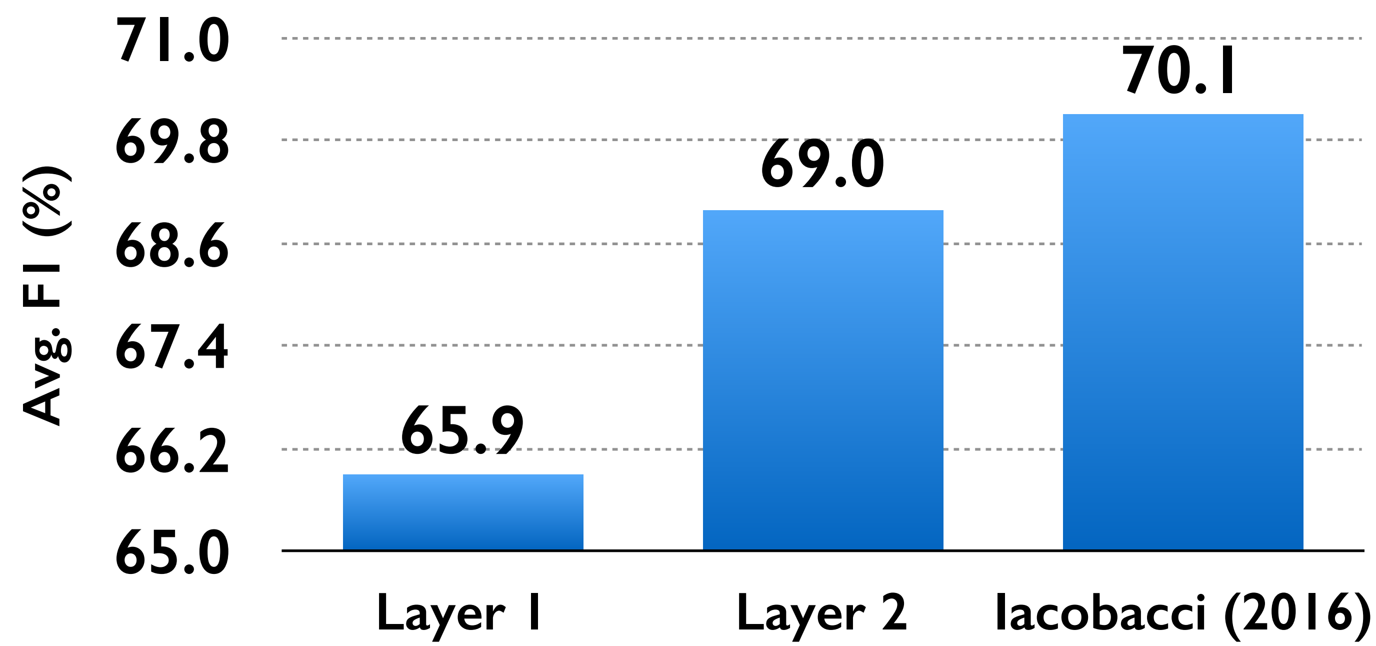
SOTA For Many Others Tasks



What Does it Learn?

