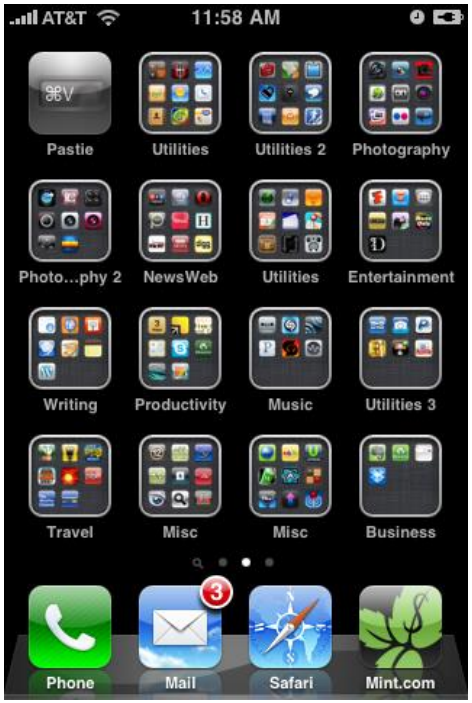
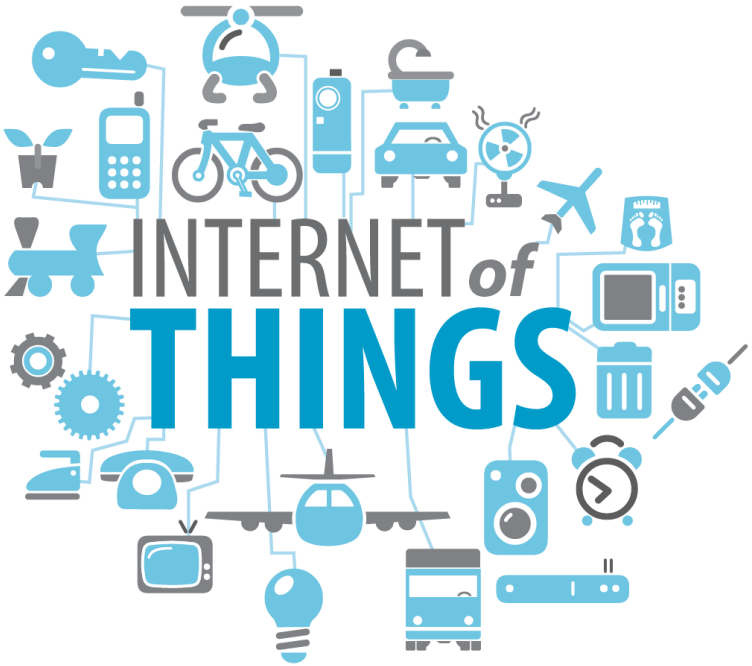
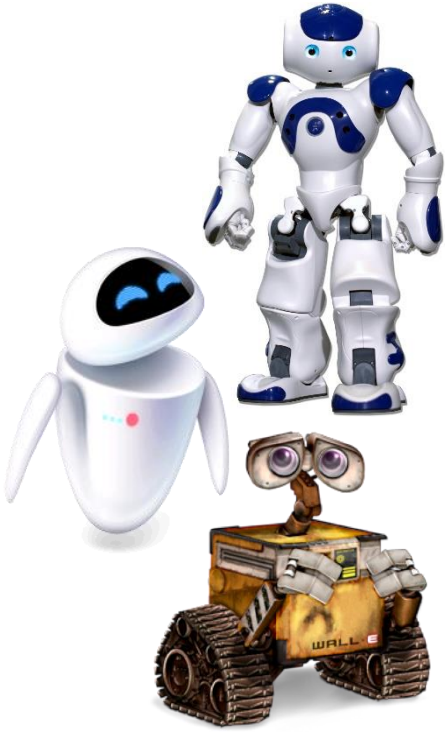
On Building Natural Language Interface to Data and Services

Yu Su
Department of Computer Science
University of California, Santa Barbara

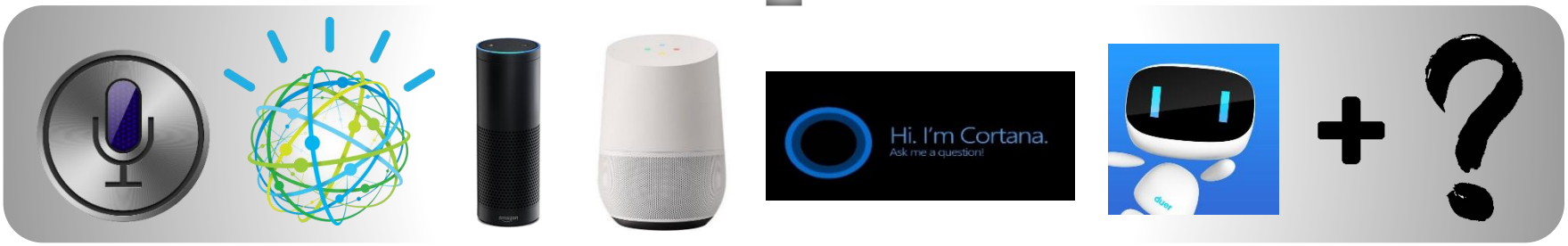
Human-machine interface for digitalized world

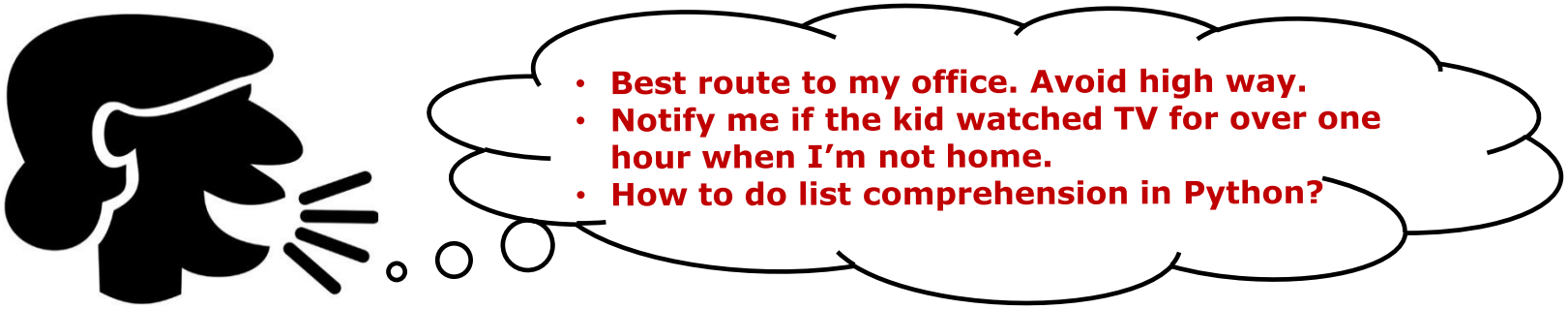




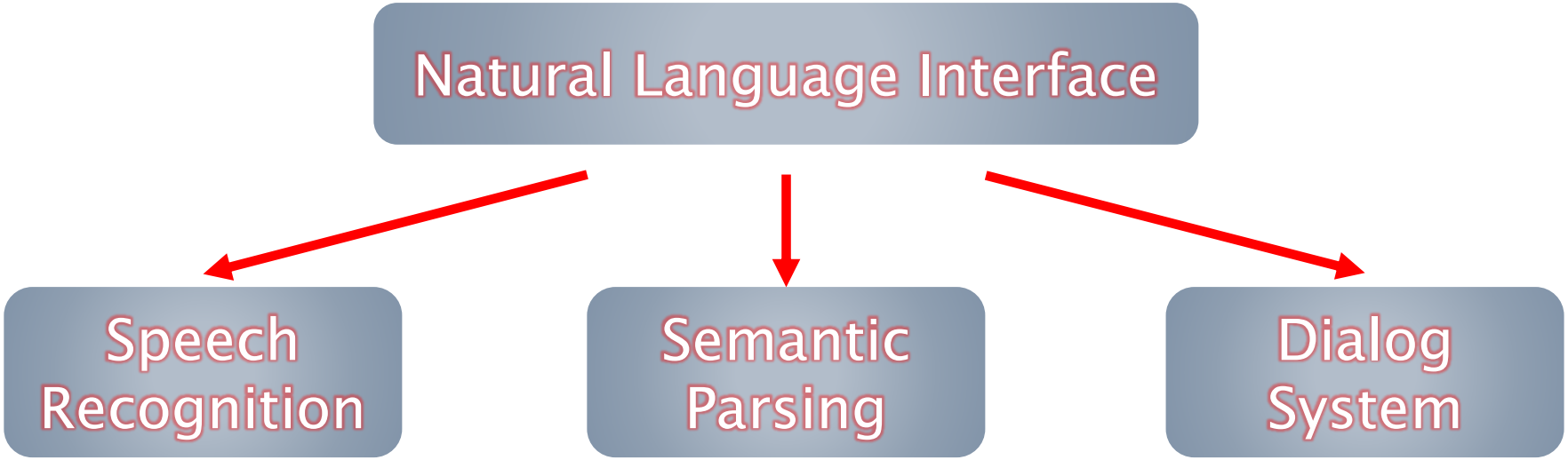


One interface for all





Research on natural language interface



Near human performance

Sentence level
Natural language understanding
Broader domains
More complex inputs

Multi-round Dialog
Dialog management
Discourse analysis
(Currently) limited domains, vocabulary, domain complexity, etc.

Semantic parsing

Natural Language → Formal Language

[Woods et al. 1973]

[Zettlemoyer and Collins 2005]

Relational Database

"What is the largest state?" → argmax(λx.state(x), λx.λy.size(x,y))

[Matuszek et al. 2013]

Robot Instruction

"Take the left, then next right." → (do-sequentially (turn-left) (turn-right))

[Berant et al. 2013]
[Su et al. 2016]

[Yih et al. 2015]

Knowledge Base

"How many children of Eddard Stark were born in Winterfell?" → count(λx.children(Eddard_Stark, x) $\wedge$ place_of_birth(x, Winterfell))

[Su et al. 2017]

API

"How many unread emails about PhD application do I have?" → GET-Messages(FILTER(isRead=False), SEARCH("PhD application"), COUNT())

Table Cell Search for Question Answering (WWW'16)
Improving Semantic Parsing via Answer Type Inference (EMNLP'16)
Cross-domain Semantic Parsing via Paraphrasing (EMNLP'17)
Building Natural Language Interfaces to Web APIs (CIKM'17)

What constitutes a practical NLI?

- Semantic parsing ← most studies
- Scalability ← baby steps
- Interpretability and interactivity ← baby steps
- Robustness (answer triggering, post-inspection, etc.) ← baby steps

Scalability

- Vertical scalability
  - Scale up to more complex inputs and logical constructs

Who was the head coach when Michael Jordan started playing for the Chicago Bulls?

In which season did Michael Jordan get the most points?

What team did Michael Jordan play for?

Scalability

- Vertical scalability
  - Scale up to more complex inputs and logical constructs

- Horizontal scalability
  - Scale out to more domains
  - Weather, calendar, hotel, flight, restaurant, …
  - Knowledge base, relational database, API, robot instruction, …
  - Graph, table, text, image, audio, …

Scalability

- Vertical scalability
  - Scale up to more complex inputs and logical constructs
- Horizontal scalability
  - Scale out to more domains
  - Weather, calendar, hotel, flight, restaurant, …
  - Knowledge base, relational database, API, robot instruction, …
  - Graph, table, text, image, audio, …
- More data! Better (more data-efficient) model!

On Generating Characteristic-rich Question Sets for QA Evaluation (EMNLP'16)
Cross-domain Semantic Parsing via Paraphrasing (EMNLP'17)
Building Natural Language Interfaces to Web APIs (CIKM'17)

Robustness

- Make an NLI more robust without user interaction
- **Answer triggering**: *Is this question answerable by my knowledge sources?*
- **Post-inspection**: *Does the result from semantic parser make sense?*

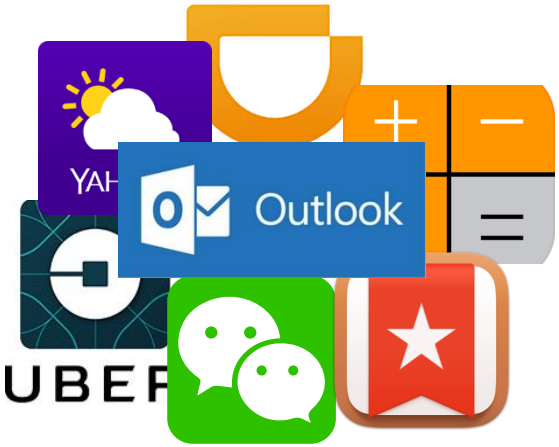
Outline

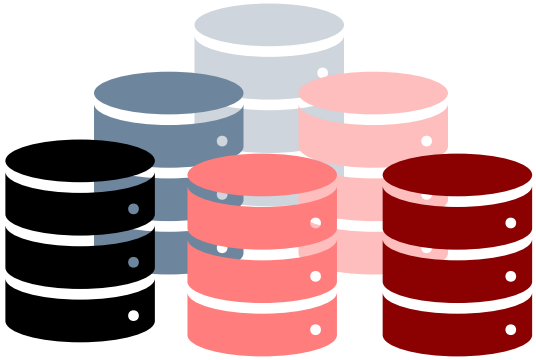
- Introduction
- The cold start problem
  - "*I want to build an NLI for my domain, but I don't have any user and training data*"
  - Can I collect training data via crowdsourcing? (EMNLP'16, CIKM'17)
  - Can I leverage existing training data from other domains? (EMNLP'17)
- Future work

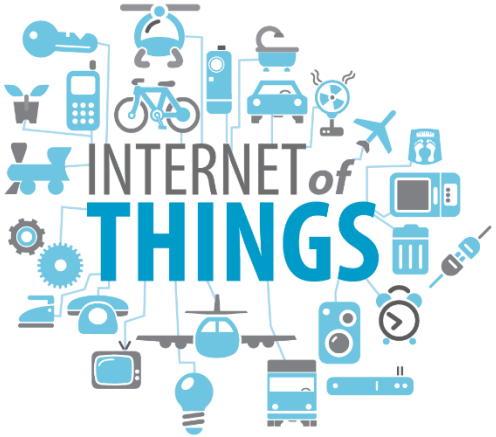
The cold start problem

The cold start problem

"I want to build an NLI for my domain, but I don't have any user and training data"



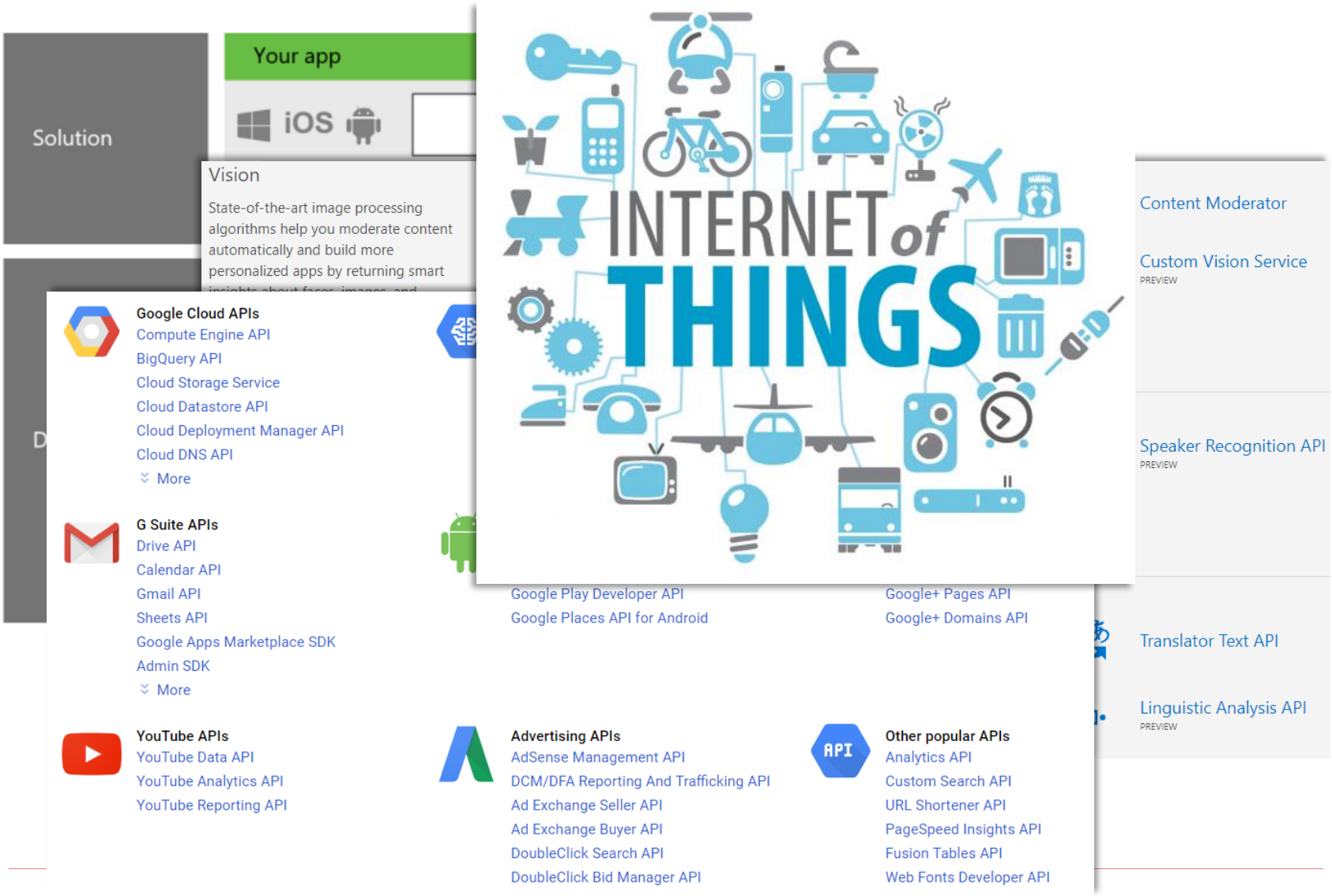




Outline

- Introduction
- The cold start problem
  - "*I want to build an NLI for my domain, but I don't have any user and training data*"
  - **Can I collect training data via crowdsourcing? (EMNLP'16, CIKM'17)**
  - Can I leverage existing training data from other domains? (EMNLP'17)
- Future work

Web API: interface to web services



Natural language interface to web API

Utterance

Show me unread emails about intern NEO, early ones first
Unread intern NEO emails reverse ordered by time
Find those emails containing intern NEO that I have not read, starting with the oldest one
… …

API Frame

HTTP Verb: `GET`
Resource: `Messages`
Return Type: `{Message}`
Required Parameters: `{ }`
Optional Parameters: `{`

```
      FILTER(isRead = False),
      SEARCH("intern NEO"),
      ORDERBY(receivedDateTime, asc)
}
```

API Call

```
GET https://graph.microsoft.com/v1.0/<user-id>/messages?
$filter=isRead%20eq%20false&
$search="intern%20neo"&
$orderby=receivedDateTime%20asc
```

Why NLI to API?

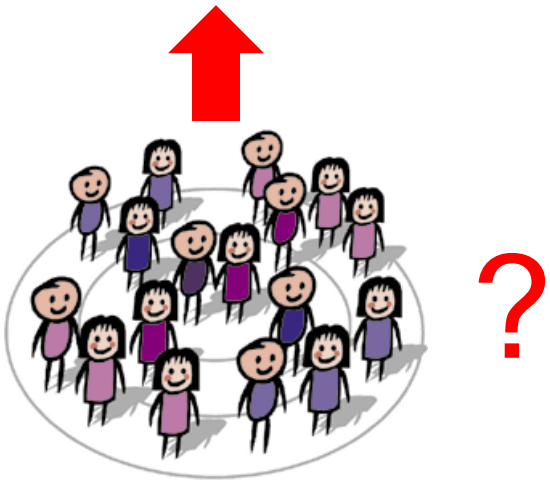
- Distributed development of virtual assistants
- Developer assistant: program by dictation
- Enterprise intelligence
- Search, discovery, recommendation of services / APIs

What we did in this work

- The first framework for building an NLI for a given web API starting from zero user and data
- Probabilistic modeling of the crowdsourcing process for NLI
- Optimization of the crowdsourcing process for NLI
  - Select which API calls to annotate?

The data challenge of NLI

"How many unread emails about PhD application do I have?"



```
GET-Messages{(FILTER(isRead=False),
SEARCH("PhD application"), COUNT()}
```

Low-cost data collection via crowdsourcing

HUGE SPACE!

WARNING

API Call

```
GET-Events{SELECT(location), TOP(1)
FILTER(start > now), ORDERBY(start, asc)}
```

Automatic

Canonical Command

get the location of the top 1 event that start time is after now, and ordered by start time in ascending order

Crowdsourcing

Paraphrases from Crowdsourcing

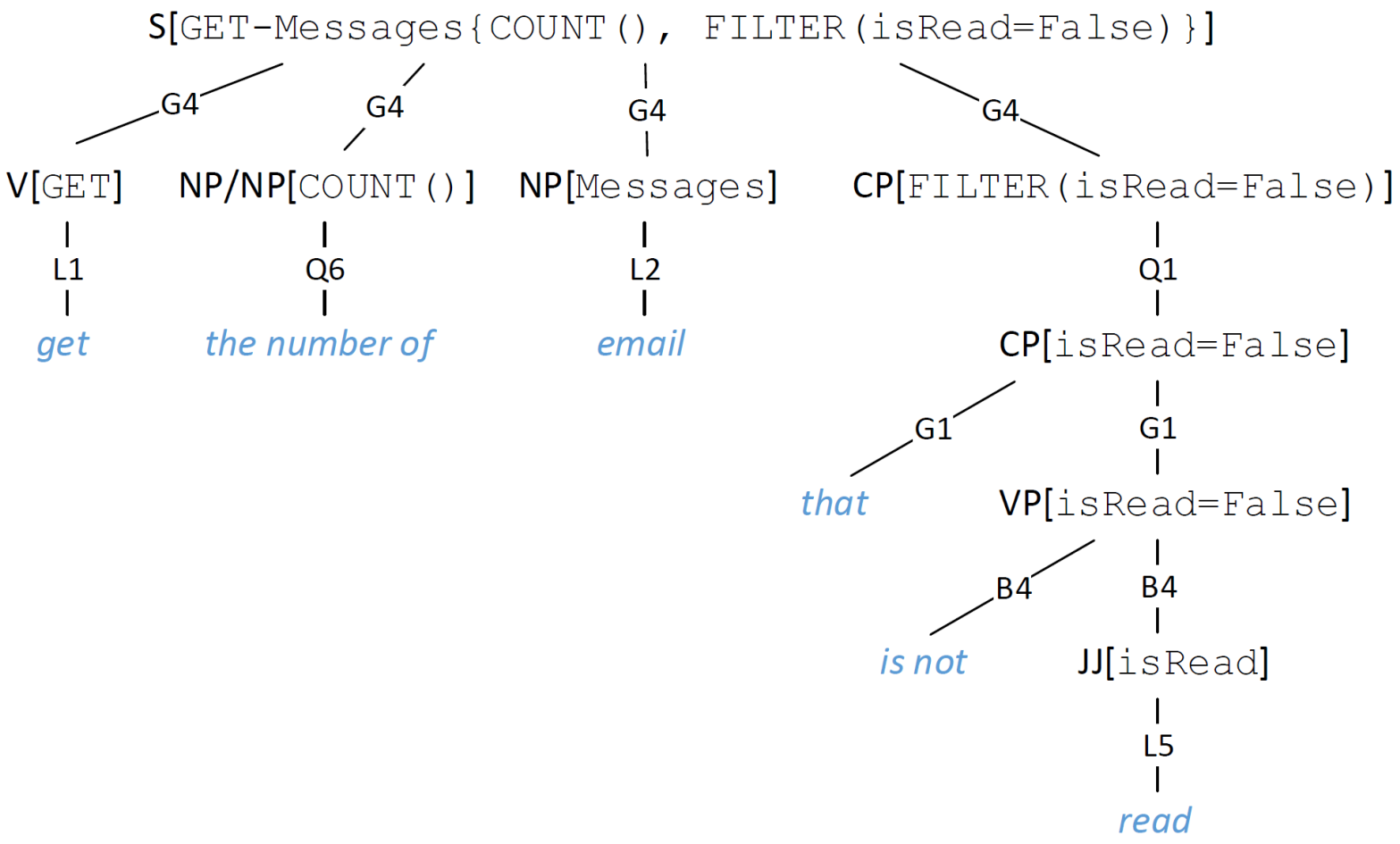
Where is my next meeting?
What's the location of my next event?
Find next meeting's location.
… …

Wang et al. Building a Semantic Parser Overnight (ACL'15)

Canonical utterance generation: grammar

	[Lexicon]		
(L1)	get	→	V[GET]
(L2)	email	→	NP[Messages]
(L3)	sender	→	NP[from]
(L4)	in the category of	→	PP/NP[categories]
(L5)	read	→	JJ[isRead]
(L6)	have attachment	→	VP[hasAttachments]
(L7)	receive time	→	NP[receivedDateTime]
…		…	
	[Boolean Expression]		
(B1)	VP[x]	→	VP[x = True]
(B2)	do not VP[x]	→	VP[x = False]
(B3)	is JJ[x]	→	VP[x = True]
(B4)	is not JJ[x]	→	VP[x = False]
(B5)	is PP/NP[x] NP[y]	→	VP[x = y]
(B6)	NP[x] is before|after DATETIME[y]	→	NP[x <|> y]
(B7)	NP[x] is|is smaller than|is larger than *[y]	→	NP[x =|<|> y]
	[Query Option]		
(Q1)	CP[x]	→	CP[FILTER(x)]
(Q2)	that contain NP[x]	→	CP[SEARCH(x)]
(Q3)	ordered by NP[x] in ascending|descending order	→	CP[ORDERBY(x, 'asc'|'desc')]
(Q4)	the NP[x] of	→	NP/NP[SELECT(x)]
(Q5)	the top NUMBER[x] of	→	NP/NP[TOP(x)]
(Q6)	the number of	→	NP/NP[COUNT())]
	[Glue]		
(G1)	that VP[x]	→	CP[x]
(G2)	CP[x], and CP[y]	→	CP[x, y]
(G3)	NP/NP[x] NP/NP[y]	→	NP/NP[x, y]
(G4)	V[x] NP/NP[w] NP[y] CP[u]	→	S[x-y{w, u}]

Canonical utterance generation: example



The data challenge (cont'd)

- **Problem**: thousands of API calls can be generated for each API
  - Neither economic nor necessary to annotate all of them
  - *How to optimize the crowdsourcing process?*

- **Solution**: Actively and iteratively select a subset of the most confusing API calls to annotate

- **Challenge**: How to quantify "confusion"?

Interplay of utterance and API call

- u: an utterance, z: an API call
- Natural language interface: estimate $p(z|u)$
- Conditional language model: estimate $p(u|z)$

Definition. Let θ^z: $p(u|z)$,
z_1 and z_2 cause confusion
if θ^{z_1} and θ^{z_2} are similar

Estimate θ^z

- If an API call has been annotated via crowdsourcing, use maximum likelihood estimation

API Call

```
GET-Events{SELECT(location), TOP(1)
FILTER(start > now), ORDERBY(start, asc)}
```

Paraphrases from Crowdsourcing

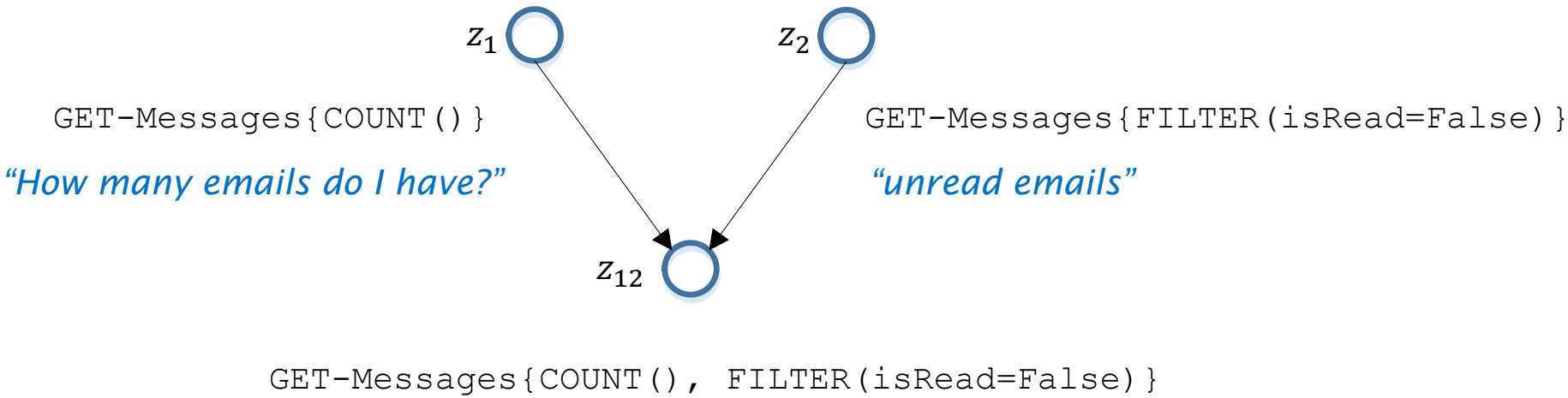
Where is my next meeting?
What's the location of my next event?
Find next meeting's location.
… …

Estimate θ^z

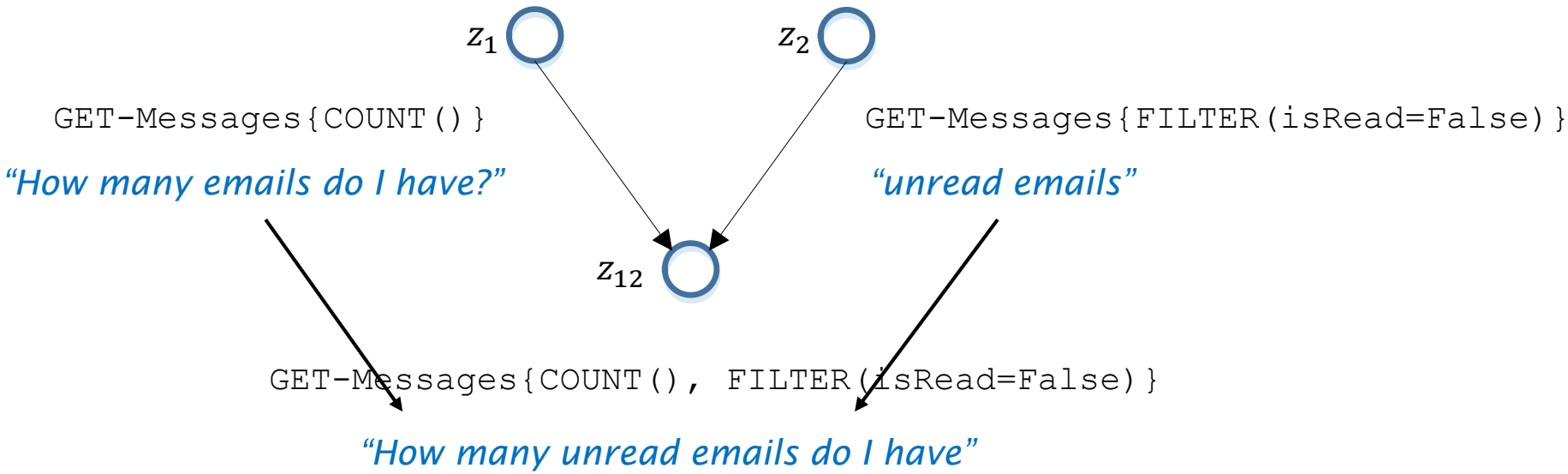
- If an API call has been annotated via crowdsourcing, use maximum likelihood estimation

- Otherwise, leverage the compositionality of API calls (or formal languages in general)

Compositionality



Compositionality

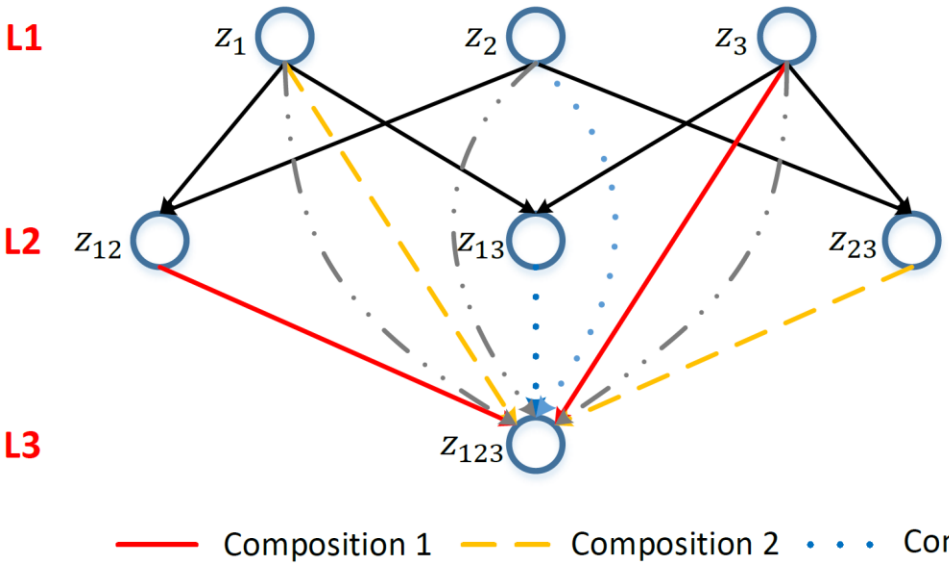


ASSUMPTION:
Utterances follow the composition structure of API calls

Predict the language model of an API call **without annotating it!**

Crowdsourcing optimization

Semantic mesh: hierarchical probabilistic model for crowdsourcing



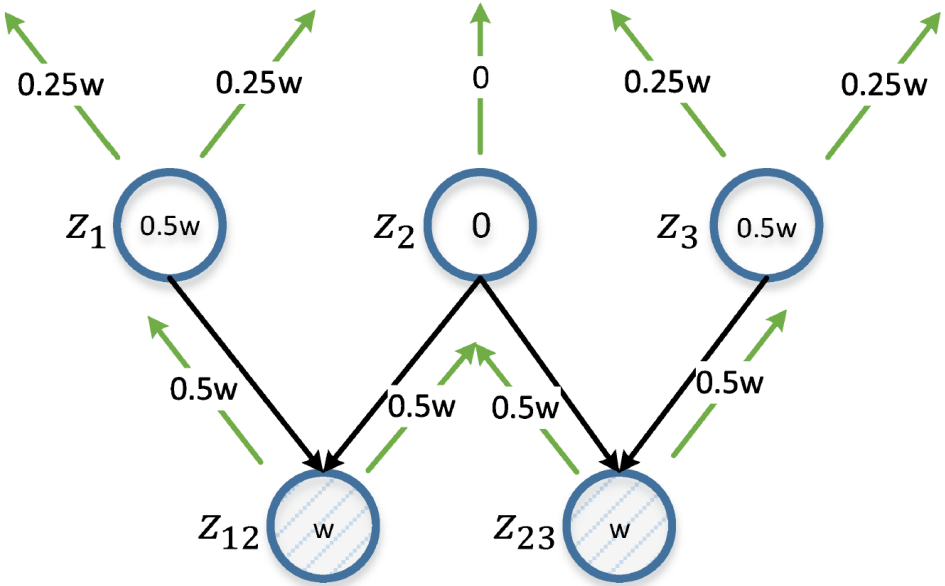
Node Properties	
$\boldsymbol{u}^z$	A set of crowdsourced utterances for z
θ^z	A distribution describing the generative process from z to utterances
θ^z_{em}	Empirical distribution estimated from $\boldsymbol{u}^z$
θ^z_{ex}	Expected distribution estimated from compositions

Table. Node Operations

ANNOTATE	collect $\boldsymbol{u}$ from crowd, estimate θ_{em} using MLE
COMPOSE	$\theta_{ex} = f(\theta^1, \ldots, \theta^n)$
INTERPOLATE	$\theta = \alpha * \theta_{em} + (1 - \alpha) * \theta_{ex}$

Optimization: differential propagation (DP)

- Objective: maximize pairwise distance $d(z_i, z_j)$



$$w = \min(1.0, \frac{1}{d(z_i, z_j)})$$

Evaluation: data collection via crowdsourcing

- **Input**: the specification of two Microsoft Graph APIs
- **Output**: a set of training data for each API
- **Cost and efficiency**: 82 examples from a worker per hour at a cost of $8.2 on average
- **Quality**: error rate = 17.4% by manual evaluation, on par with related work

# of API calls	# of workers	Avg. time per example	$ per example
1141	201	44s	$0.1

Utterances DO follow composition of API calls

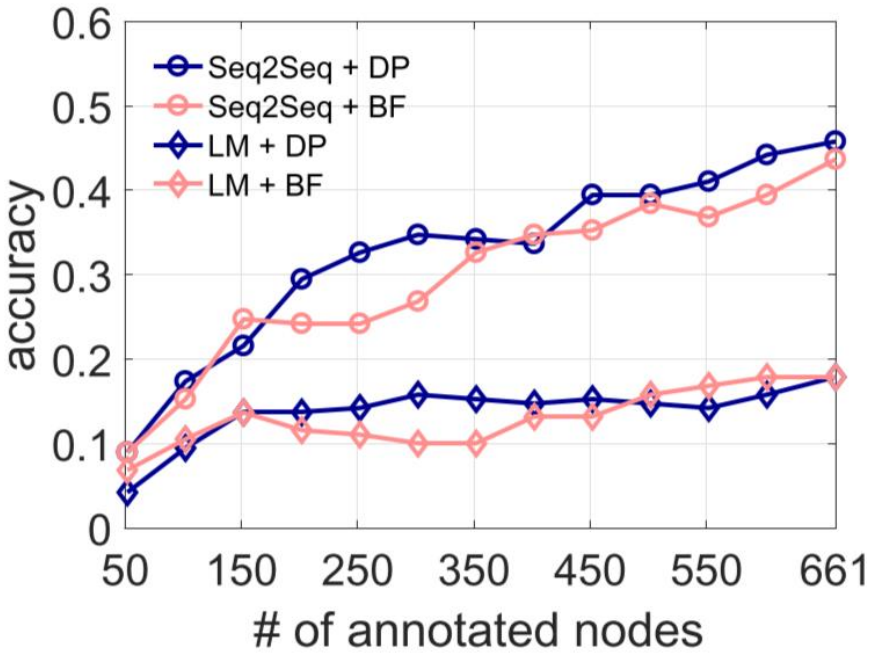
- LM: language model based retrieval model
- # of annotation: ROOT (~20), TOP2 (~100), TOP3 (~500)

		ROOT	TOP2	TOP3
GET-Messages	LM	3.18	12.1	25.5
	LM + SemMesh	**46.5**	**47.8**	50.3
	Seq2Seq	5.73	22.9	**57.3**
GET-Events	LM	7.89	13.7	17.4
	LM + SemMesh	**39.0**	**42.6**	44.2
	Seq2Seq	8.95	20.5	**45.3**

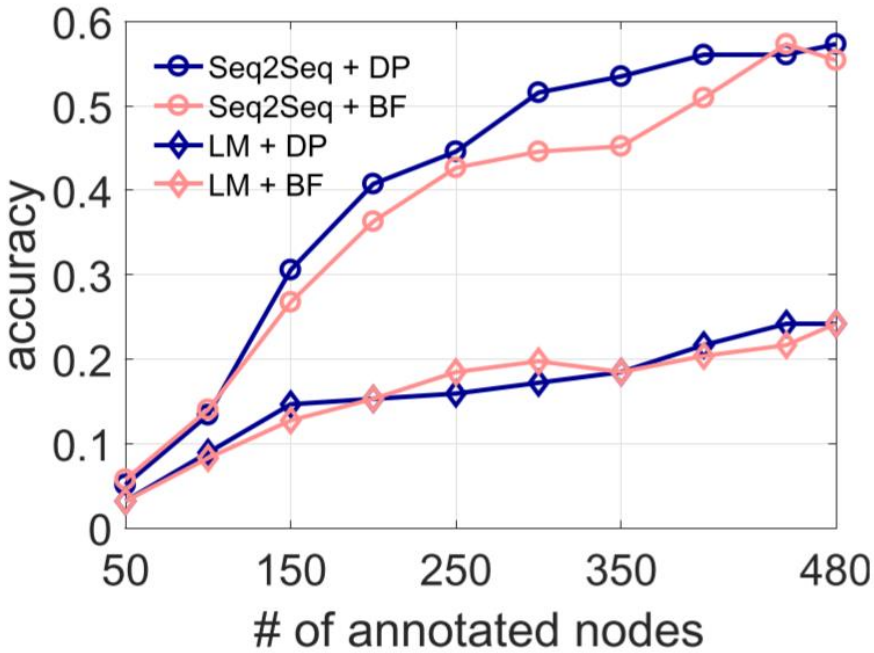
Table 1. Accuracy of different parsing models with different training sets.

Crowdsourcing optimization

- Baseline: Breadth-first annotation (BF)
- Our method: Differential propagation (DP)



(a) GET-Events



(b) GET-Messages

Outline

- Introduction
- The cold start problem
  - "*I want to build an NLI for my domain, but I don't have any user and training data*"
  - Can I collect training data via crowdsourcing? (EMNLP'16, CIKM'17)
  - **Can I leverage existing training data from other domains? (EMNLP'17)**
- Future work

What is transferrable in semantic parsing?

In which season did Kobe Bryant play for the Lakers?

$\Rightarrow$

$\mathbf{R}[\texttt{season}].(\texttt{player.KobeBryant} \sqcap \texttt{team.Lakers})$

What is transferrable in semantic parsing?

In which season did Kobe Bryant play for the Lakers?

$\mathbf{R}[\texttt{season}].(\texttt{player.KobeBryant} \sqcap \texttt{team.Lakers})$

$p(\texttt{team}|\text{"}\textit{play for}\text{"})$

What is transferrable in semantic parsing?

In which season did Kobe Bryant play for the Lakers?

$\mathbf{R}[\texttt{season}].(\texttt{player.KobeBryant} \sqcap \texttt{team.Lakers})$

$p(\texttt{team}|\text{"}\textit{play for}\text{"})$

When did Alice start working for Mckinsey?

$\mathbf{R}[\texttt{start}].(\texttt{employee.Alice} \sqcap \texttt{employer.Mckinsey})$

What is transferrable in semantic parsing?

In which season did Kobe Bryant play for the Lakers?

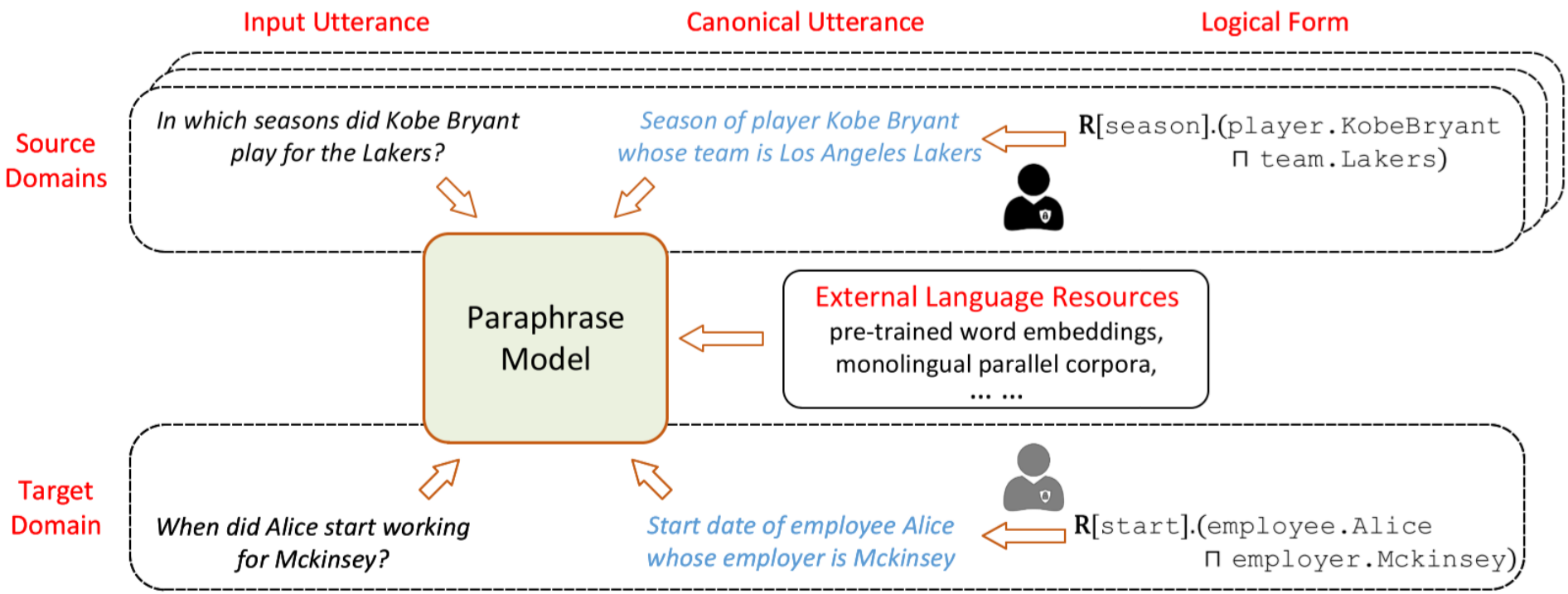
$\mathbf{R}[\texttt{season}].(\texttt{player.KobeBryant} \sqcap \texttt{team.Lakers})$

$p(\texttt{team}|\text{"}\textit{play for}\text{"})$

When did Alice start working for Mckinsey?

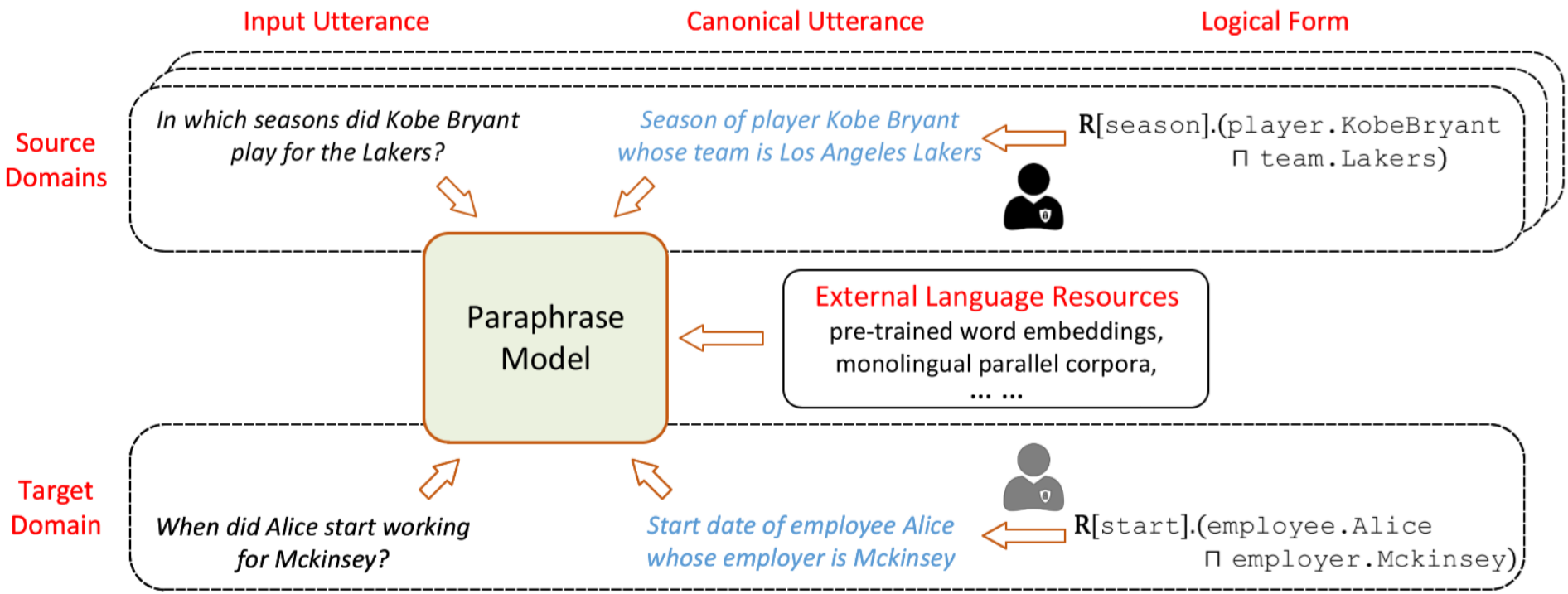
$\mathbf{R}[\texttt{start}].(\texttt{employee.Alice} \sqcap \texttt{employer.Mckinsey})$

Cross-domain semantic parsing via paraphrasing

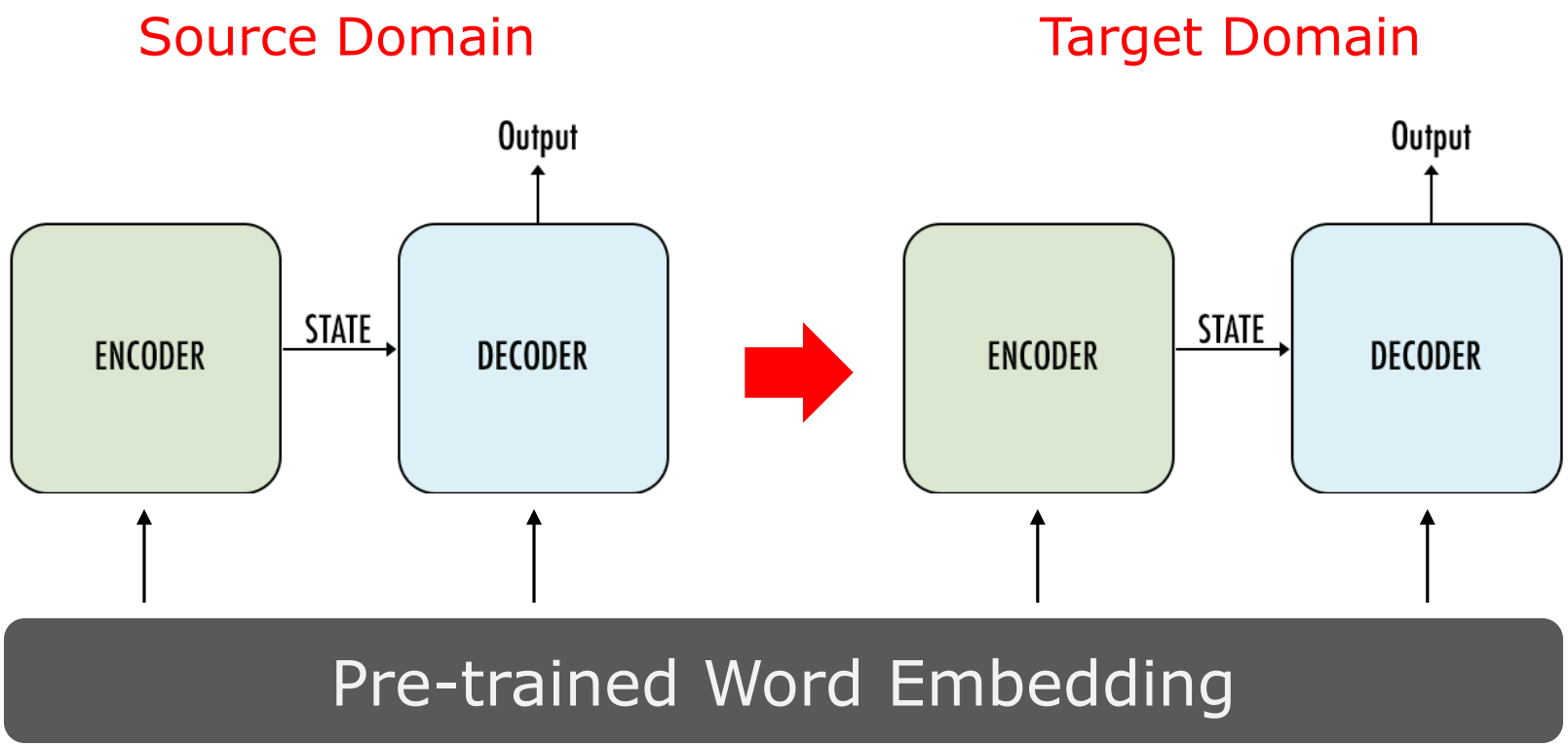


Transfer across domains becomes possible

- Source domain: “play for” ⇒ “whose team is”
- Word embedding: “play” ⇒ “work”, “team” ⇒ “employer”
- Target domain: “work for” ⇒ “whose employer is”



Neural transfer learning for semantic parsing



Vocabulary shifting

- Only 45%~70% target domain vocabulary are covered by source domains
- Pre-trained word embedding can potentially alleviate the vocabulary shifting problem
  - Word2vec: 300-d vectors pre-trained on the 100B-token Google News Corpus; vocabulary size = 3M

Metric	CALENDAR	BLOCKS	HOUSING	RESTAURANTS	PUBLICATIONS	RECIPES	SOCIAL	BASKETBALL
# of example (N)	837	1995	941	1657	801	1080	4419	1952
# of logical form ($\lvert\mathcal{Z}\rvert$, $\lvert\mathcal{C}\rvert$)	196	469	231	339	149	124	624	252
vocab. size ($\lvert\mathcal{V}\rvert$)	228	227	318	342	203	256	533	360
% $\in$ other domains	71.1	61.7	60.7	55.8	65.6	71.9	46.0	45.6
% $\in$ WORD2VEC	91.2	91.6	88.4	88.6	91.1	93.8	86.9	86.9
% $\in$ other domains + WORD2VEC	**93.9**	**93.8**	**90.9**	**90.4**	**95.6**	**97.3**	**89.3**	**89.4**

Dataset: Overnight, 8 closed-domain KBs

[1] Wang et al. Building a Semantic Parser Overnight. ACL'15

Problems of pre-trained word embedding

- Small *micro variance*: hurt optimization
- Large *macro variance*: hurt generalization

Initialization	L2 norm	Variance	Cosine Sim.
Random	17.3 ± 0.45	1.00 ± 0.05	0.00 ± 0.06
WORD2VEC	2.04 ± 1.08	0.02 ± 0.02	0.13 ± 0.11
WORD2VEC + ES	17.3 ± 0.05	1.00 ± 0.00	0.13 ± 0.11
WORD2VEC + FS	16.0 ± 8.47	1.09 ± 1.31	0.12 ± 0.10
WORD2VEC + EN	1.00 ± 0.00	0.01 ± 0.00	0.13 ± 0.11

Random: randomly draw from $U(-\sqrt{3}, \sqrt{3})$
ES: per-example standardization (per row)
FS: per-feature standardization (per column)
EN: per-example normalization

Problems of pre-trained word embedding

- Small *micro variance*: hurt optimization
- Large *macro variance*: hurt generalization

Initialization	L2 norm	Variance	Cosine Sim.
Random	17.3 ± 0.45	1.00 ± 0.05	0.00 ± 0.06
WORD2VEC	2.04 ± 1.08	0.02 ± 0.02	0.13 ± 0.11
WORD2VEC + ES	17.3 ± 0.05	1.00 ± 0.00	0.13 ± 0.11
WORD2VEC + FS	16.0 ± 8.47	1.09 ± 1.31	0.12 ± 0.10
WORD2VEC + EN	1.00 ± 0.00	0.01 ± 0.00	0.13 ± 0.11

Random: randomly draw from $U(-\sqrt{3}, \sqrt{3})$
ES: per-example standardization (per row)
FS: per-feature standardization (per column)
EN: per-example normalization

Problems of pre-trained word embedding

- Small *micro variance*: hurt optimization
- Large *macro variance*: hurt generalization

Initialization	L2 norm	Variance	Cosine Sim.
Random	17.3 ± 0.45	1.00 ± 0.05	0.00 ± 0.06
WORD2VEC	2.04 ± 1.08	0.02 ± 0.02	0.13 ± 0.11
WORD2VEC + ES	17.3 ± 0.05	1.00 ± 0.00	0.13 ± 0.11
WORD2VEC + FS	16.0 ± 8.47	1.09 ± 1.31	0.12 ± 0.10
WORD2VEC + EN	1.00 ± 0.00	0.01 ± 0.00	0.13 ± 0.11

Random: randomly draw from $U(-\sqrt{3}, \sqrt{3})$
ES: per-example standardization (per row)
FS: per-feature standardization (per column)
EN: per-example normalization

Problems of pre-trained word embedding

- Small *micro variance*: hurt optimization
- Large *macro variance*: hurt generalization

Initialization	L2 norm	Variance	Cosine Sim.
Random	17.3 ± 0.45	1.00 ± 0.05	0.00 ± 0.06
WORD2VEC	2.04 ± 1.08	0.02 ± 0.02	0.13 ± 0.11
WORD2VEC + ES	17.3 ± 0.05	1.00 ± 0.00	0.13 ± 0.11
WORD2VEC + FS	16.0 ± 8.47	1.09 ± 1.31	0.12 ± 0.10
WORD2VEC + EN	1.00 ± 0.00	0.01 ± 0.00	0.13 ± 0.11

Random: randomly draw from $U(-\sqrt{3}, \sqrt{3})$
ES: per-example standardization (per row)
FS: per-feature standardization (per column)
EN: per-example normalization

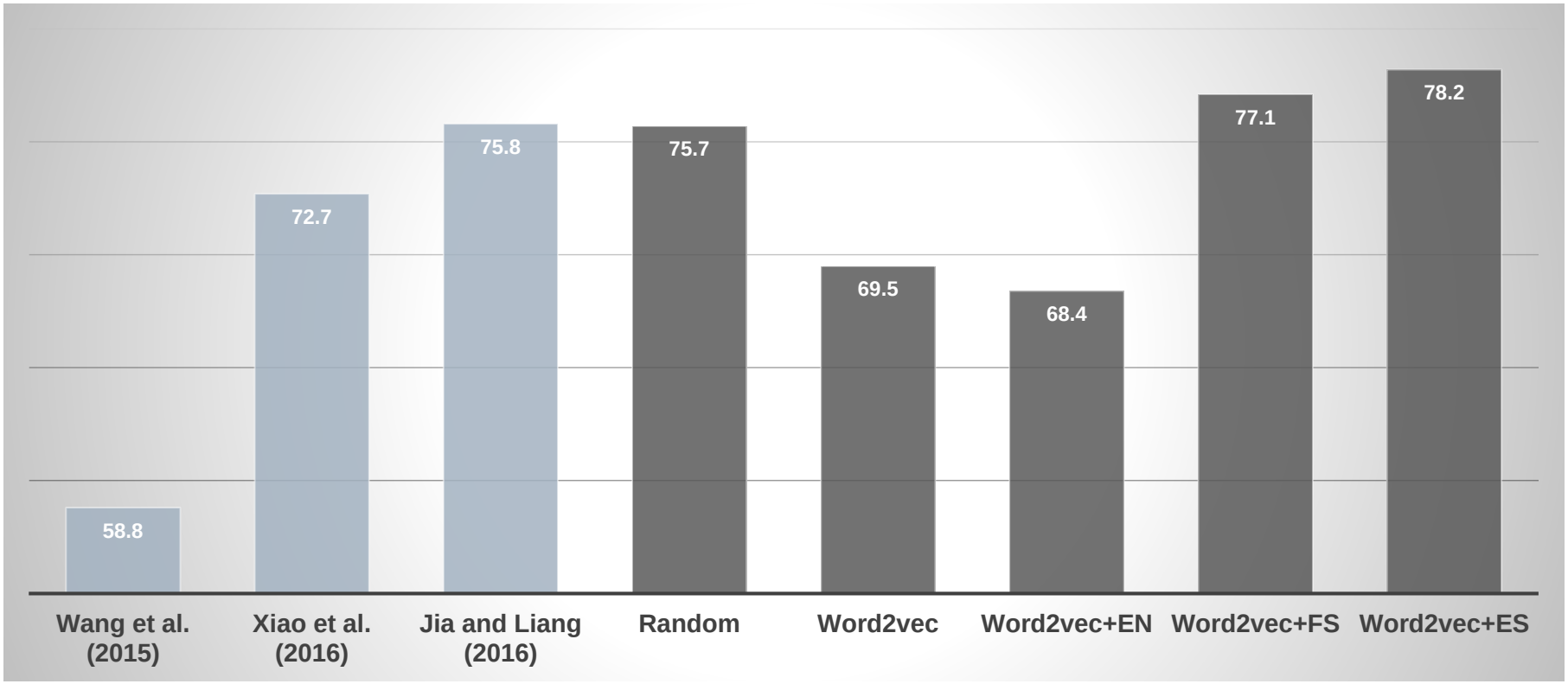
Problems of pre-trained word embedding

- Small *micro variance*: hurt optimization
- Large *macro variance*: hurt generalization

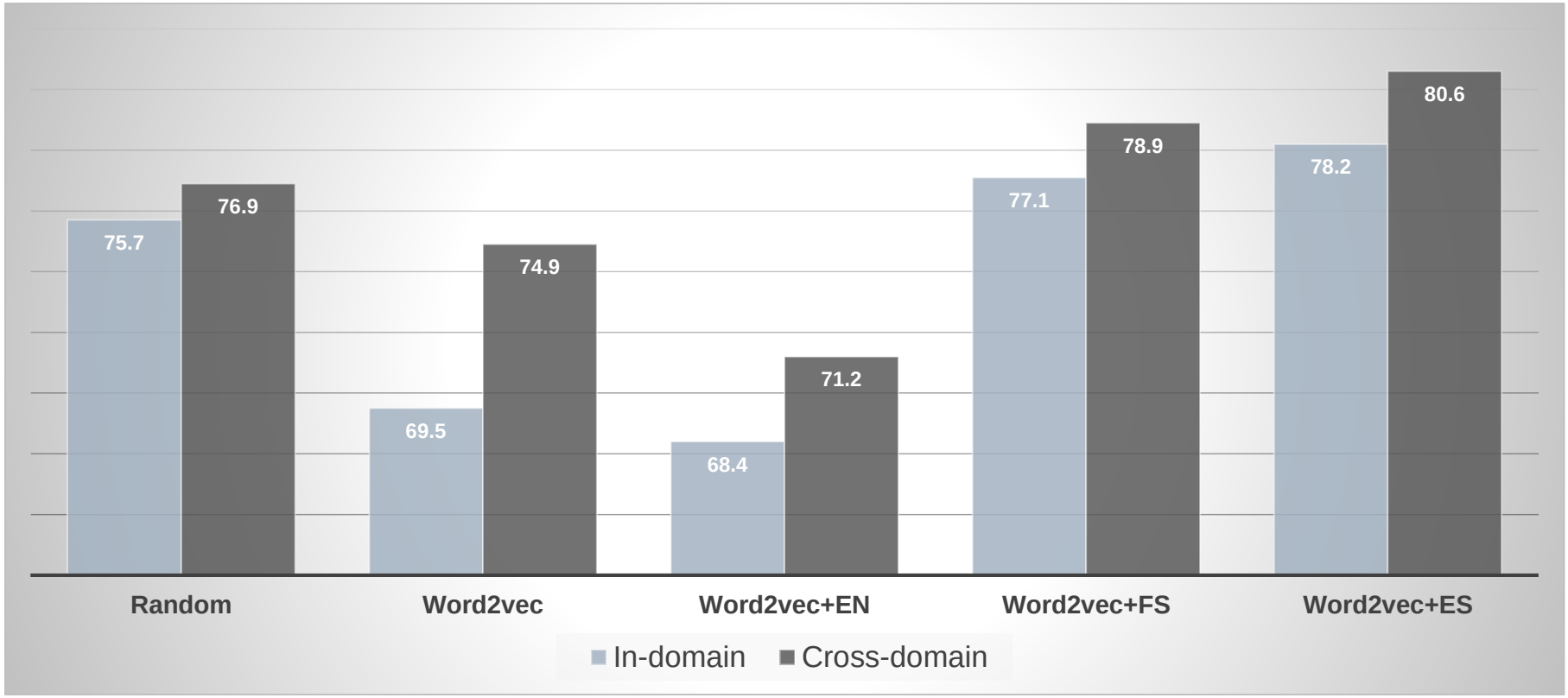
Initialization	L2 norm	Variance	Cosine Sim.
Random	17.3 ± 0.45	1.00 ± 0.05	0.00 ± 0.06
WORD2VEC	2.04 ± 1.08	0.02 ± 0.02	0.13 ± 0.11
WORD2VEC + ES	17.3 ± 0.05	1.00 ± 0.00	0.13 ± 0.11
WORD2VEC + FS	16.0 ± 8.47	1.09 ± 1.31	0.12 ± 0.10
WORD2VEC + EN	1.00 ± 0.00	0.01 ± 0.00	0.13 ± 0.11

Random: randomly draw from $U(-\sqrt{3}, \sqrt{3})$
ES: per-example standardization (per row)
FS: per-feature standardization (per column)
EN: per-example normalization

In-domain evaluation



Cross-domain evaluation

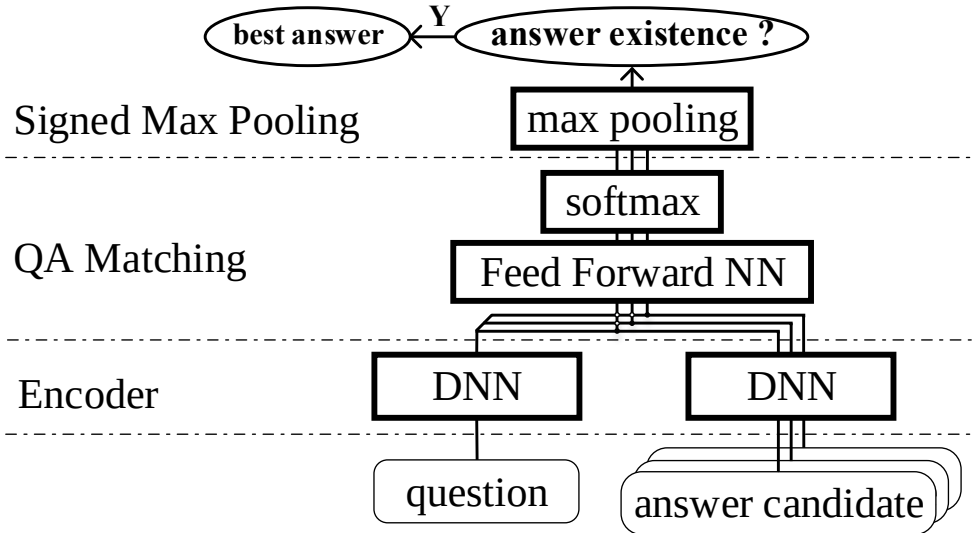


Recap

- "*I want to build an NLI for my domain, but I don't have any training data*"
- Can I collect training data via crowdsourcing?
  - Yes, and it's not so expansive
  - Cost can be further reduced by crowdsourcing optimization
- Can I leverage existing training data from other domains?
  - Yes, if you turn it into a paraphrasing problem
  - Pre-trained word embedding can greatly help neural transfer learning, but only when properly standardized

Answer triggering

- **Motivation**: *Most QA systems always output an answer for any question, no matter whether it's answerable*
- **Definition**: Given a question and a set of answer candidates, determine whether the candidate set contains any correct answer, and if so, select a correct answer as system output

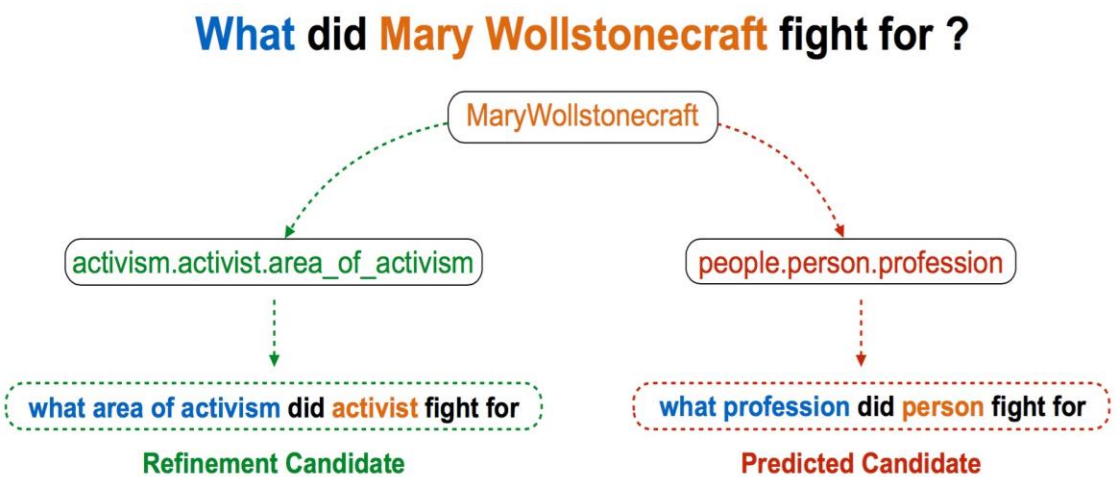


Method	F1
(Yang et al., 2015)	32.2
(Jurczyk et al., 2016)	36.7
Ours	**43.3**

Dataset: WikiQA

Post-inspection

- **Motivation**: *Before execution, can we check whether the parse result makes sense?*
- **Solution**: Cross-check the rational behind the parse result by revising the original question



Method	F1
STAGG (Yih et al., 2015)	52.5
(Xu et al., 2016) (SOTA)	53.8
STAGG + Ours	**53.9**

Dataset: WebQuestions

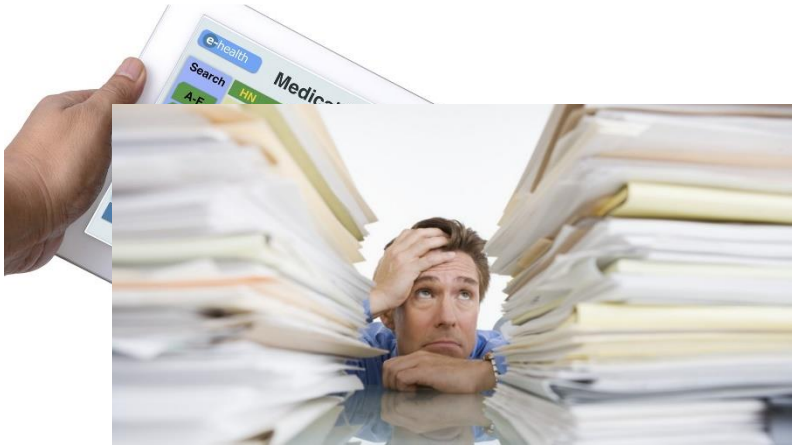
Outline

- Introduction
- The cold start problem
  - "*I want to build an NLI for my domain, but I don't have any user and training data*"
  - Can I collect training data via crowdsourcing? (EMNLP'16, CIKM'17)
  - Can I leverage existing training data from other domains? (EMNLP'17)
- Future work

One interface for all

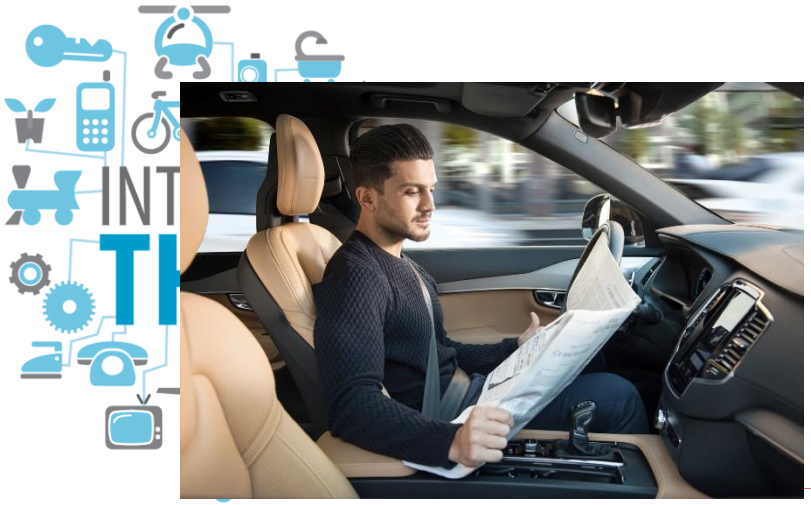
- One NLI simultaneously supports all domains
- Incrementally learn new domains without forgetting (or instead boosting) existing ones

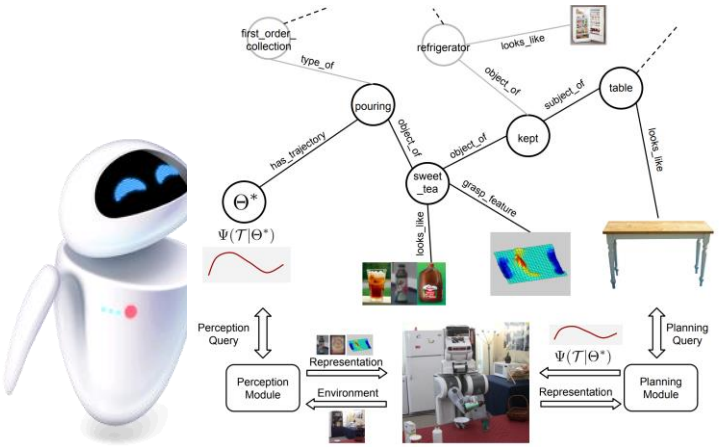
Application to important vertical domains



"Which cement stocks go up the most when a Category 3 hurricane hits Florida?"

KENSHO





Programming: the new inequality

We are living in a divided world

programmers

non-programmers

Ultimate Goal

Natural language as a programming language

— Let everyone be a programmer!

Come join us if interested!

- Interdisciplinary research
  - Data mining, natural language processing, machine/deep learning, bioinformatics, programming, literature study, etc.
- Deal with real problems and get your hands dirty
- Always keep an open mind

Xifeng Yan, UCSB

Huan Sun, OSU

Acknowledgements

- Advisor: Xifeng Yan (UCSB)
- Ahmed H. Awadallah, Patrick Pantel, Michael Gamon, and Scott Yih (Microsoft Research)
- Brian Sadler, Michelle Vanni, and Sue Kase (Army Research Lab)
- Mudhakar Srivatsa (IBM Research)
- Ph.D. Committee: Ambuj Singh, Amr El Abbadi (UCSB)
- Many other great collaborators!

On Building Natural Language Interface to Data and Services

Contact Info:

Yu Su

Ph.D. Candidate

Department of Computer Science

University of California, Santa Barbara

ysu@cs.ucsb.edu

Thank you! Questions?