Unsupervised Spoken Language Understanding in Dialogue Systems

YUN-NUNG (VIVIAN) CHEN 陳縕儂

CARNEGIE MELLON UNIVERSITY

2015/01/16 @NCTU

HTTP://VIVIANCHEN.IDV.TW

Outline

Introduction

Unsupervised Slot Induction [Chen et al., ASRU'13 & Chen et al., SLT'14]

Unsupervised Domain Exploration [Chen and Rudnicky, SLT'14]

Unsupervised Relation Detection [Chen et al., SLT'14]

Conclusions & Future Work

Outline

Introduction

Unsupervised Slot Induction [Chen et al., ASRU'13 & Chen et al., SLT'14]

Unsupervised Domain Exploration [Chen and Rudnicky, SLT'14]

Unsupervised Relation Detection [Chen et al., SLT'14]

Conclusions & Future Work

Spoken Language Understanding (SLU)

SLU in dialogue systems

- SLU maps *natural language inputs* to *semantic forms*

  "I would like to go to NCTU on Friday."

  location: NCTU date: Friday

- Semantic frames, slots, and values
  - often manually defined by domain experts or developers.

What are the problems?

Problems with Predefined Information

Generalization: may not generalize to real-world users.

Bias propagation: can bias subsequent data collection and annotation.

Maintenance: when new data comes in, developers need to start a new round of annotation to analyze the data and update the grammar.

Efficiency: time consuming, and high costs.

Can we automatically induce semantic information w/o annotations?

Outline

Introduction

Unsupervised Slot Induction [Chen et al., ASRU'13 & Chen et al., SLT'14]

Unsupervised Domain Exploration [Chen and Rudnicky, SLT'14]

Unsupervised Relation Detection [Chen et al., SLT'14]

Conclusions & Future Work

Unsupervised Slot Induction

Motivation

- Spoken dialogue systems (SDS) require predefined semantic slots to parse users' input into semantic representations
- *Frame semantics theory* provides generic semantics
- *Distributional semantics* capture contextual latent semantics

Probabilistic Frame-Semantic Parsing

FrameNet [Baker et al., 1998]

- a linguistically-principled semantic resource, based on the frame-semantics theory.
- "low fat milk" → "milk" evokes the "food" frame;
  "low fat" fills the descriptor frame element
  - **Frame** (*food*): contains words referring to items of food.
  - **Frame Element**: a descriptor indicates the characteristic of food.

SEMAFOR [Das et al., 2010; 2013]

- a state-of-the-art frame-semantics parser, trained on manually annotated FrameNet sentences

Step 1: Frame-Semantic Parsing for ASR outputs

Good!

Bad! can i have a cheap restaurant *Good!*

Frame: expensiveness
FT LU: cheap

Frame: capability
FT LU: can FE LU: i

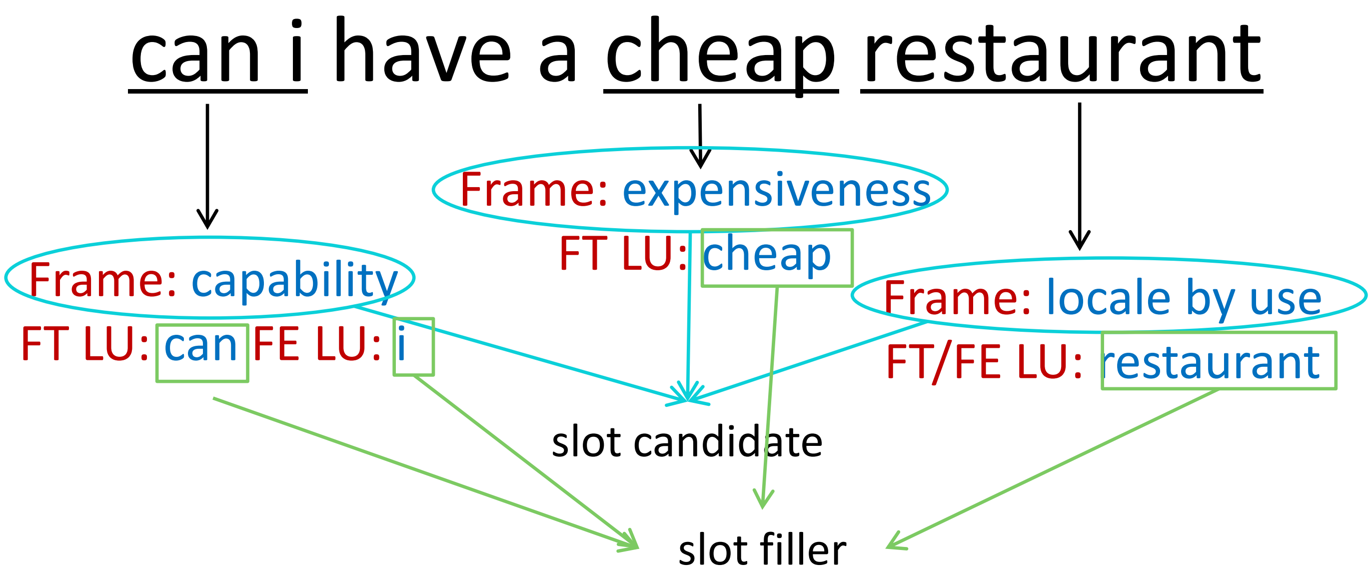
Frame: locale by use
FT/FE LU: restaurant

Task: adapting ***generic*** frames to ***task-specific*** settings for SDSs

Step 2: Slot Ranking Model

Main Idea

- Ranking domain-specific concepts higher than generic semantic concepts



Step 2: Slot Ranking Model

Rank the slot candidates by integrating two scores

$$w(s_i) = (1-\alpha)\log f(s_i) + \alpha \cdot \log h(s_i)$$

the frequency of the slot candidate in the SEMAFOR-parsed corpus

the coherence of slot fillers

slots with higher frequency may be more important

domain-specific concepts should focus on fewer topics and be similar to each other

slot: quantity

one
a
three all

lower coherence in topic space

slot: expensiveness

cheap inexpensive
expensive

higher coherence in topic space

Step 2: Slot Ranking Model

Measure coherence by pair-wised similarity of slot fillers

- For each slot candidate s_i

$$V(\boxed{s_i}) = \{\boxed{x_a}, x_b, \ldots\}$$

slot candidate: expensiveness

corresponding slot filler: "cheap", "not expensive"

$$h(s_i) = \frac{\sum_{x_a, x_b \in V(s_i), x_a \neq x_b} \mathrm{Sim}(x_a, x_b)}{|V(s_i)|^2}$$

The slot with higher $h(s_i)$ usually focuses on fewer topics, which are more specific, which is preferable for slots of SDS.

Step 2: Slot Ranking Model

How to define the vector for each slot filler?

- Run clustering and then build vectors based on clustering results
  - K-means, spectral clustering, etc.
- Use distributional semantics to transfer words into vectors
  - LSA, PLSA, neural word embeddings (word2vec)

Experiments for Slot Induction

Dataset

- Cambridge University SLU corpus [Henderson, 2012]
  - Restaurant recommendation in an in-car setting in Cambridge
    - WER = 37%
    - vocabulary size = 1868
    - 2,166 dialogues
    - 15,453 utterances
    - dialogue slot: addr, area, food, name, phone, postcode, price range, task, type

Induced slots	Reference slot
locale by use, building	type
food, origin	food
speak on topic	addr
seeking, desiring, locating	task
commerce scenario, expensiveness, range	price range
part orientational, direction, locale, part inner outer	area
contacting	phone
sending	postcode

The mapping table between induced and reference slots

Experiments for Slot Induction

- **Slot Induction Evaluation**: MAP of the slot ranking model to measure the quality of induced slots via the mapping table
- **Slot Filling Evaluation**: MAP-F-H/S: weight the MAP score with F-measure of two slot filler lists

Approach				ASR		
				MAP	MAP-F-H	MAP-F-S
Frame Sem	(a)	Frequency		67.61	26.96	27.29
	(b)	K-Means		67.38	27.38	27.99
	(c)	Spectral Clustering		68.06	30.52	28.40
Frame Sem + Dist Sem	(d)	Google News	RepSim	72.71	31.14	31.44
	(e)		NeiSim	73.35	**31.44**	**31.81**
	(f)	Freebase	RepSim	71.48	29.81	30.37
	(g)		NeiSim	73.02	30.89	30.72
	(h)	(d) + (e) + (f) + (g)		**76.22**	30.17	30.53

Experiments for Slot Induction

Approach				ASR		
				MAP	MAP-F-H	MAP-F-S
Frame Sem	(a)	Frequency		67.61	26.96	27.29
	(b)	K-Means		67.38	27.38	27.99
	(c)	Spectral Clustering		68.06	30.52	28.40
Frame Sem + Dist Sem	(d)	Google News	RepSim	72.71	31.14	31.44
	(e)		NeiSim	73.35	**31.44**	**31.81**
	(f)	Freebase	RepSim	71.48	29.81	30.37
	(g)		NeiSim	73.02	30.89	30.72
	(h)	(d) + (e) + (f) + (g)		**76.22**	30.17	30.53

Adding distributional information outperforms our baselines

Experiments for Slot Induction

Approach				ASR		
				MAP	MAP-F-H	MAP-F-S
Frame Sem	(a)	Frequency		67.61	26.96	27.29
	(b)	K-Means		67.38	27.38	27.99
	(c)	Spectral Clustering		68.06	30.52	28.40
Frame Sem + Dist Sem	(d)	Google News	RepSim	72.71	31.14	31.44
	(e)		NeiSim	73.35	**31.44**	**31.81**
	(f)	Freebase	RepSim	71.48	29.81	30.37
	(g)		NeiSim	73.02	30.89	30.72
	(h)	(d) + (e) + (f) + (g)		**76.22**	30.17	30.53

Combining two datasets to integrate the coverage of Google and precision of Freebase can rank correct slots higher and performs the best MAP scores

Outline

Introduction

Unsupervised Slot Induction [Chen et al., ASRU'13 & Chen et al., SLT'14] **Question?**

Unsupervised Domain Exploration [Chen and Rudnicky, SLT'14]

Unsupervised Relation Detection [Chen et al., SLT'14]

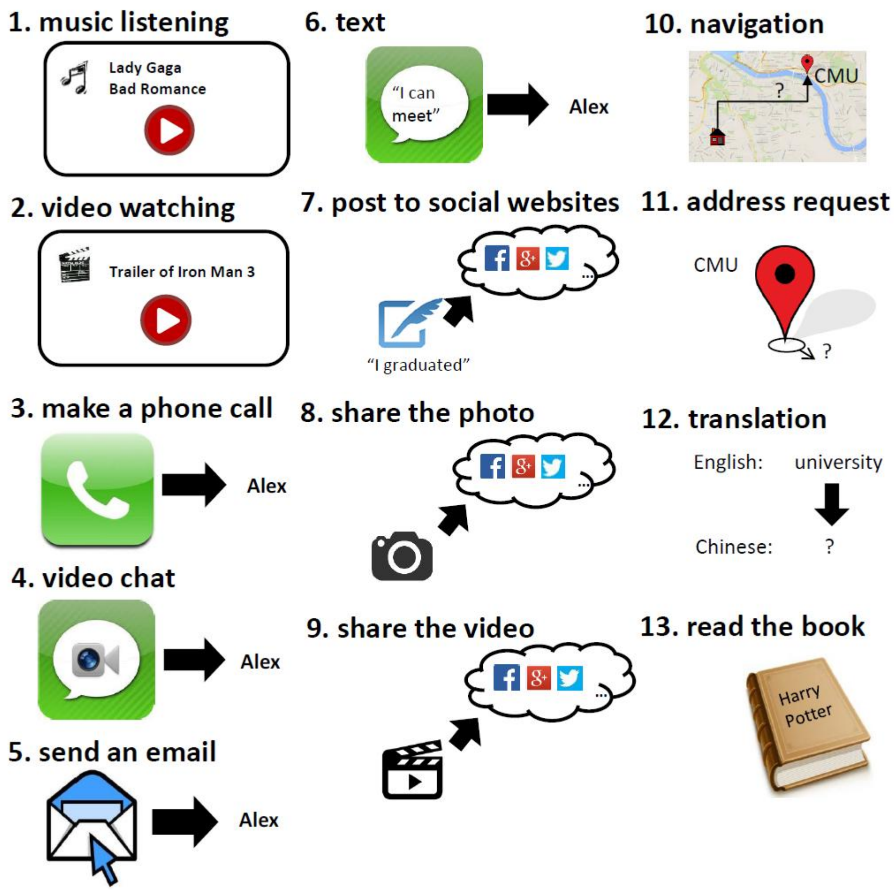
Conclusions & Future Work

Unsupervised Domain Exploration

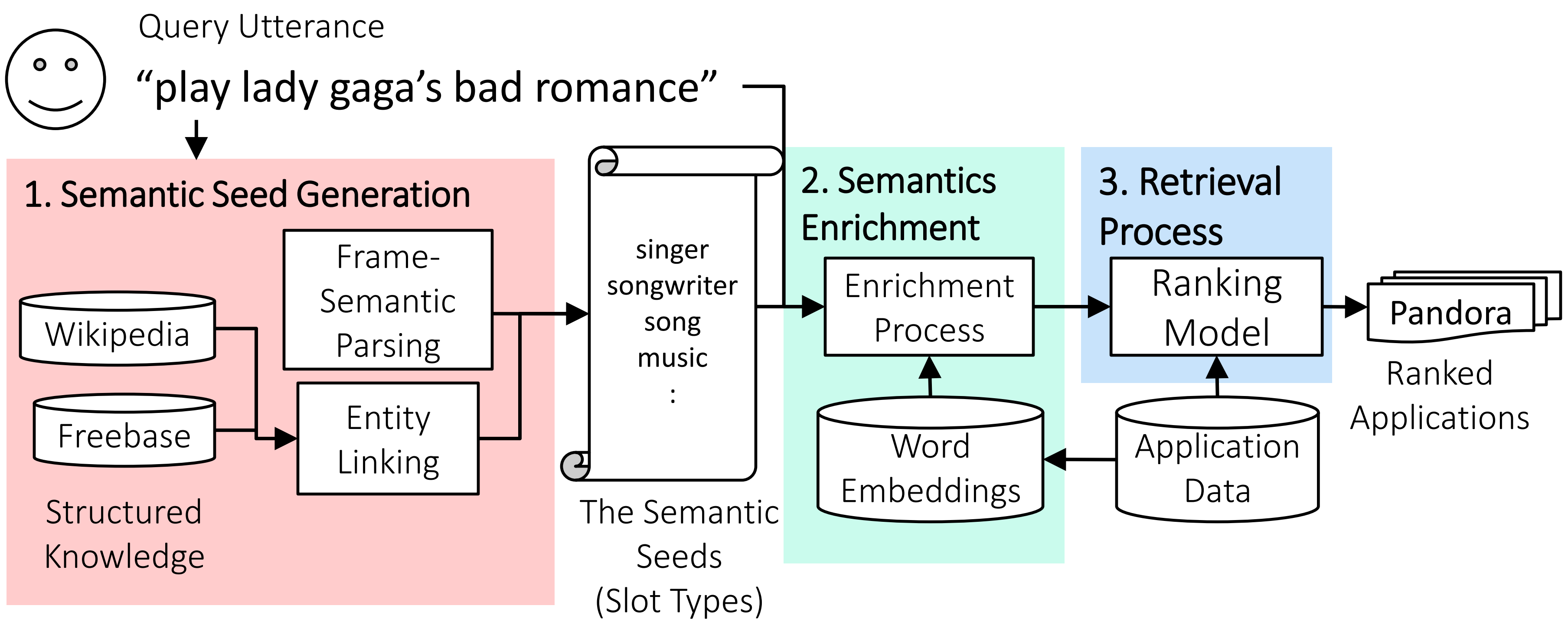
Target: given conversation interaction with SDS, predicting which application the user wants to launch

Approach:

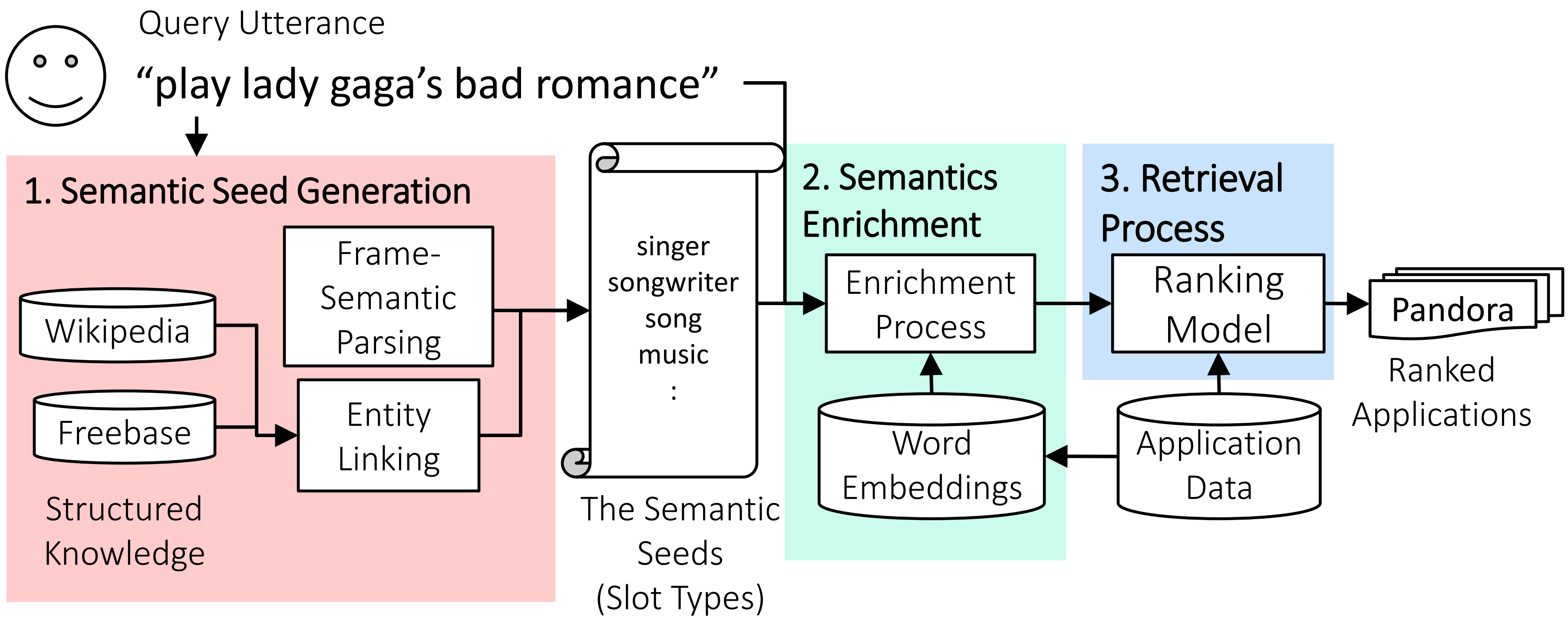
- Step 1: enriching the semantics using word embeddings
- Step 2: using the descriptions of applications as a retrieval cue to find relevant applications



Proposed Framework



Proposed Framework



Semantic Seed Generation

- Main idea: Slot types help imply semantic meaning of the utterance for expanding domain knowledge.
- Frame Type of Semantic Parsing

Q: compose an email to alex

Frame: text creation　　Frame: contacting

FT LU: compose FE LU: an email　　FT LU: email

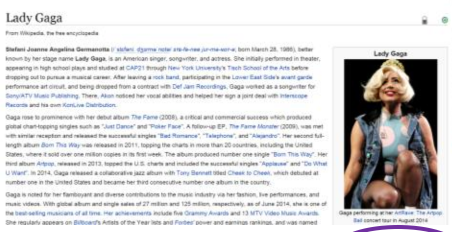
$S_{frm}(Q)$: frame-based semantic seeds

Semantic parsing performs well on a generic domain, and cannot recognize domain-specific named entities.

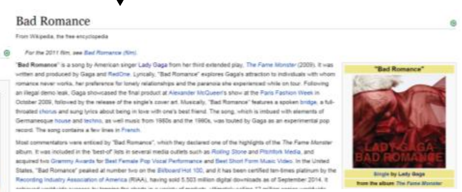
Semantic Seed Generation

- Main idea: Slot types help imply semantic meaning of the utterance for expanding domain knowledge.
- Entity Type from Linked Structured Knowledge
  - Wikipedia Page Linking
  - Freebase List Linking

Q: play lady gaga's bad romance


Lady Gaga

From Wikipedia, the free encyclopedia


Bad Romance

From Wikipedia, the free encyclopedia

… is a song by American singer …

… is an American singer, songwriter, and actress.

$S_{wk}(Q)$: wikipedia-based semantic seeds

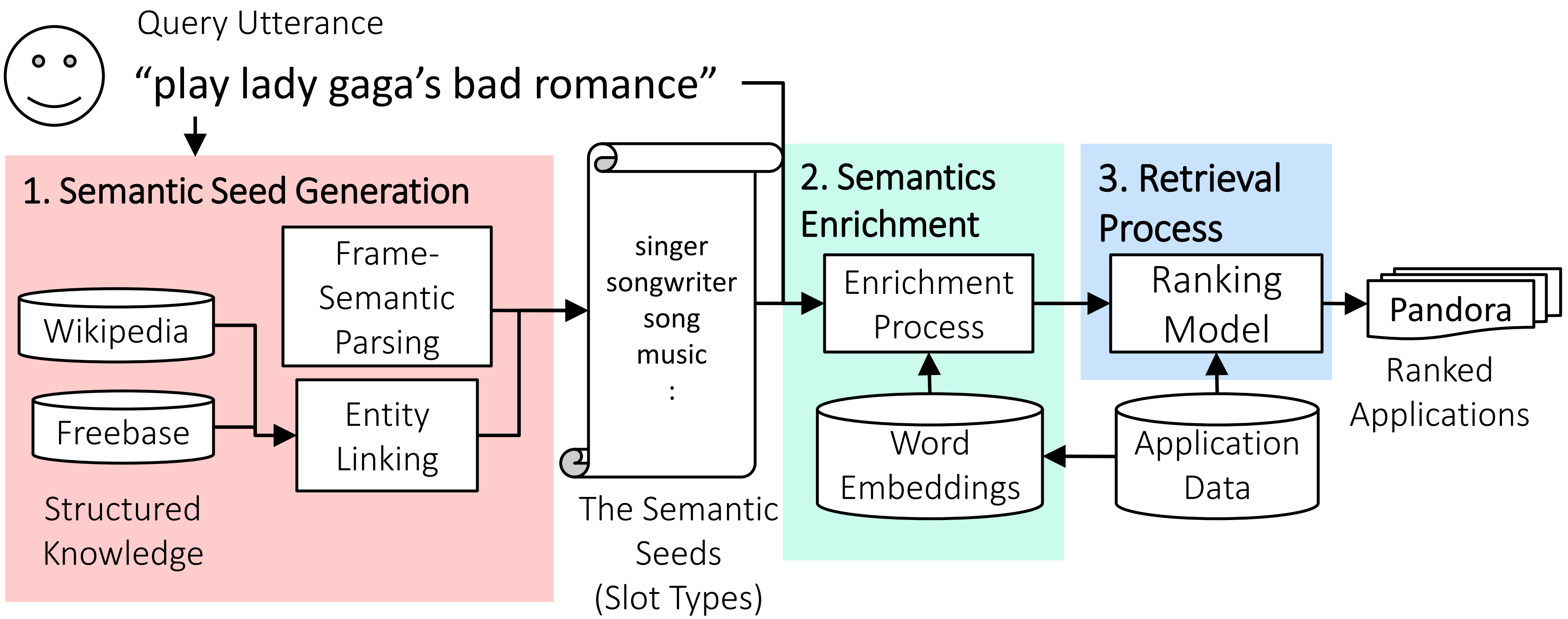
Q: play lady gaga's bad romance

celebrity
composition
:

composition
canonical version
musical recording
:

$S_{fb}(Q)$: freebase-based semantic seeds

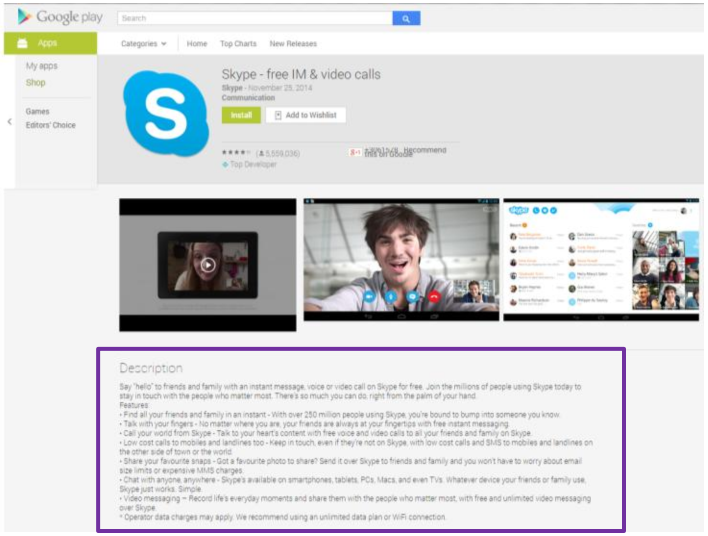
Proposed Framework



Semantic Enrichment

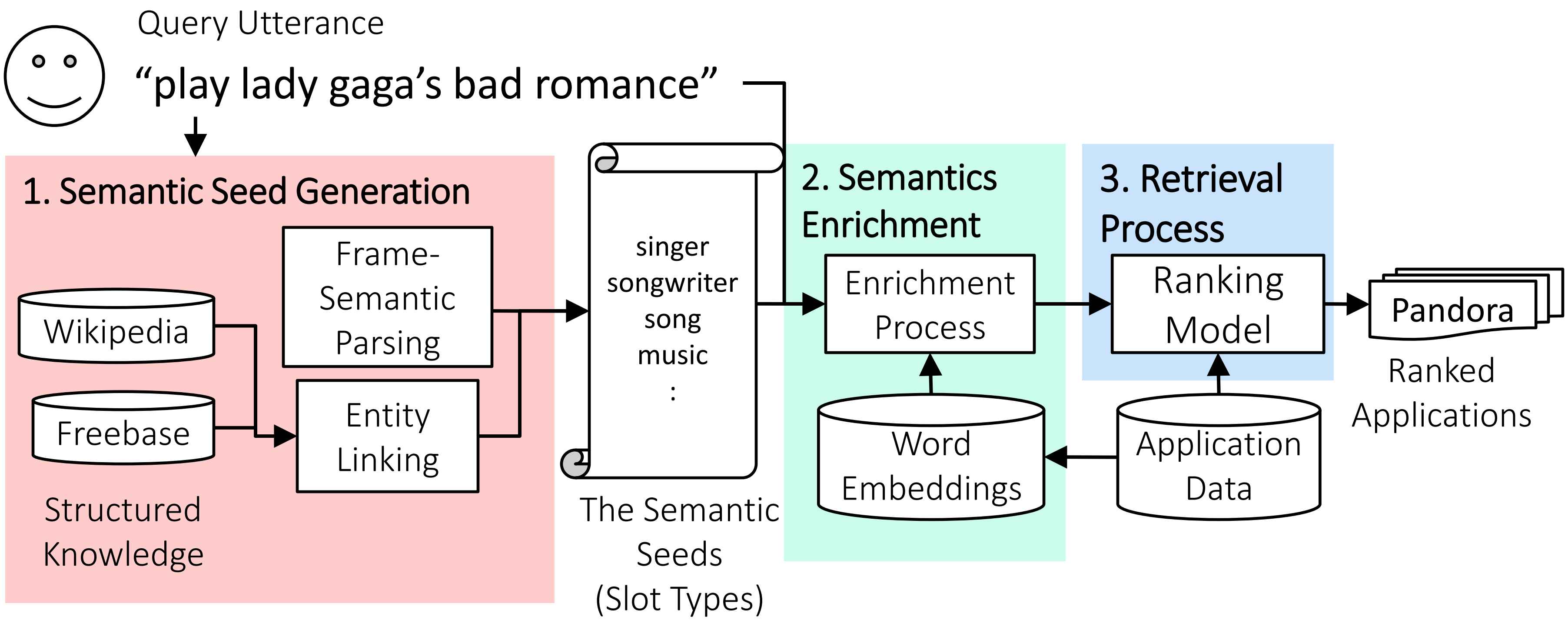
- Main idea: Utilizing distributed word embeddings to obtain the semantically related knowledge of each word.

1) Modeling word embeddings by the application vender descriptions.
2) Extracting the most related words by trained word embeddings for each word. (ex. “text” → “message”, “msg”)



Words with higher similarity suggest that they are often occurs with common contexts in the embedding training data.

Proposed Framework



Retrieval Process

- Main idea: Retrieving the applications that are more likely to support users' requests.
  - Query Reformulation (Q')
    - Embedding-Enriched Query: integrates similar words to all words in Q
    - Type-Embedding-Enriched Query: additionally adds similar words to semantic seeds $S(Q)$
  - Ranking Model

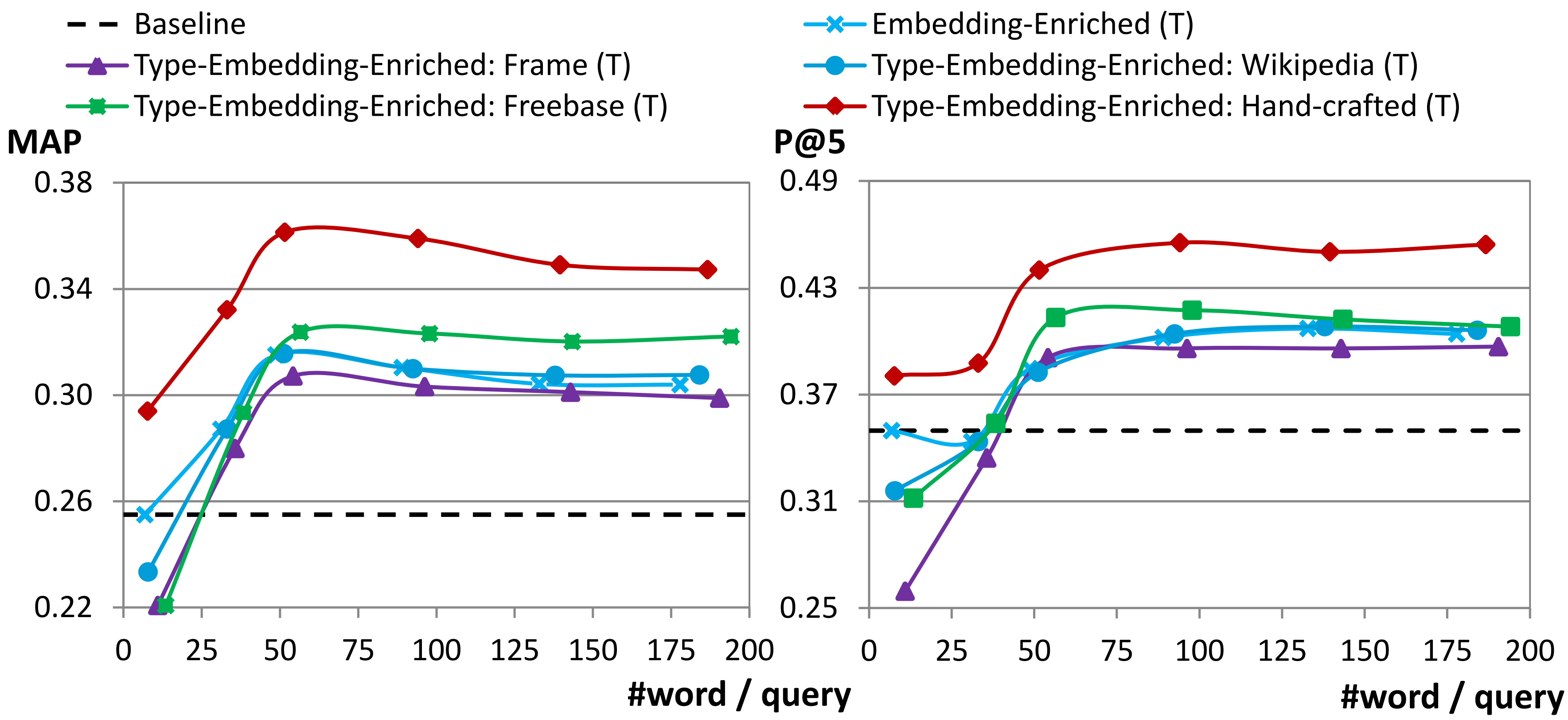
$$\boxed{P(Q \mid A)} = \frac{1}{|Q'|} \sum_{x \in Q'} \log \boxed{P(x \mid A)}$$

probability that user speaks Q to make the request for launching the application A

probability that word x occurs in the application

The application with higher $P(Q \mid A)$ is more likely to be able to support the user desired functions.

Results



Baseline
Type-Embedding-Enriched: Frame (T)
Type-Embedding-Enriched: Freebase (T)
Embedding-Enriched (T)
Type-Embedding-Enriched: Wikipedia (T)
Type-Embedding-Enriched: Hand-crafted (T)
MAP
0.38
0.34
0.30
0.26
0.22
0 25 50 75 100 125 150 175 200
#word / query
P@5
0.49
0.43
0.37
0.31
0.25
0 25 50 75 100 125 150 175 200
#word / query

Overall Results

Tune the thresholds by develop set

Approach		ASR	
		MAP	P@5
Original Query		25.50	34.97
Embedding-Enriched		30.42	40.72
Type-Embed.-Enriched	Frame	30.11	39.59
	Wikipedia	30.74	40.82
	Freebase	**32.02**	**41.23**
	Hand-Craft	34.91	45.03

- Enriching semantics improves performance by involving domain-specific knowledge.
- Freebase results are better than the embedding-enriched method, showing that we can effectively and efficiently expand domain-specific knowledge by types of slots from Freebase.
- Hand-crafted mapping shows that the correct types of slots offer better understanding and tells the room of improvement.

Outline

Introduction

Unsupervised Slot Induction [Chen et al., ASRU'13 & Chen et al., SLT'14]

Unsupervised Domain Exploration [Chen and Rudnicky, SLT'14] **Question?**

Unsupervised Relation Detection [Chen et al., SLT'14]

Conclusions & Future Work

Unsupervised Relation Detection

Spoken Language Understanding (SLU): convert ASR outputs into pre-defined semantic output format

"when was james cameron's avatar released"

Intent: FIND_RELEASE_DATE
Slot-Val: MOVIE_NAME="avatar", DIRECTOR_NAME="james cameron"

Relation: semantic interpretation of input utterances

- movie.release_date, movie.name, movie.directed_by, director.name

Unsupervised SLU: utilize external knowledge to help relation detection without labelled data

Semantic Knowledge Graph

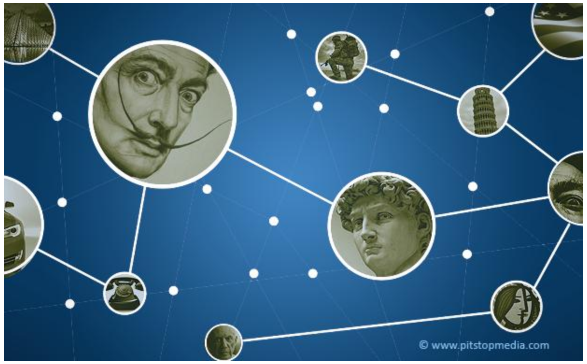
Priors for SLU

What are knowledge graphs?

- Graphs with
  - strongly typed and uniquely identified entities (nodes)
  - facts/literals connected by relations (edge)

Examples:

- Satori, Google KG, Facebook Open Graph, Freebase



How large?

- > 500M entities, >1.5B relations, > 5B facts

How broad?

- Wikipedia-breadth: "American Football" ←→ "Zoos"

- Slides of Larry Heck, Dilek Hakkani-Tur, and Gokhan Tur, Leveraging Knowledge Graphs for Web-Scale Unsupervised Semantic Parsing, in *Proceedings of Interspeech*, 2013.

Semantic Interpretation via Relations

Two Examples

- differentiate two examples by including the **originating node types** in the relation

User Utterance:

find movies produced by james cameron

SPARQL Query (simplified):

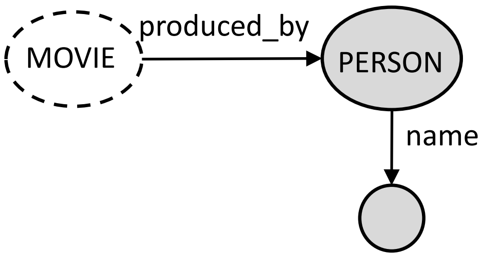
SELECT ?movie {?movie. ?movie.produced_by?producer. ?producer.name"James Cameron".}

Logical Form:

λx. ∃y. movie.produced_by(x, y) ∧ person.name(y, z) ∧ z=“James Cameron”

Relation:

movie.produced_by producer.name



User Utterance:

who produced avatar

SPARQL Query (simplified):

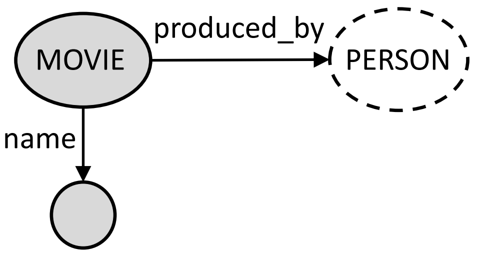
SELECT ?producer {?movie.name"Avatar“. ?movie.produced_by?producer.}

Logical Form:

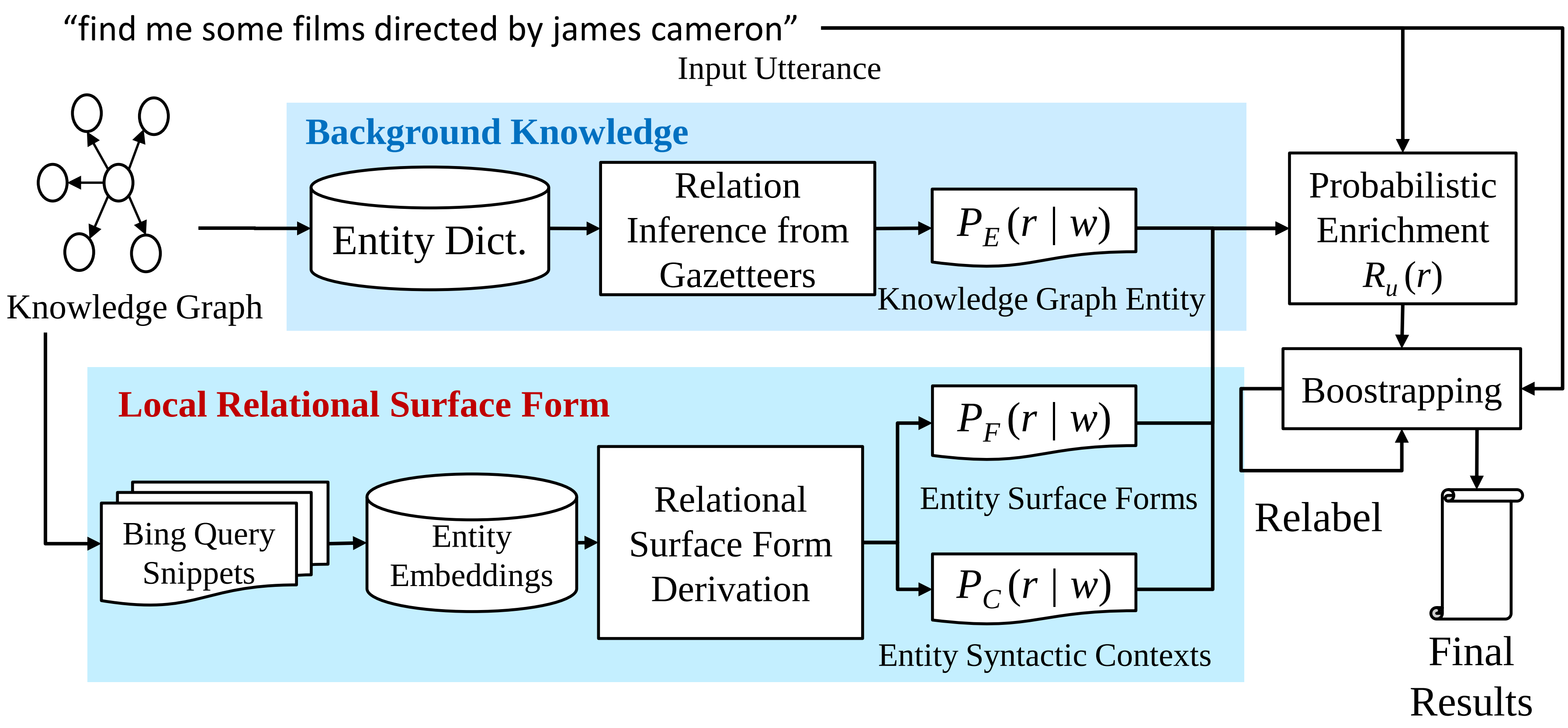
λy. ∃x. movie.produced_by(x, y) ∧ movie.name(x, z) ∧ z=“Avatar”

Relation:

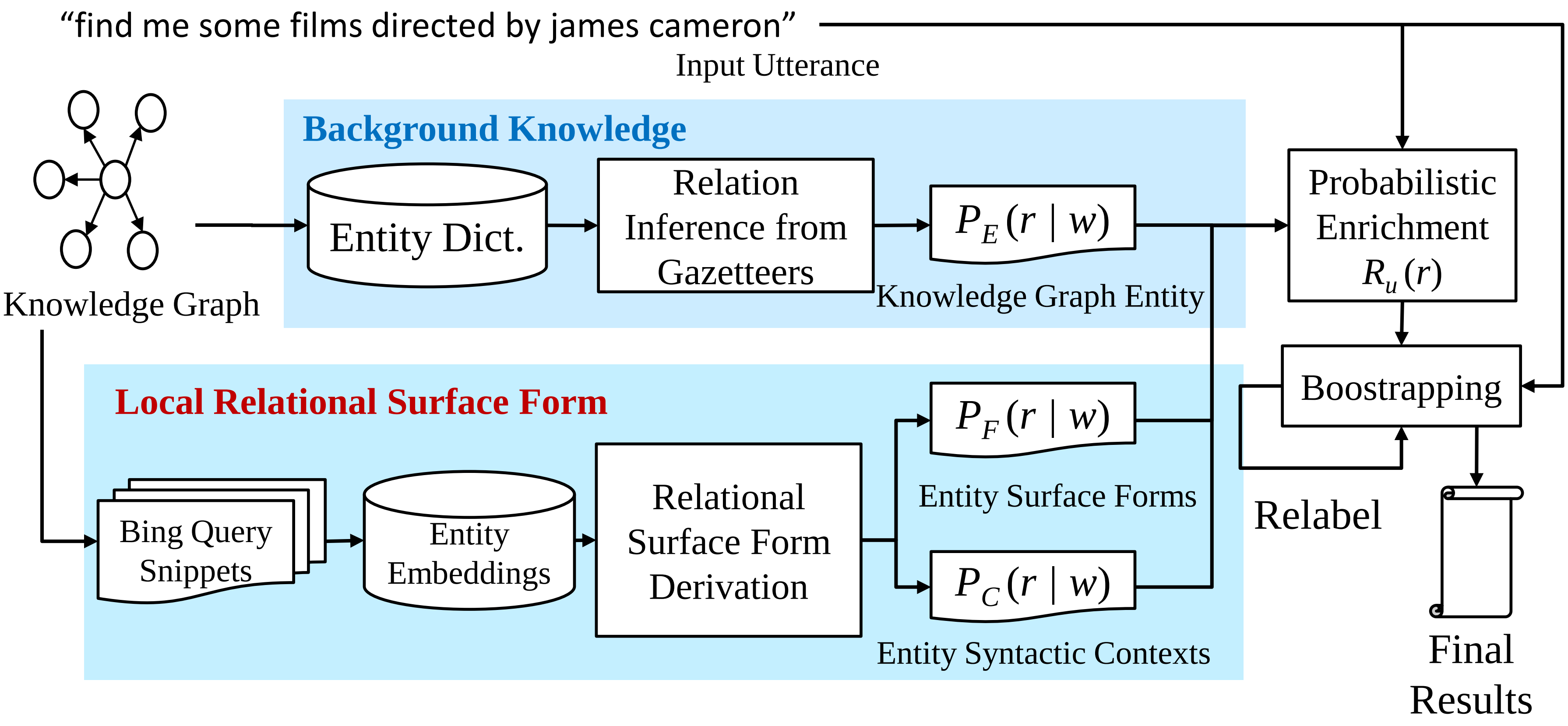
movie.name movie.produced_by



Proposed Framework



Proposed Framework



Relation Inference from Gazetteers

Gazetteers (entity lists)

"james cameron"

#movies James Cameron directed

$$P_E(t_i \mid w) = \frac{\boxed{C(w, t_i)}}{\sum_{t_k \in T(w)} C(w, t_k)}$$

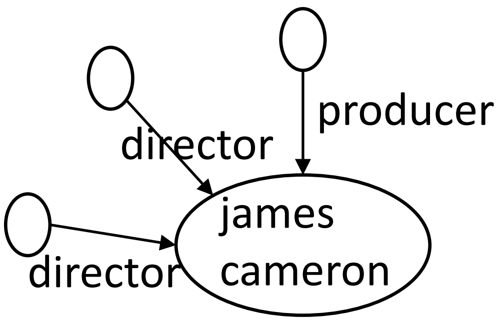
director

director
producer
:

$$P_E(r_i \mid w) = P_E(t_i \mid w)$$

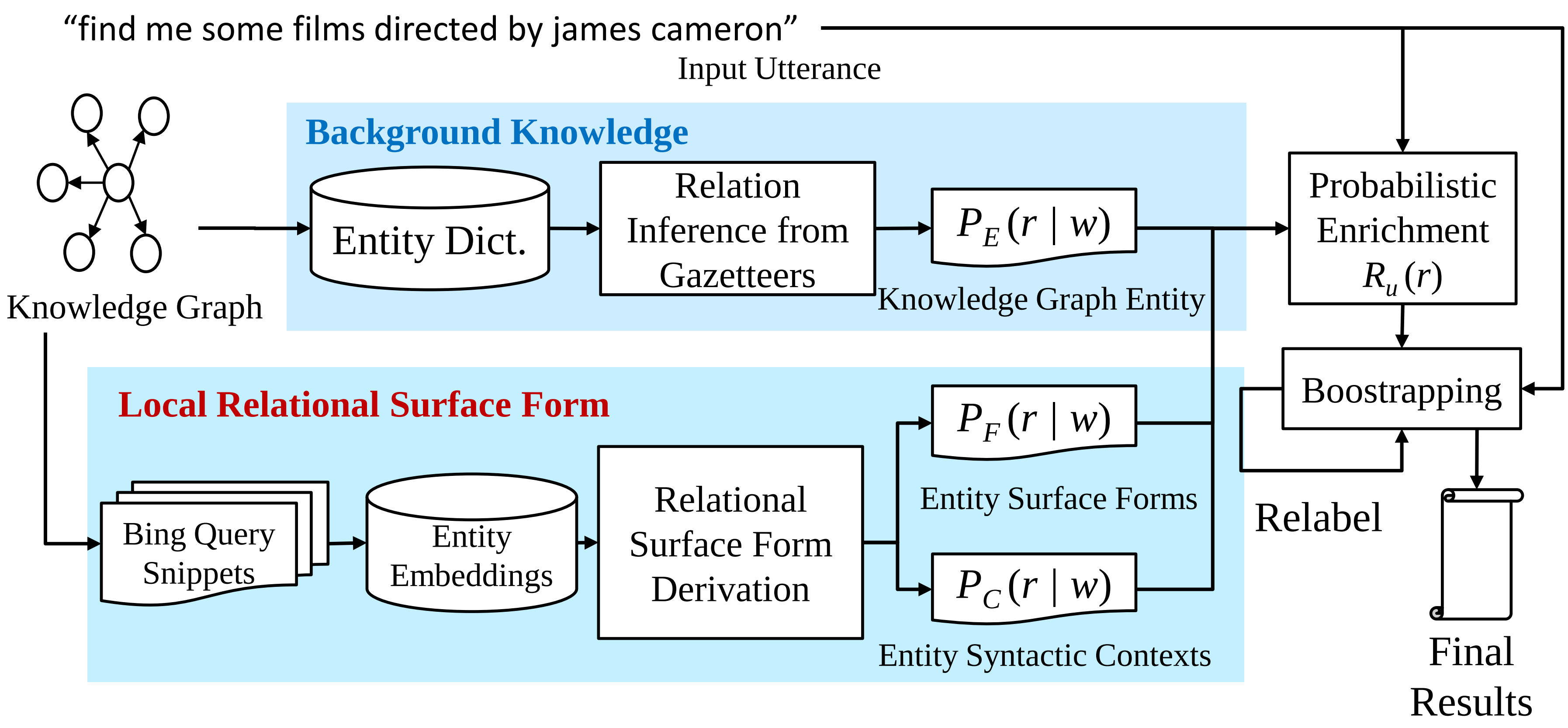
movie.directed_by
director.name

director



- Dilek Hakkani-Tur, Asli Celikyilmaz, Larry Heck, and Gokhan Tur, Probabilistic enrichment of knowledge graph entities for relation detection in conversational understanding, in *Proceedings of Interspeech,* 2014.

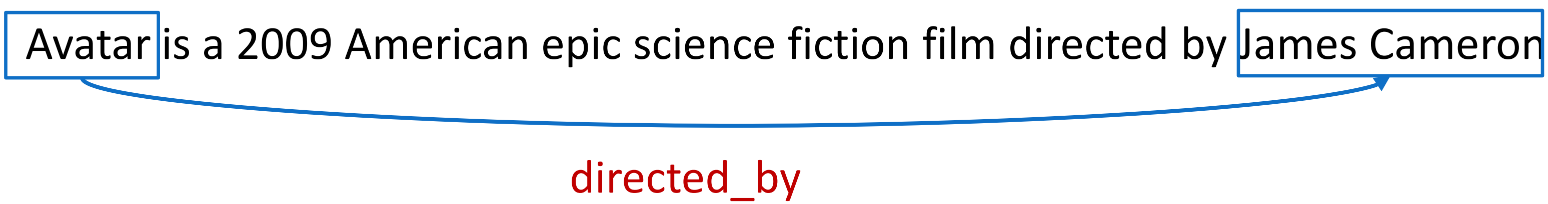
Proposed Framework



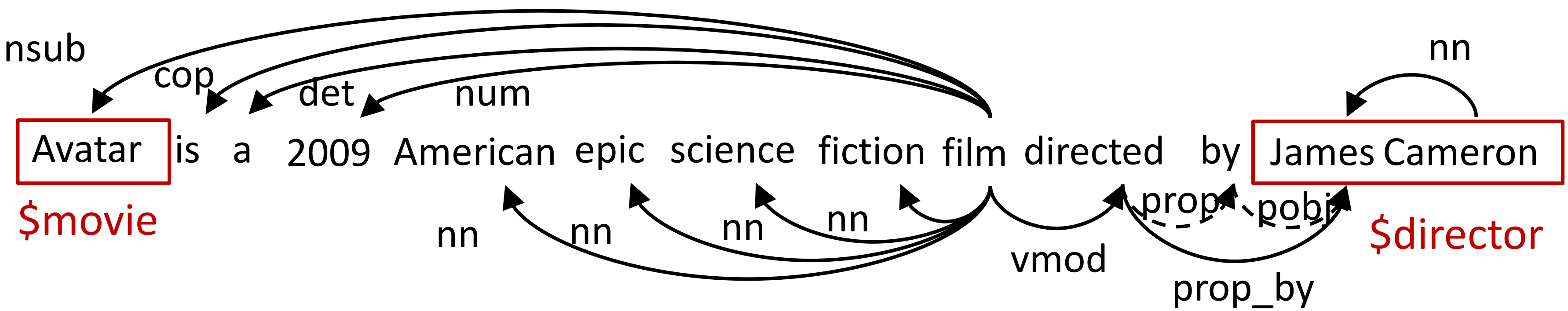
Relational Surface Form Derivation

Web Resource Mining

Bing query snippets including entity pairs connected with specific relations in KG



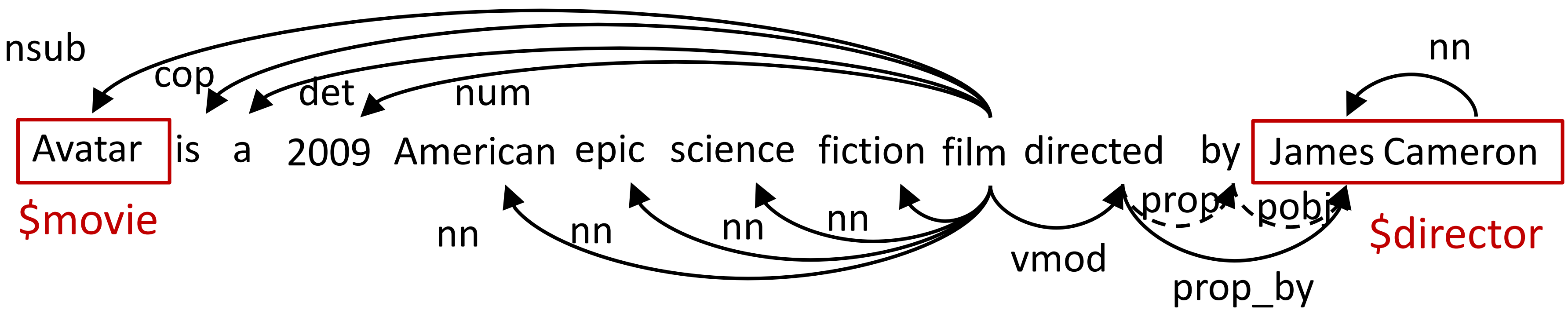
Dependency Parsing



Relational Surface Form Derivation

Dependency-Based Entity Embeddings

1) Word & Context Extraction



Word	Contexts
$movie	film/nsub^{-1}
is	film/cop^{-1}
a	film/det^{-1}
2009	film/num^{-1}
american, epic, science, fiction	film/nn^{-1}

Word	Contexts
film	film/nsub, is/cop, a/det, 2009/num, american/nn, epic/nn, science/nn, fiction/nn, directed/vmod
directed	$director/prep_by
$director	directed/prep_by^{-1}

Relational Surface Form Derivation

Dependency-Based Entity Embeddings

2) Training Process
   - Each word w is associated with a vector v_w and each context c is represented as a vector v_c
   - Learn vector representations for both words and contexts such that the dot product $v_w \cdot v_c$ associated with good word-context pairs belonging to the training data D is maximized
   - Objective function: $$\arg\max_{v_w, v_c} \sum_{(w,c) \in D} \log \frac{1}{1 + \exp(-v_c \cdot v_w)}$$

Word	Contexts
$movie	film/nsub^{-1}
is	film/cop^{-1}
a	film/det^{-1}
2009	film/num^{-1}
american, epic, science, fiction	film/nn^{-1}

Word	Contexts
film	film/nsub, is/cop, a/det, 2009/num, american/nn, epic/nn, science/nn, fiction/nn, directed/vmod
directed	$director/prep_by
$director	directed/prep_by^{-1}

Relational Surface Form Derivation

Surface Form Derivation

Entity Surface Forms

- learn the surface forms corresponding to entities

$char, $director, etc.

$$S_i^F(w_j) = \boxed{\frac{\mathrm{sim}(w_j, e_i)}{\sum_{e_k \in E} \mathrm{sim}(w_j, e_k)}}$$

$$P^F(r_i \mid w_j)$$

based on word vector v_w

$char: "character", "role", "who"
$director: "director", "filmmaker"
$genre: "action", "fiction"

→ with similar contexts

Entity Syntactic Contexts

- learn the important contexts of entities

$$S_i^C(w_j) = \boxed{\frac{\mathrm{sim}(\hat{w}_j, e_i)}{\sum_{e_k \in E} \mathrm{sim}(\hat{w}_j, e_k)}}$$

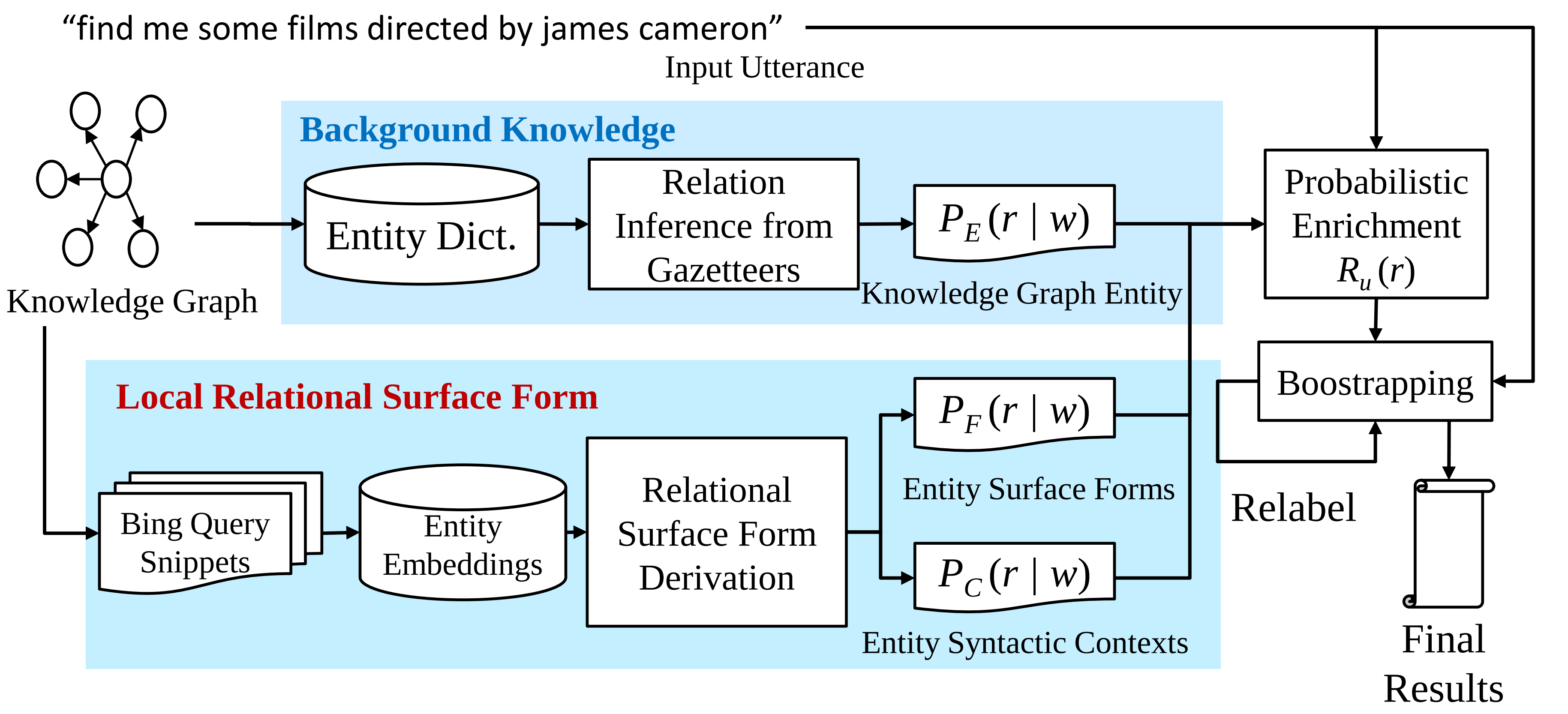
$$P^C(r_i \mid w_j)$$

based on context vector v_c

$char: "played"
$director: "directed"

→ frequently occurring together

Proposed Framework



Probabilistic Enrichment

Integrate relations from

- Prior knowledge $P_E(r \mid w)$
- Entity surface forms $P_F(r \mid w)$
- Entity syntactic contexts $P_C(r \mid w)$

r	actor	produced_by	location
$P_E(r \mid w)$	0.7	0.3	0
$P_F(r \mid w)$	0.4	0	0.6
$P_C(r \mid w)$	0	0	0
Unweighted $R_w(r)$	1	1	1
Weighted $R_w(r)$	0.7	0.3	0.6
Highest Weighted $R_w(r)$	0.7	0	0.6

Integrated Relations for Words by

- **Unweighted**: combine all relations with binary values
- **Weighted**: combine all relations and keep the highest weights of relations
- **Highest Weighted**: combine the most possible relation of each word

Integrated Relations for Utterances by

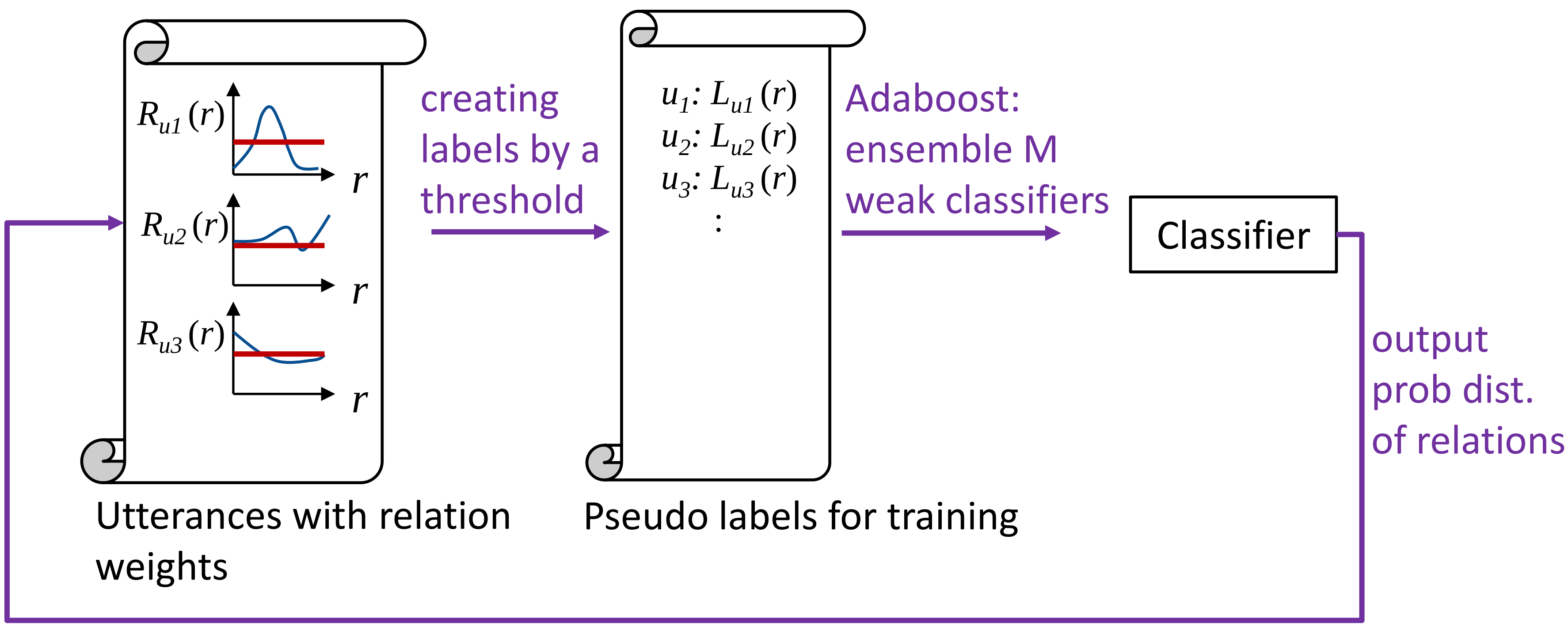
$$R_u(r_i) = \max_{w \in u} R_w(r_i)$$

- Dilek Hakkani-Tur, Asli Celikyilmaz, Larry Heck, and Gokhan Tur, Probabilistic enrichment of knowledge graph entities for relation detection in conversational understanding, in *Proceedings of Interspeech,* 2014.

Boostrapping

Unsupervised Self-Training

Training a multi-label multi-class classifier estimating relations given an utterance



Experiments of Relation Detection
Dataset

Knowledge Base: Freebase

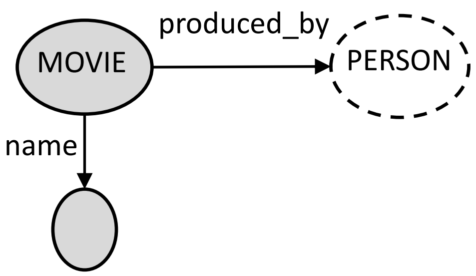
- 670K entities
- 78 entity types (movie names, actors, etc)

Relation Detection Data

- Crowd-sourced utterances
- Manually annotated with SPARQL queries → relations

Query Statistics	Dev	Test
% entity only	8.9%	10.7%
% rel only w/ specified movie names	<u>27.1%</u>	<u>27.5%</u>
% rel only w/ specified other names	39.8%	39.6%
% more complicated relations	15.4%	14.7%
% not covered	8.8%	7.6%
#utterances	3338	1084

User Utterance:
who produced avatar
Relation:
movie.name
movie.produced_by



Experiments of Relation Detection
All performance

Evaluation Metric: micro F-measure (%)

Approach	Unweighted		Weighted		Highest Weighted	
	Ori	Boostrap	Ori	Boostrap	Ori	Boostrap
Gazetteer	35.21	36.91	37.93	40.10	36.08	38.89
Gazetteer + Weakly Supervised	25.07	37.39	39.04	39.07	39.40	39.98
Gazetteer + Entity Surface Form (Reg)	34.23	34.91	36.57	38.13	34.69	37.16

Baseline

Experiments of Relation Detection
All performance

Evaluation Metric: micro F-measure (%)

Approach	Unweighted		Weighted		Highest Weighted	
	Ori	Boostrap	Ori	Boostrap	Ori	Boostrap
Gazetteer	35.21	36.91	37.93	40.10	36.08	38.89
Gazetteer + Weakly Supervised	25.07	37.39	39.04	39.07	39.40	39.98
Gazetteer + Entity Surface Form (Reg)	34.23	34.91	36.57	38.13	34.69	37.16
Gazetteer + Entity Surface Form (Dep)	37.44	38.37	41.01	41.10	39.19	42.74

Baseline

Words derived by dependency embeddings can successfully capture the surface forms of entity tags, while words derived by regular embeddings cannot.

Experiments of Relation Detection
All performance

Evaluation Metric: micro F-measure (%)

Approach	Unweighted		Weighted		Highest Weighted	
	Ori	Boostrap	Ori	Boostrap	Ori	Boostrap
Gazetteer	35.21	36.91	37.93	40.10	36.08	38.89
Gazetteer + Weakly Supervised	25.07	37.39	39.04	39.07	39.40	39.98
Gazetteer + Entity Surface Form (Reg)	34.23	34.91	36.57	38.13	34.69	37.16
Gazetteer + Entity Surface Form (Dep)	37.44	38.37	41.01	41.10	39.19	42.74
Gazetteer + Entity Context	35.31	37.23	38.04	38.88	37.25	38.04

Baseline

Words derived from entity contexts slightly improve performance.

Experiments of Relation Detection

All performance

Evaluation Metric: micro F-measure (%)

	Approach	Unweighted		Weighted		Highest Weighted	
		Ori	Boostrap	Ori	Boostrap	Ori	Boostrap
Baseline	Gazetteer	35.21	36.91	37.93	40.10	36.08	38.89
	Gazetteer + Weakly Supervised	25.07	37.39	39.04	39.07	39.40	39.98
	Gazetteer + Entity Surface Form (Reg)	34.23	34.91	36.57	38.13	34.69	37.16
Proposed	Gazetteer + Entity Surface Form (Dep)	37.44	38.37	41.01	41.10	39.19	42.74
	Gazetteer + Entity Context	35.31	37.23	38.04	38.88	37.25	38.04
	Gazetteer + Entity Surface Form + Context	37.66	38.64	40.29	41.98	40.07	43.34

Combining all approaches performs best, while the major improvement is from derived entity surface forms.

Experiments of Relation Detection

All performance

Evaluation Metric: micro F-measure (%)

	Approach	Unweighted		Weighted		Highest Weighted	
		Ori	Boostrap	Ori	Boostrap	Ori	Boostrap
Baseline	Gazetteer	35.21	36.91	37.93	40.10	36.08	38.89
	Gazetteer + Weakly Supervised	25.07	37.39	39.04	39.07	39.40	39.98
	Gazetteer + Entity Surface Form (Reg)	34.23	34.91	36.57	38.13	34.69	37.16
Proposed	Gazetteer + Entity Surface Form (Dep)	37.44	38.37	41.01	41.10	39.19	42.74
	Gazetteer + Entity Context	35.31	37.23	38.04	38.88	37.25	38.04
	Gazetteer + Entity Surface Form + Context	37.66	38.64	40.29	41.98	40.07	43.34

With the same information, learning surface forms from dependency-based embedding performs better, because there's mismatch between written and spoken language.

Experiments of Relation Detection

All performance

Evaluation Metric: micro F-measure (%)

	Approach	Unweighted		Weighted		Highest Weighted	
		Ori	Boostrap	Ori	Boostrap	Ori	Boostrap
Baseline	Gazetteer	35.21	36.91	37.93	40.10	36.08	38.89
	Gazetteer + Weakly Supervised	25.07	37.39	39.04	39.07	39.40	39.98
	Gazetteer + Entity Surface Form (Reg)	34.23	34.91	36.57	38.13	34.69	37.16
Proposed	Gazetteer + Entity Surface Form (Dep)	37.44	38.37	41.01	41.10	39.19	42.74
	Gazetteer + Entity Context	35.31	37.23	38.04	38.88	37.25	38.04
	Gazetteer + Entity Surface Form + Context	37.66	38.64	40.29	41.98	40.07	43.34

Weighted methods perform better when less features, and highest weighted methods perform better when more features.

Microsoft Research

Experiments of Relation Detection

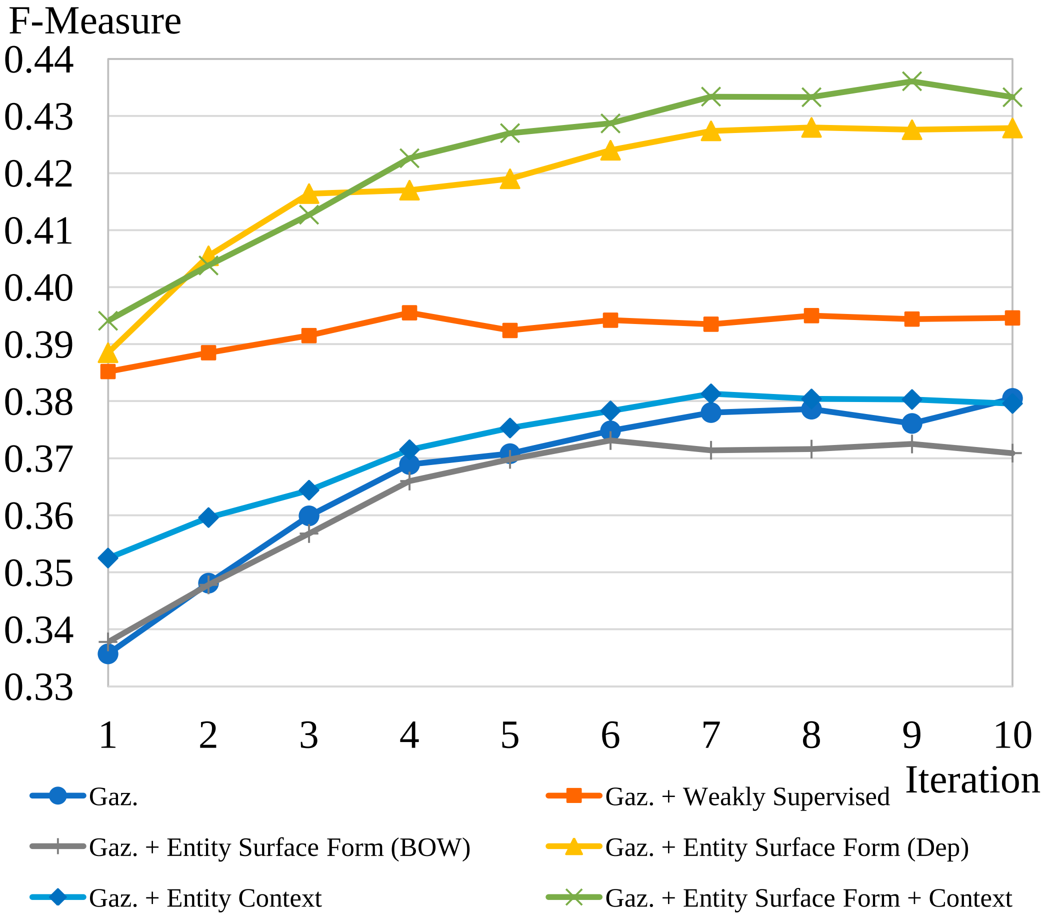
Entity Surface Forms Derived from Dependency Embeddings

The functional similarity carried by dependency-based entity embeddings effectively benefits relation detection task.

Entity Tag	Derived Word
$character	character, role, who, girl, she, he, officier
$director	director, dir, filmmaker
$genre	comedy, drama, fantasy, cartoon, horror, sci
$language	language, spanish, english, german
$producer	producer, filmmaker, screenwriter

Experiments of Relation Detection

Effectiveness of Boosting



- The best result is the combination of all approaches, because probabilities came from different resources can complement each other.
- Only adding entity surface forms performs similarly, showing that the major improvement comes from relational entity surface forms.
- Boosting significantly improves most performance

Outline

Introduction

Unsupervised Slot Induction [Chen et al., ASRU'13 & Chen et al., SLT'14]

Unsupervised Domain Exploration [Chen and Rudnicky, SLT'14]

Unsupervised Relation Detection [Chen et al., SLT'14] **Question?**

Conclusions & Future Work

Conclusions & Future Work

Conclusions

- Unsupervised SLU are more and more popular.
- Using external knowledge helps SLU in different ways.
- Word embeddings is very useful

Future Work

- Fusion of various knowledge resources
  - Different resources help SLU in different ways
- Relation between slots
  - Understanding Inter-slot relations can help develop better SDS
- Active learning
  - In terms of practical and efficiency, manually labeling a small set of samples can boost performance.

Q & A ☺

THANKS FOR YOUR ATTENTIONS!!