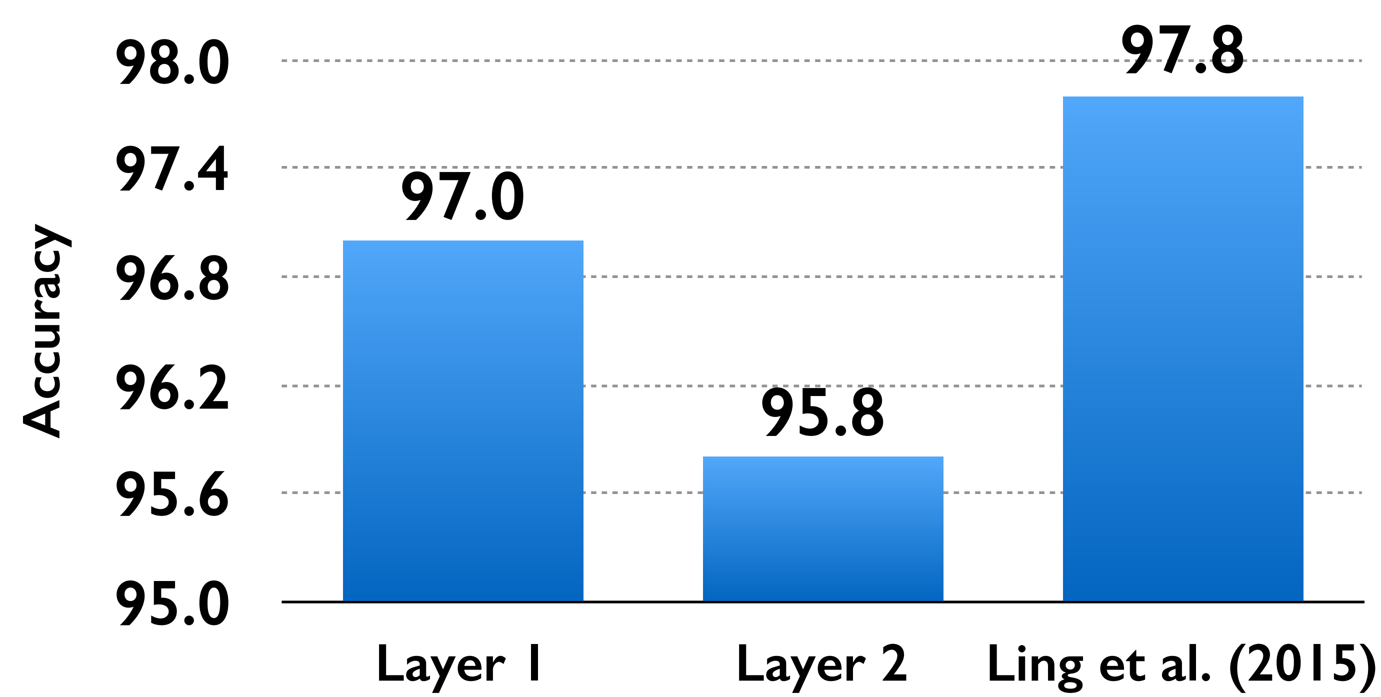
Semantics:

- Supervised WSD task [Miller et al.,1994]
- Use N-th layer in NN classifier



Syntax:

- Label POS corpus [Marcus et al., 1993]
- Learn classifier on N-th layer



Where Will the Data Come From???

Option 1: Semi-supervised learning

- E.g. word2vec and GloVe are in wide use
  [Mikolov et al., 2013; Pennington et al., 2014]
- Can we learn better word representations?

Option 2: Supervised learning

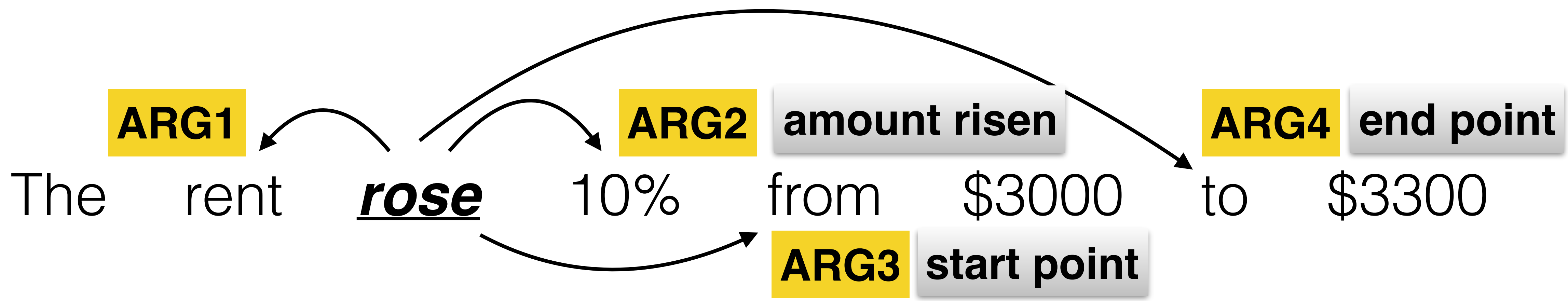
- Can we gather more direct forms of supervision?

A First Data Step: QA-SRL

- Introduce a **new SRL** formulation with **no frame or role inventory**
- Use **question-answer pairs** to model verbal predicate-argument relations
- Annotated **over 3,000 sentences in weeks** with **non-expert**, part-time annotators
- Showed that this data is **high-quality** and **learnable**

[He et al, 2015]

Previous Method: Annotation with Frames



Frameset: rise.01 , *go up*

Arg1-: *Logical subject, patient, thing rising*
Arg2-EXT: *EXT, amount risen*
Arg3-DIR: *start point*
Arg4-LOC: *end point*
Argm-LOC: *medium*

- Depends on pre-defined frame inventory, requires syntactic parses
- Annotators need to:
  1) Identify the Frameset
  2) Find arguments in the parse
  3) Assign labels accordingly
- If frame doesn't exist, create new

The Proposition Bank: An Annotated Corpus of Semantic Roles, Palmer et al., 2005
http://verbs.colorado.edu/propbank/framesets-english/rise-v.html

Our Annotation Scheme

Given sentence and a verb:

They ***increased*** the rent this year .

Step 1: Ask a question about the verb:

Step 2: Answer with words in the sentence:

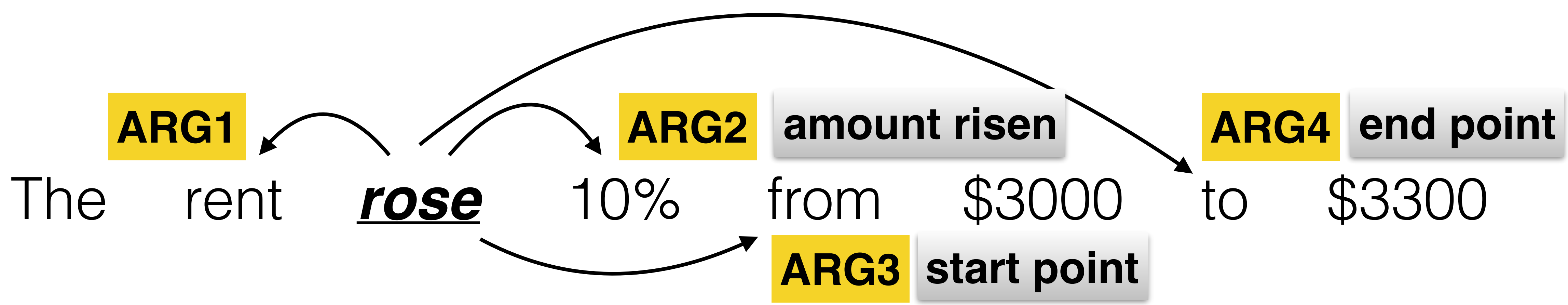
Who increased something ? They

Step 3: Repeat, write as many QA pairs as possible …

What is increased ? the rent

When is something increased ? this year

Our Method: Q/A Pairs for Semantic Relations



Wh-Question	**Answer**
What ***rose*** ?	the rent
How much did something ***rise*** ?	10%
What did something ***rise*** from ?	$3000
What did something ***rise*** to ?	$3300

Wh-words vs. PropBank Roles

	Who	What	When	Where	Why	How	HowMuch
ARG0	1575	414	3	5	17	28	2
ARG1	285	2481	4	25	20	23	95
ARG2	85	364	2	49	17	51	74
ARG3	11	62	7	8	4	16	31
ARG4	2	30	5	11	2	4	30
ARG5	0	0	0	1	0	2	0
AM-ADV	5	44	9	2	25	27	6
AM-CAU	0	3	1	0	23	1	0
AM-DIR	0	6	1	13	0	4	0
AM-EXT	0	4	0	0	0	5	5
AM-LOC	1	35	10	89	0	13	11
AM-MNR	5	47	2	8	4	108	14
AM-PNC	2	21	0	1	39	7	2
AM-PRD	1	1	0	0	0	1	0
AM-TMP	2	51	341	2	11	20	10

Advantages

- Easily explained
- No pre-defined roles, few syntactic assumption
- Can capture implicit arguments
- Generalizable across domains

Limitations

- Only modeling verbs (for now)
- Not annotating verb senses directly
- Can have multiple equivalent questions

Challenges

- What questions to ask?
- How much data do we need?
- Can we generalize to other tasks, such as coref?

Does the Recipe Work for Broad Coverage Semantics?

Step 1: Gather lots of training data!

Challenge 1: Data is costly and limited (e.g. linguists required to label PennTreebank / OntoNotes)

Step 2: Apply Deep Learning!!

Challenge 2: Pipeline of structured prediction problems with cascading errors (e.g. POS->Parsing->SRL->Coref)

Step 3: Observe Impressive Gains!!!

Contributions

Models

- End-to-end deep learning for SRL and coreference
- No preprocessing (e.g. no parser or POS tagger)

Data

- Contextualized word embeddings from a language model
- First steps towards scalable data annotation

The End: Questions?

Future Directions

- Multi-task learning, given architectural similarities
- Multi-lingual should work, in theory…
- Need to scale up data annotation efforts, and focus on out of domain performance

Recent Release

- AllenNLP: Deep Learning Semantic NLP toolkit
- See demos and code at AllenNLP.org

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

AI2 ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE