FACULTY OF ENGINEERING
AND ARCHITECTURE

INTERNET & DATA LAB

Knowledge Base Population from Text and Graphs

Lucas Sterckx

About Me

- Fourth-year PhD Student
- Research Topics:
  - Relation Extraction
  - Keyphrase Extraction
  - Sequence-to-Sequence Models
- Currently visiting Machine Intelligence Lab under supervision of **prof. Bill Byrne** and **dr. Jason Naradowsky**

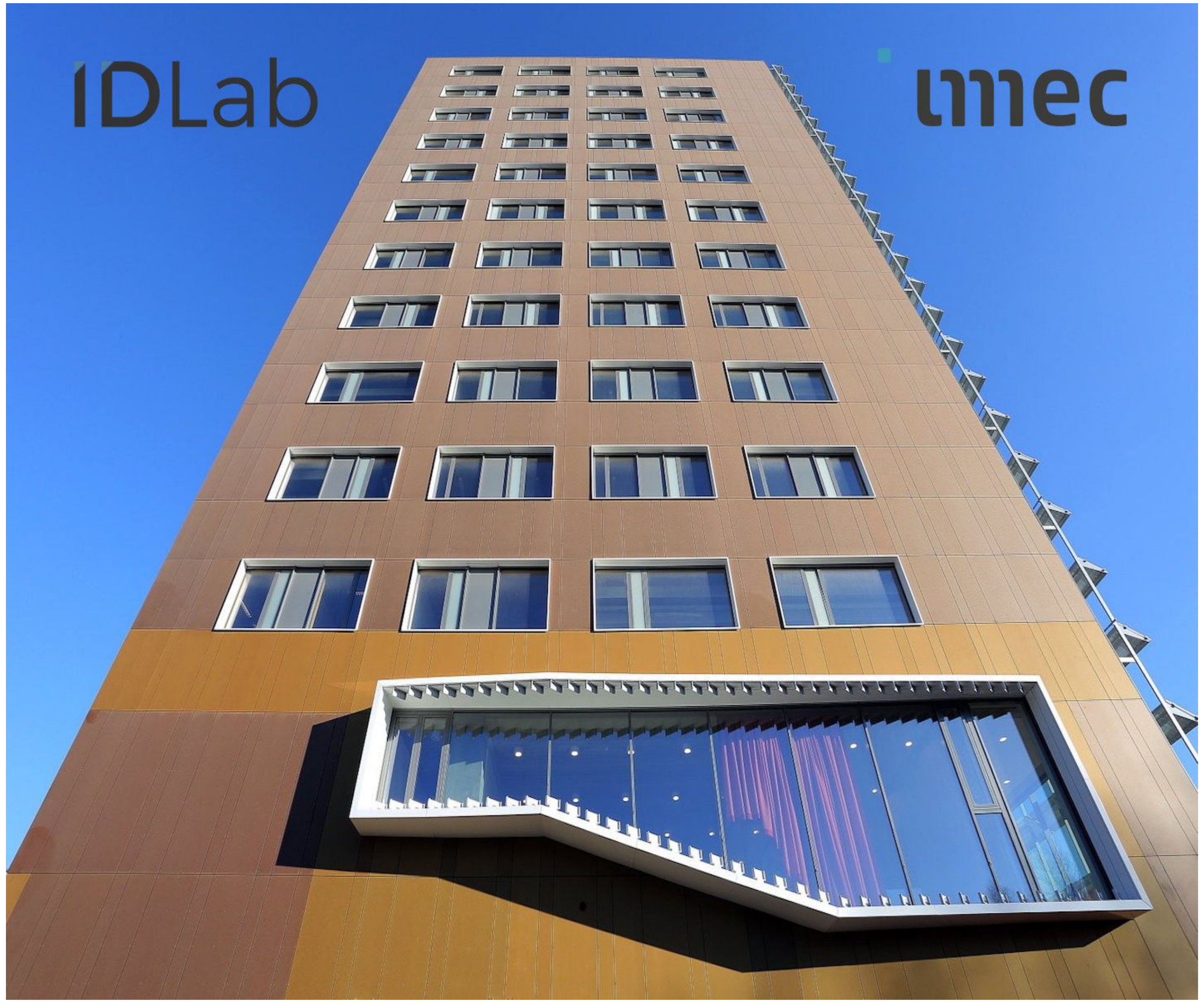
IDLab INTERNET & DATA LAB

GHENT UNIVERSITY

NLP @ Ghent University

IDLab
INTERNET & DATA LAB

- Part of Internet Technology and Data Science Lab
- Initial focus: Information Retrieval
- NLP and Information Extraction since 2013
- Projects with Flanders' major media providers
  - Named Entity Recognition
  - Named Entity Linking
  - Text Classification
  - Keyphrase extraction

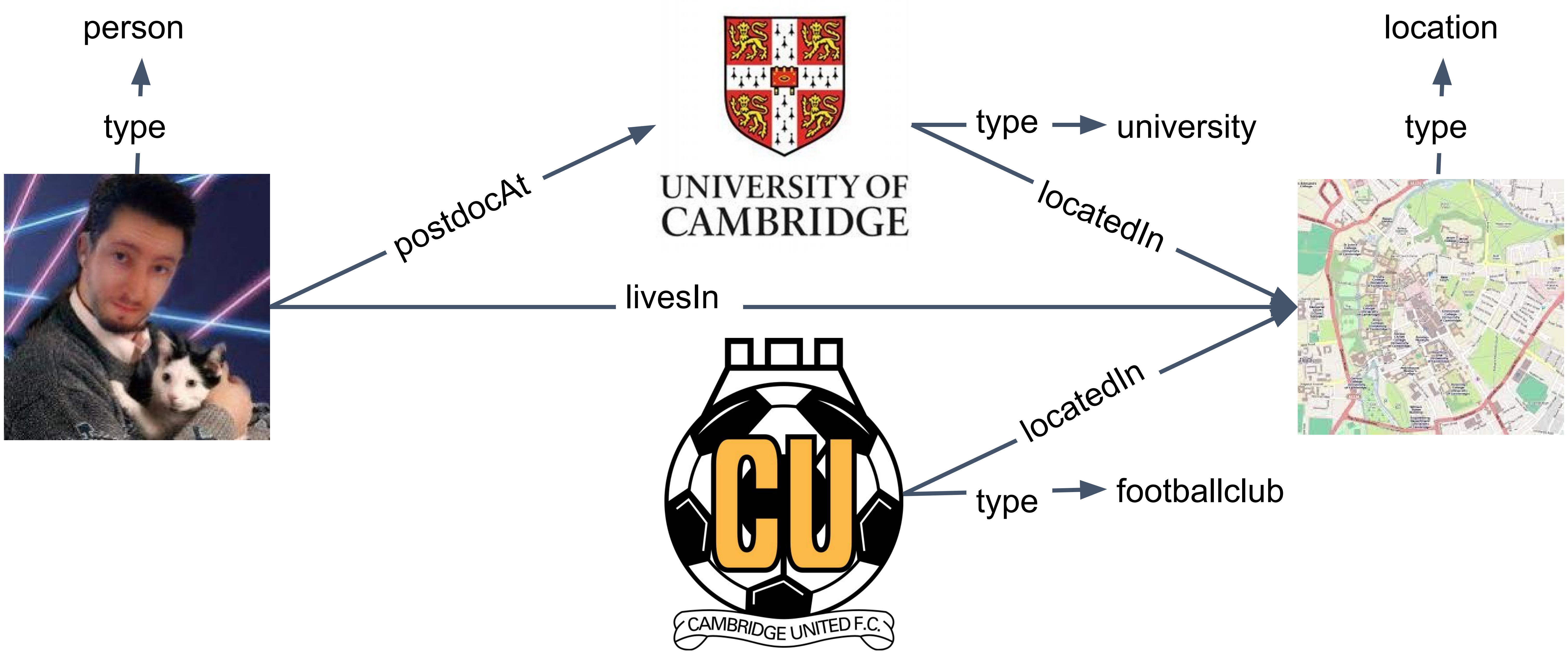


GHENT UNIVERSITY

Outline

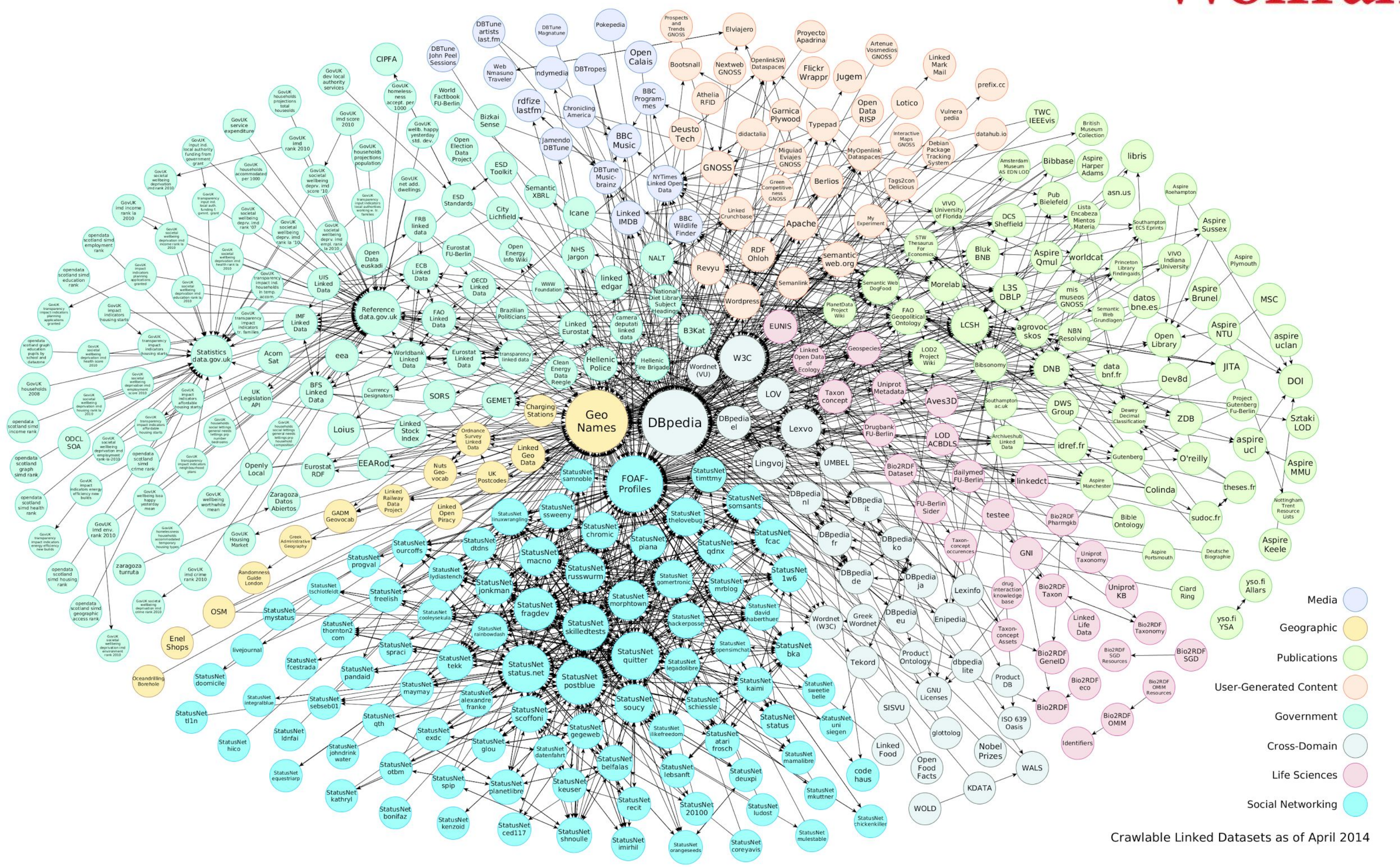
1. **Knowledge Bases**
2. Knowledge Base Population
    a. Knowledge Extraction from Text - TAC KBP
    b. Link Prediction
3. Research Topic at Cambridge

Knowledge Bases as Labeled Graphs

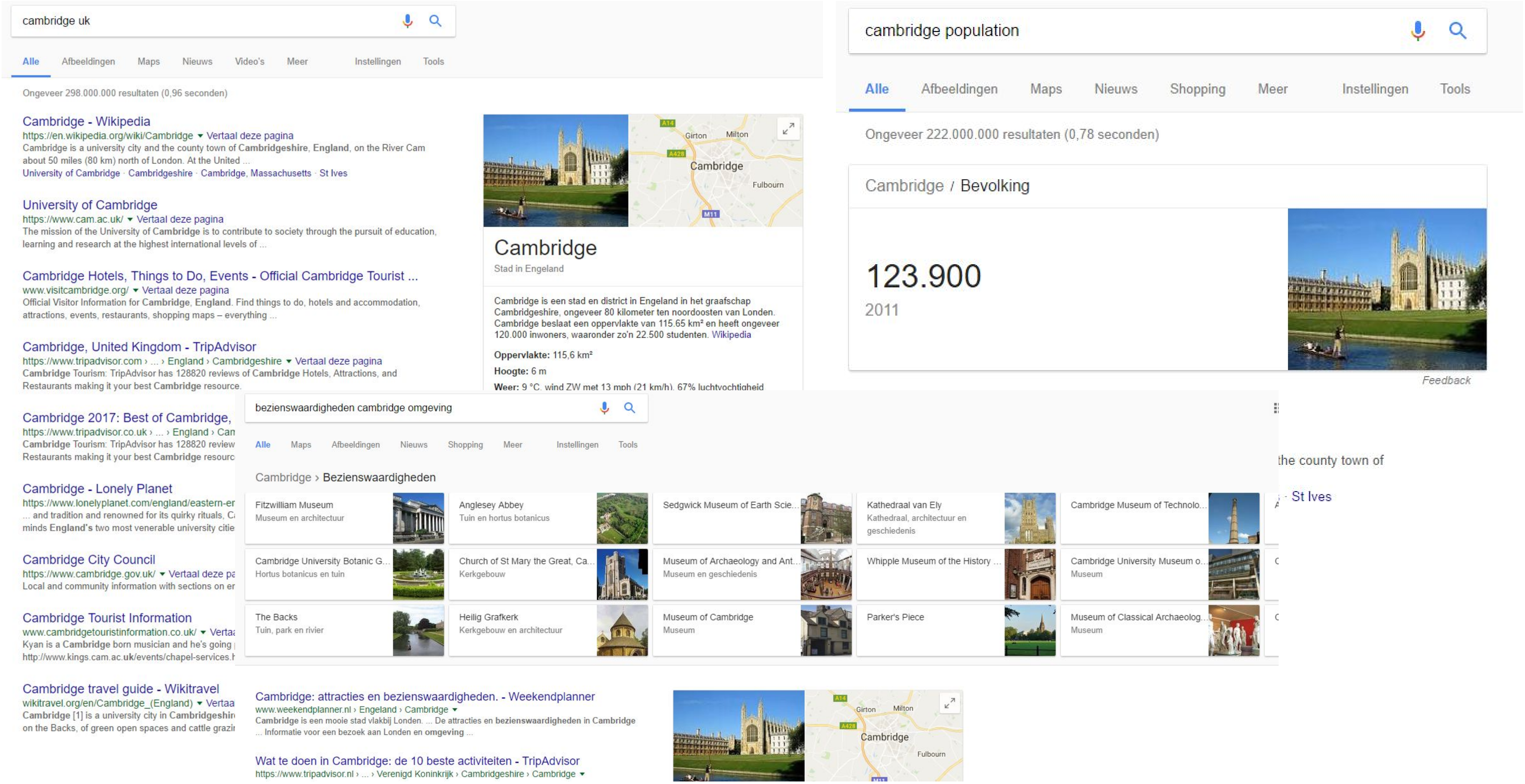


Comprehensive and semantically organized **machine-readable** collection of universally relevant or domain-specific **entities**, **classes**, and **SPO facts** (attributes, relations)

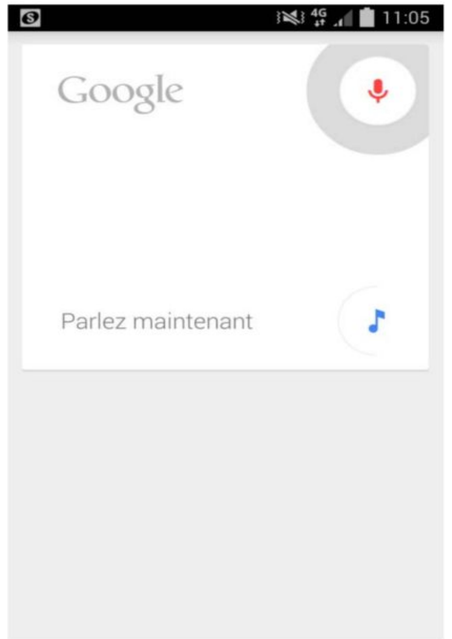
Modern Knowledge Graphs

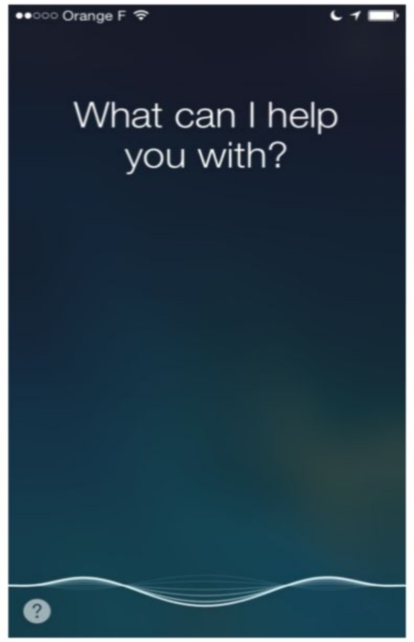


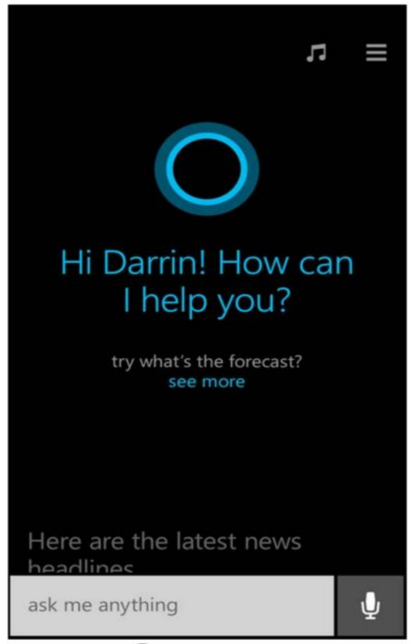
Applications



Applications







Google ASSISTANT

Cortana

Knowledge Bases: Research Challenges

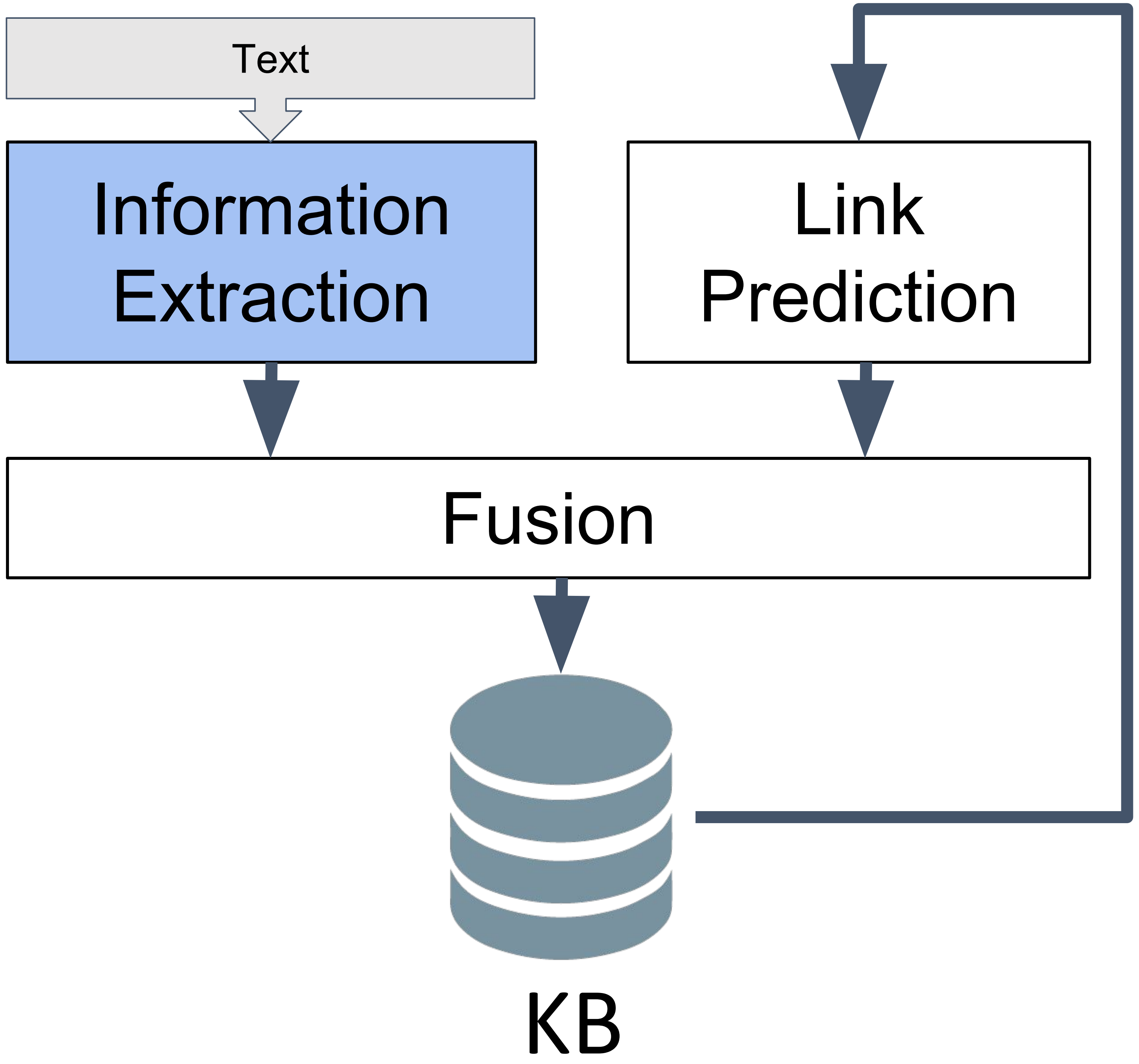
- **Population: Knowledge bases are incomplete**
  - Knowledge extraction from text
  - Link prediction
- **Validation: Knowledge bases contain errors**
  - Entity resolution
  - Error detection, trustworthiness
- **Interface: How to easily access knowledge**
  - Semantic parsing
  - Question answering
- **General AI: can AI emerge from Knowledge Graphs?**
  - Automatic reasoning and planning
  - Generalization and abstraction

Knowledge Bases: Research Challenges

- **Population: Knowledge bases are incomplete**
  - **Knowledge extraction from text**
  - **Link prediction**
- **Validation: Knowledge bases contain errors**
  - Entity resolution
  - Error detection, trustworthiness
- **Interface: How to easily access knowledge**
  - Semantic parsing
  - Question answering
- **General AI: can AI emerge from Knowledge Graphs?**
  - Automatic reasoning and planning
  - Generalization and abstraction

Outline

1. Knowledge Bases
2. **Knowledge Base Population**
    a. **Knowledge Extraction from Text**
    b. Link Prediction
3. Research Topic at Cambridge



Text Analysis Conference (TAC)

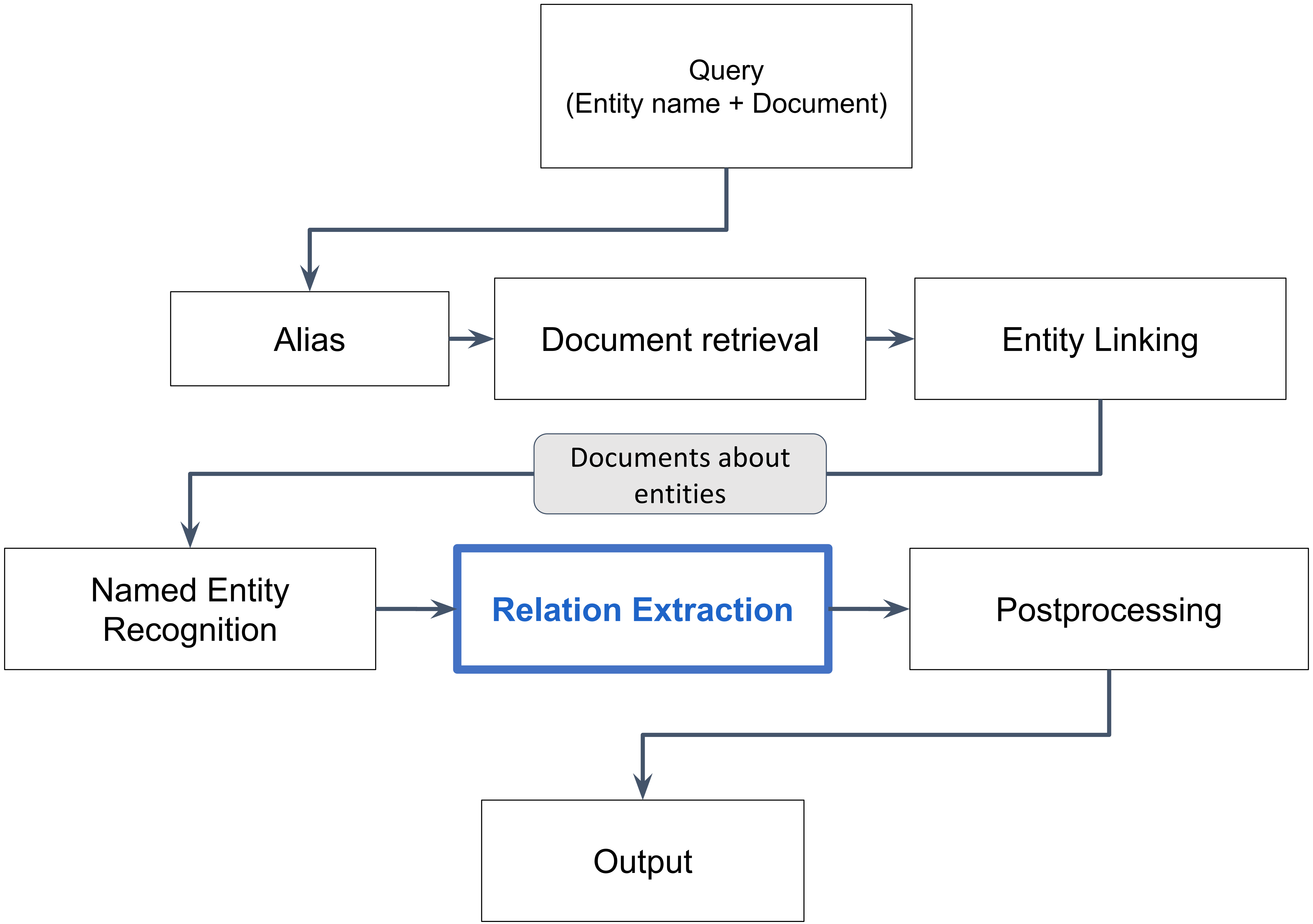
- **Benchmark to promote research** on automated systems that discover information about named entities and incorporate this information in a knowledge source, or database
- Continuation of previous conferences and evaluations, such as MUC and ACE
- Organised by **NIST**, sponsored by US department of Defense
- Extract **41 relations** for **50 persons** and **50 organizations** from **4 million documents**

```
<query id="SF_002">
    <name>PhillyInquirer</name>
    <docid>eng-NG-31-141808-9966244</docid>
    <beg>757</beg>
    <end>770</end>
    <enttype>ORG</enttype>
</query>
```

Person Slots

Name	Type	List?
per:alternate_names	Name	Yes
per:date_of_birth	Value	
per:age	Value	
per:country_of_birth	Name	
per:stateorprovince_of_birth	Name	
per:city_of_birth	Name	

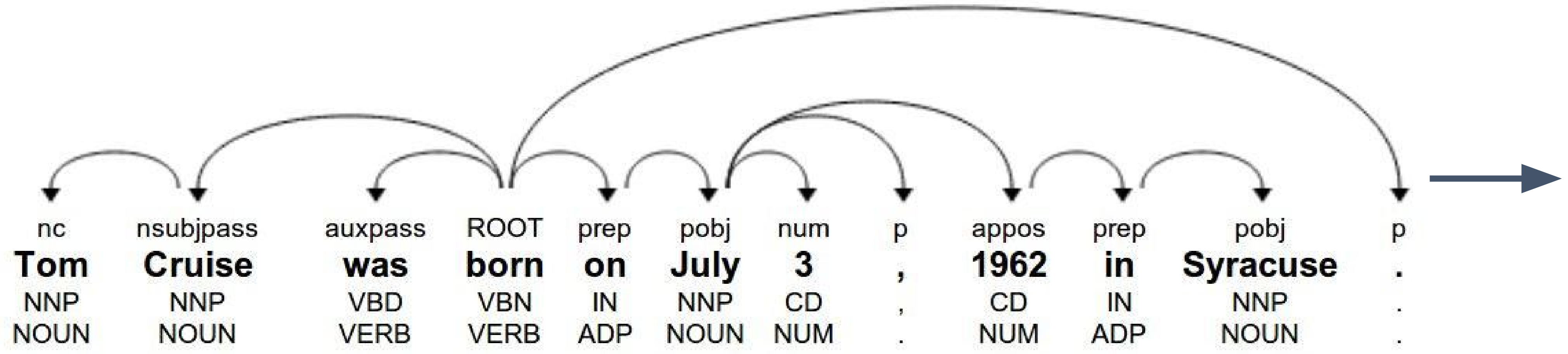
Ghent University at TAC KBP



Supervised Relation Extraction

- **Extracting semantic relations between sets of grounded entities**
- Train classifiers from **+/- Examples**

< person, city_of_birth, location >



X was born on DDDD in Y

- **DEP: X←nsubjpass←on→pobj→ date→prep_in→ Y**
- NER: X=PER, Y=LOC
- POS: X = NOUN, NNP; Y=NOUN, NNP
- Context:born, on, in, “born_on”

✓ Pro’s

- High quality training data
- Explicit negative examples

\- Con’s

- Expensive!
- Can’t generalize to other relations and domains

Distant Supervision

Knowledge Base

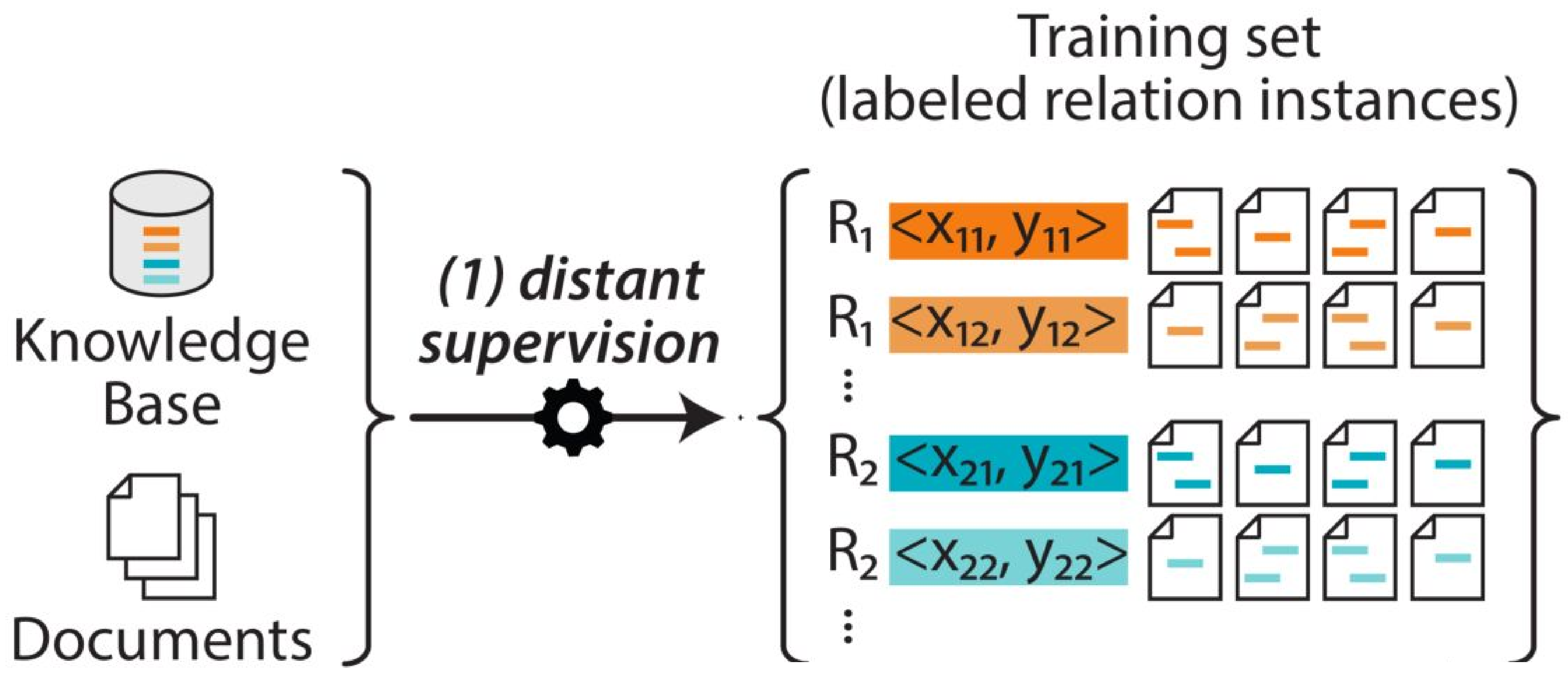
Relation (r)	Entity 1 (e_1)	Entity 2 (e_2)
born_in	Barrack Obama	U.S.
spouse	Barrack Obama	Michelle
...	...	...

Mentions in free text	**True +?**
Barack was born in Honolulu, Hawaii, **U.S.**	✓
Barrack Obama ended **U.S.** military involvement in the Iraq War.	✗
Michelle and **Barack** are visiting Cuba.	✗
Barack and his wife **Michelle** are meeting with Xi Jinpeng	✓

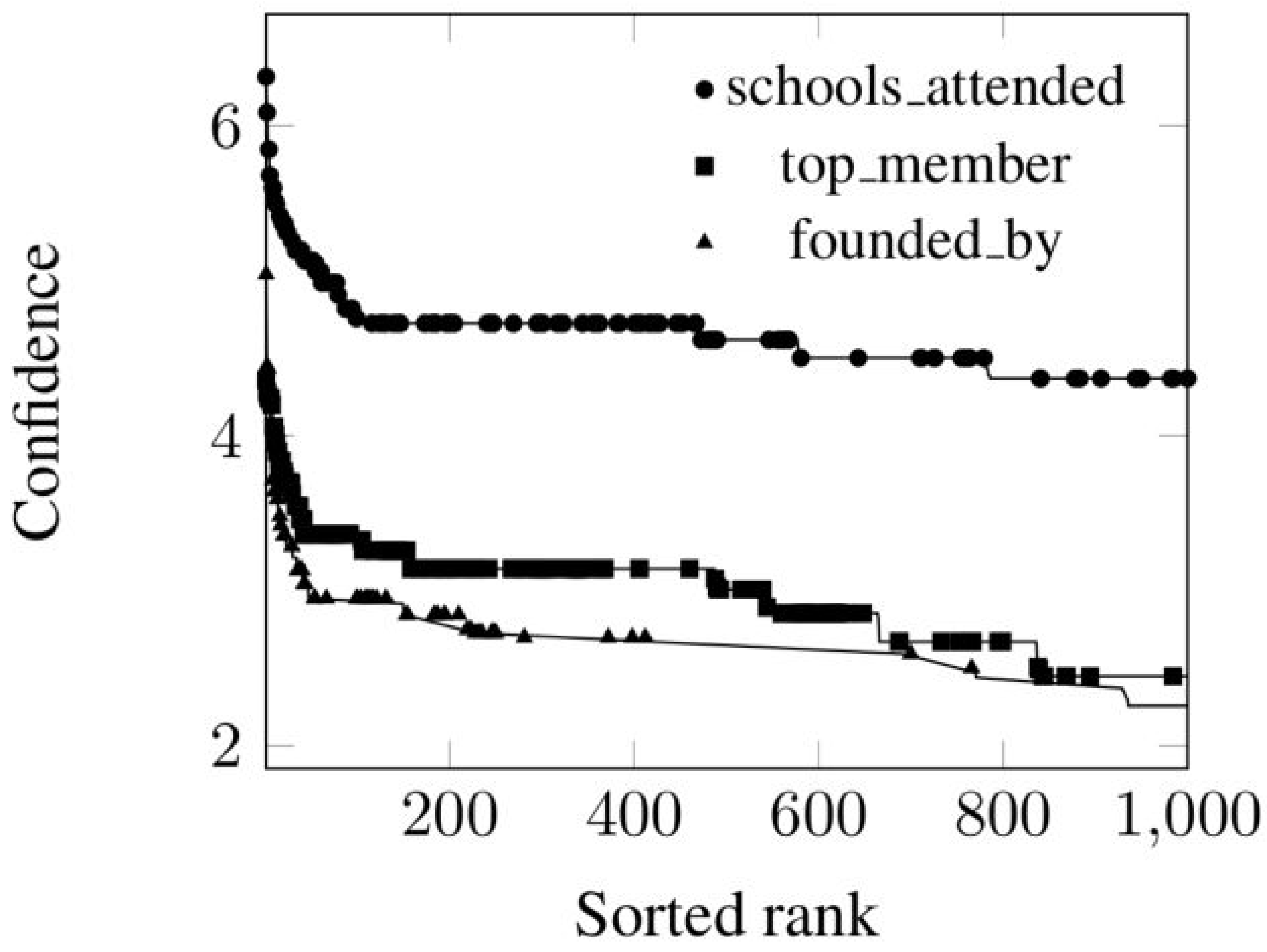
- Existing **Knowledge base + Unlabeled text**
  1. Collect many pairs of entities **co-occurring** in sentences from the corpus (Mintz, 2009)
     - Noise

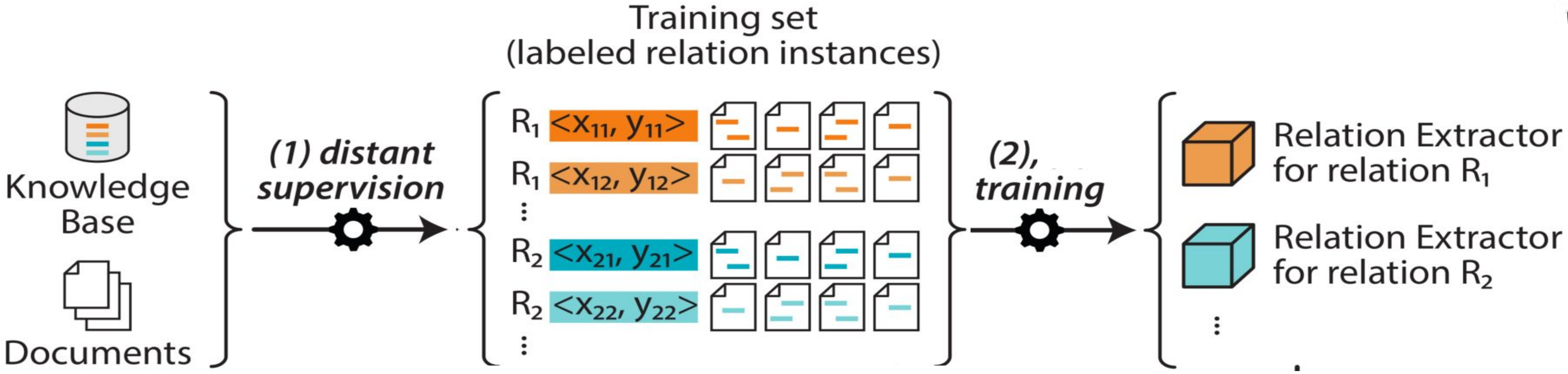
Guiding Bootstrapped Relation Extractors

Knowledge Base Population using Semantic Label Propagation
(Knowledge Based Systems, Sterckx, 2016)



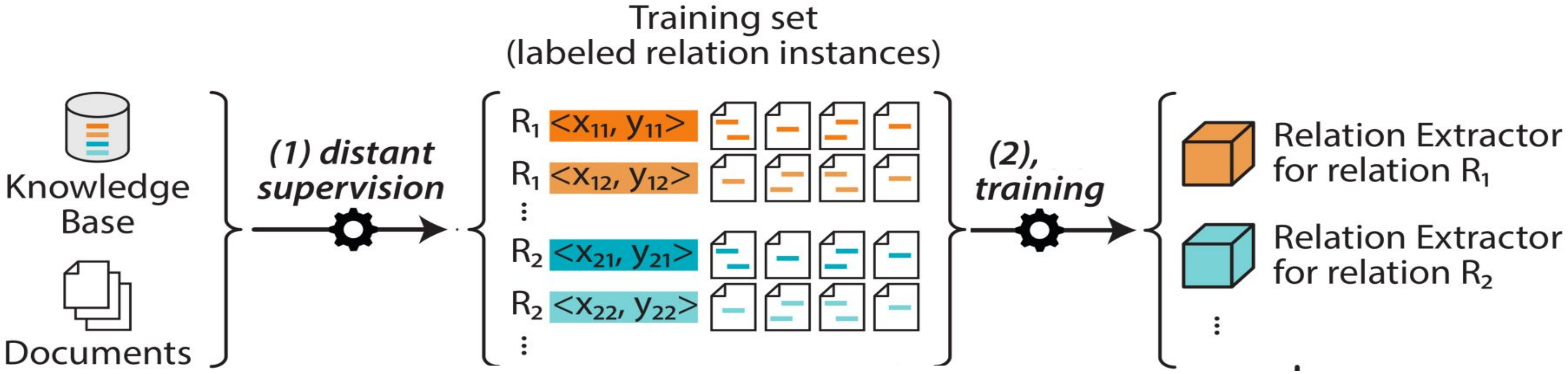
Guiding Bootstrapped Relation Extractors





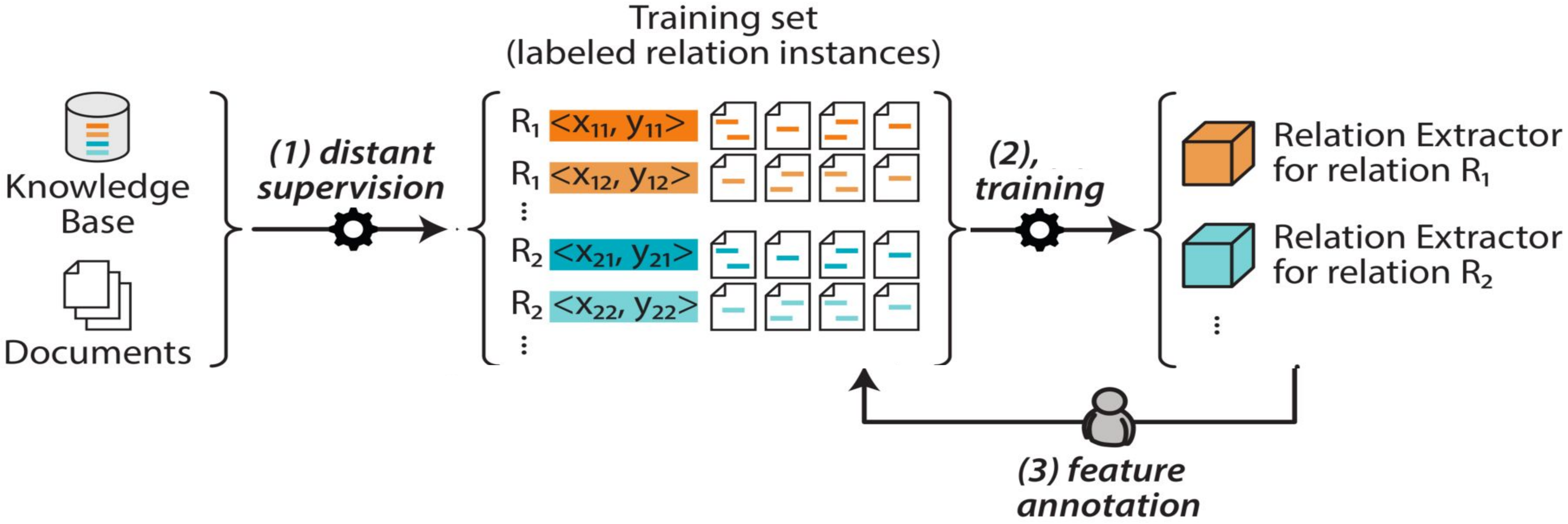
Guiding Bootstrapped Relation Extractors

Relation	Top SDP
top_members_employees	$\mathrm{PER} \xleftarrow{appos} \textit{executive} \xrightarrow{prep\_of} \mathrm{ORG}$ $\mathrm{PER} \xleftarrow{appos} \textit{chairman} \xrightarrow{appos} \mathrm{ORG}$ $\mathrm{ORG} \xleftarrow{nn} \textit{founder} \xrightarrow{prep\_of} \mathrm{PER}$
children	$\mathrm{PER\text{-}2} \xleftarrow{appos} \textit{son} \xrightarrow{prep\_of} \mathrm{PER\text{-}1}$ $\mathrm{PER\text{-}1} \xleftarrow{appos} \textit{father} \xrightarrow{prep\_of} \mathrm{PER\text{-}2}$ $\mathrm{PER\text{-}2} \xleftarrow{nn} \textit{grandson} \xrightarrow{prep\_of} \mathrm{PER\text{-}1}$
city_of_birth	$\mathrm{PER} \xleftarrow{rcmod} \textit{born} \xrightarrow{prep\_in} \mathrm{LOC}$ $\mathrm{PER} \xleftarrow{nsubj} \textit{mayor} \xrightarrow{prep\_of} \mathrm{LOC}$ $\mathrm{PER} \xleftarrow{appos} \textit{historian} \xrightarrow{prep\_from} \mathrm{LOC}$



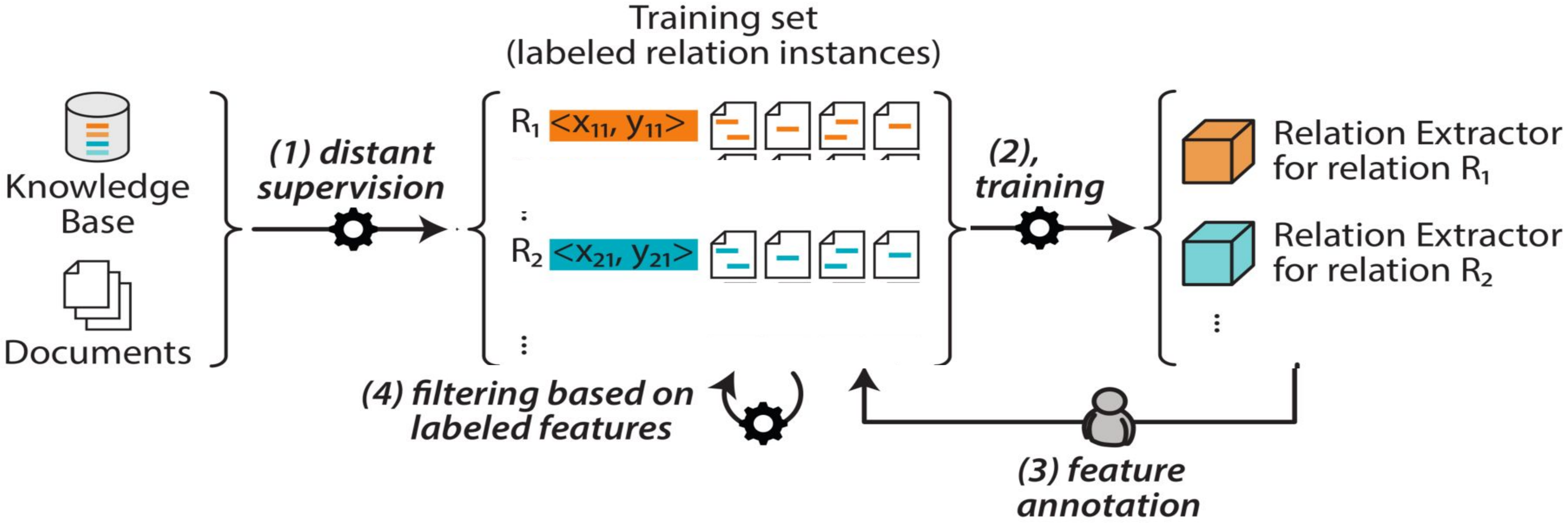
Guiding Bootstrapped Relation Extractors

Relation	Top SDP	Assessment
top_members_employees	PER $\xleftarrow{appos}$ *executive* $\xrightarrow{prep\_of}$ ORG	✓
	PER $\xleftarrow{appos}$ *chairman* $\xrightarrow{appos}$ ORG	✓
	ORG $\xleftarrow{nn}$ *founder* $\xrightarrow{prep\_of}$ PER	✗
children	PER-2 $\xleftarrow{appos}$ *son* $\xrightarrow{prep\_of}$ PER-1	✓
	PER-1 $\xleftarrow{appos}$ *father* $\xrightarrow{prep\_of}$ PER-2	✓
	PER-2 $\xleftarrow{nn}$ *grandson* $\xrightarrow{prep\_of}$ PER-1	✗
city_of_birth	PER $\xleftarrow{rcmod}$ *born* $\xrightarrow{prep\_in}$ LOC	✓
	PER $\xleftarrow{nsubj}$ *mayor* $\xrightarrow{prep\_of}$ LOC	✗
	PER $\xleftarrow{appos}$ *historian* $\xrightarrow{prep\_from}$ LOC	✗

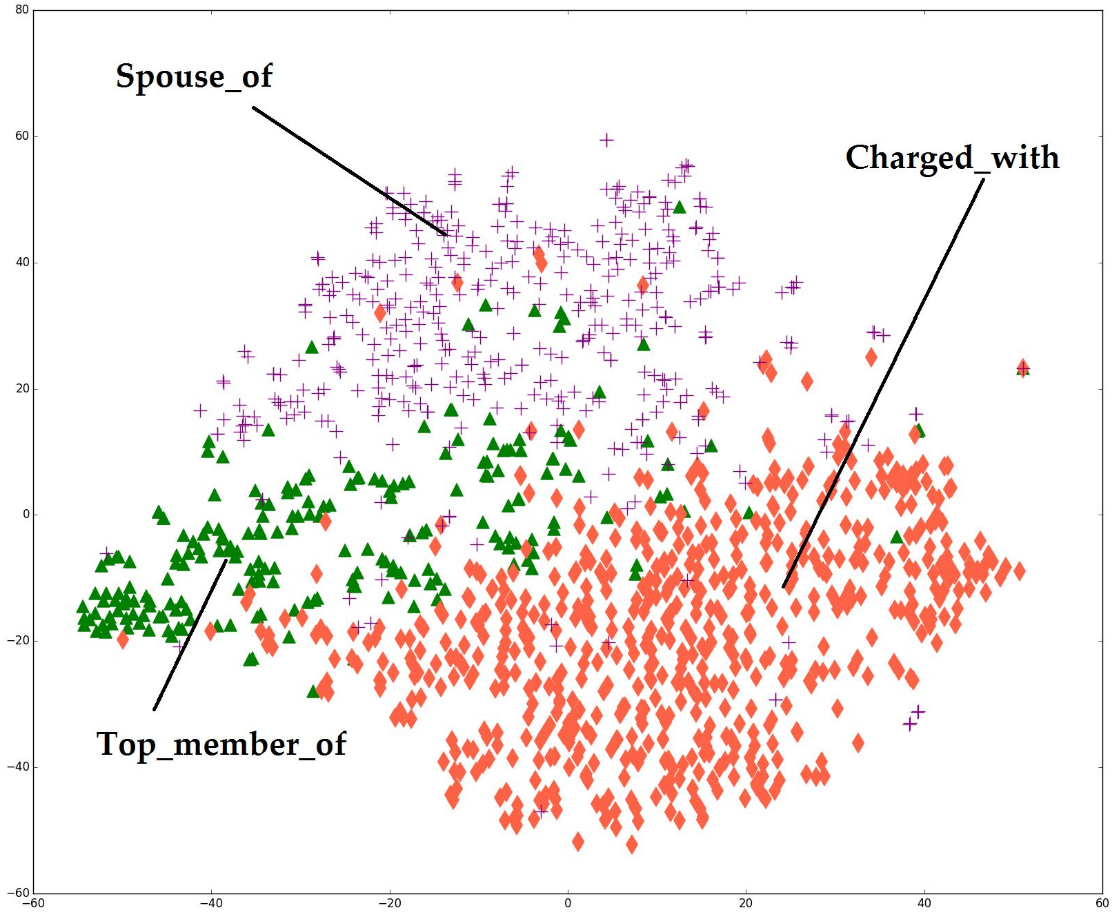


Guiding Bootstrapped Relation Extractors

- **Filter non-labeled patterns**
  - **Include weaker features in log-linear classifiers**
  - **Regularize**
- **Recall ↓↓**



Guiding Bootstrapped Relation Extractors



Labeled Shortest Dependency Path

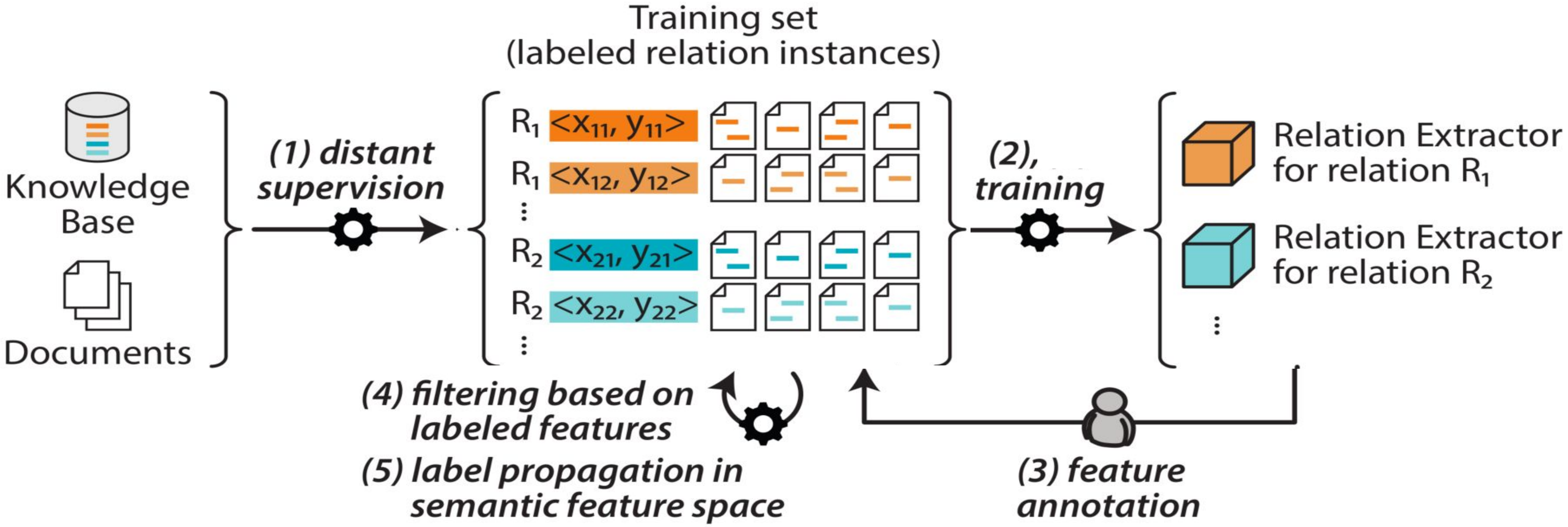
$$\text{PER} \xleftarrow{appos} \textit{executive} \xrightarrow{prep\_of} \text{ORG}$$

Embedding

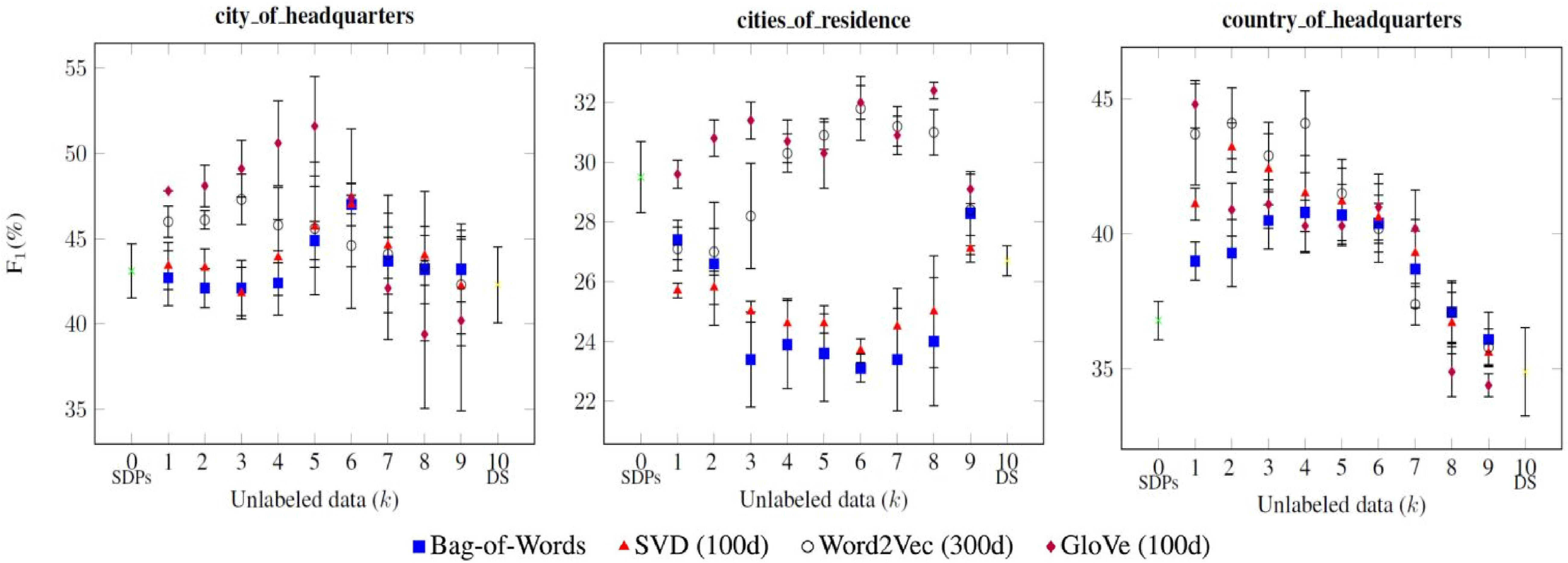
$$\vec{C} = \text{CBOW(executive,of)}$$

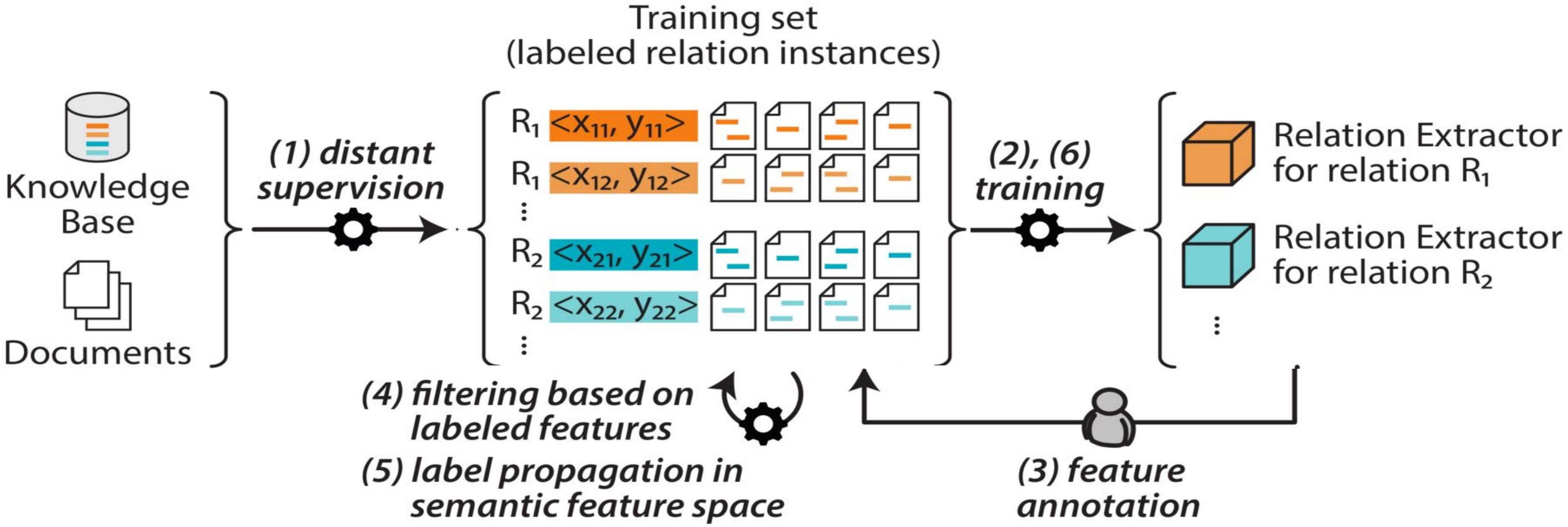
Cosine Similarity

$$Sim(\vec{C}_t, \vec{C}_{DS}) = \frac{\vec{C}_t.\vec{C}_{DS}}{|\vec{C}_t|.|\vec{C}_{DS}|}$$



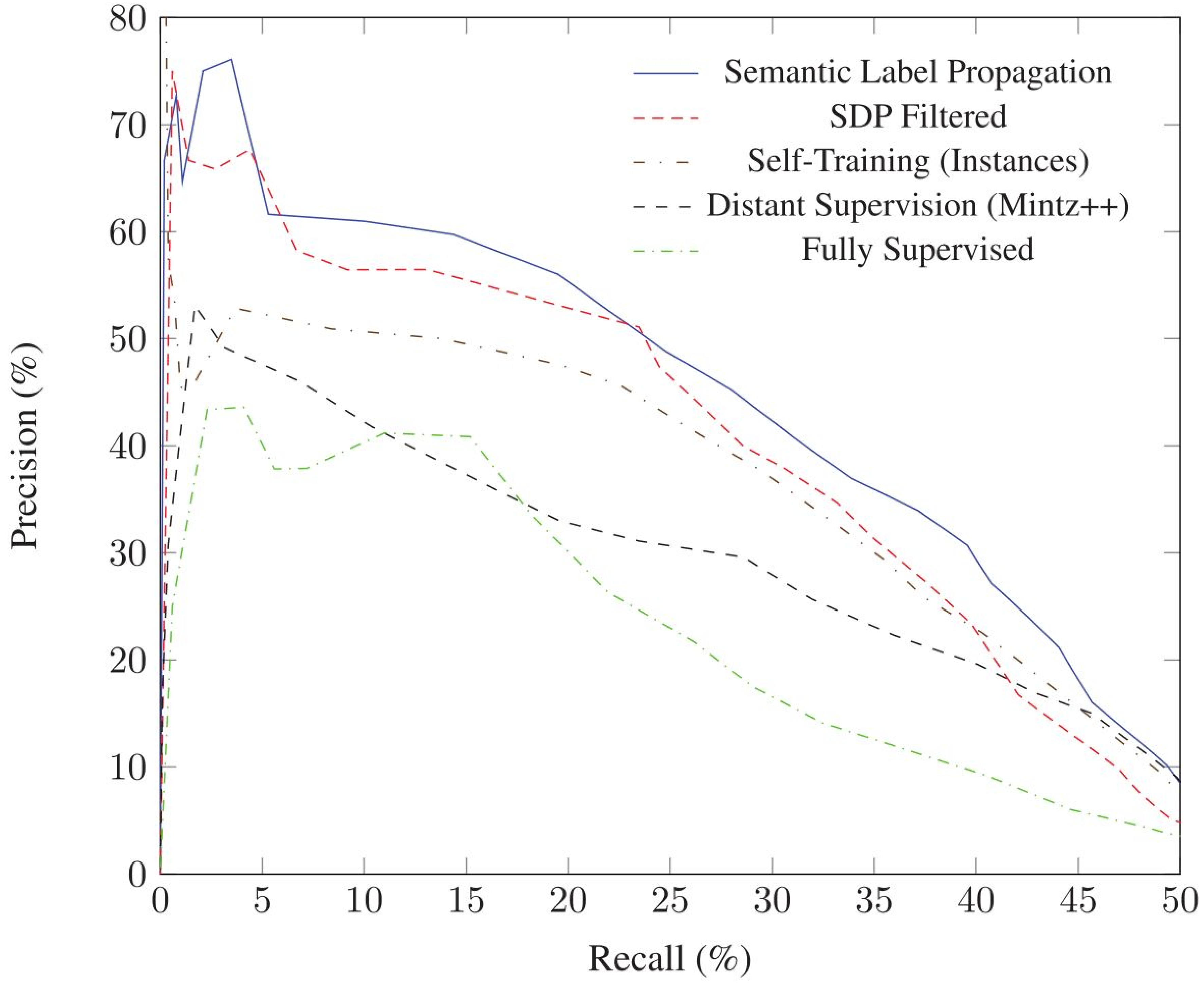
Guiding Bootstrapped Relation Extractors



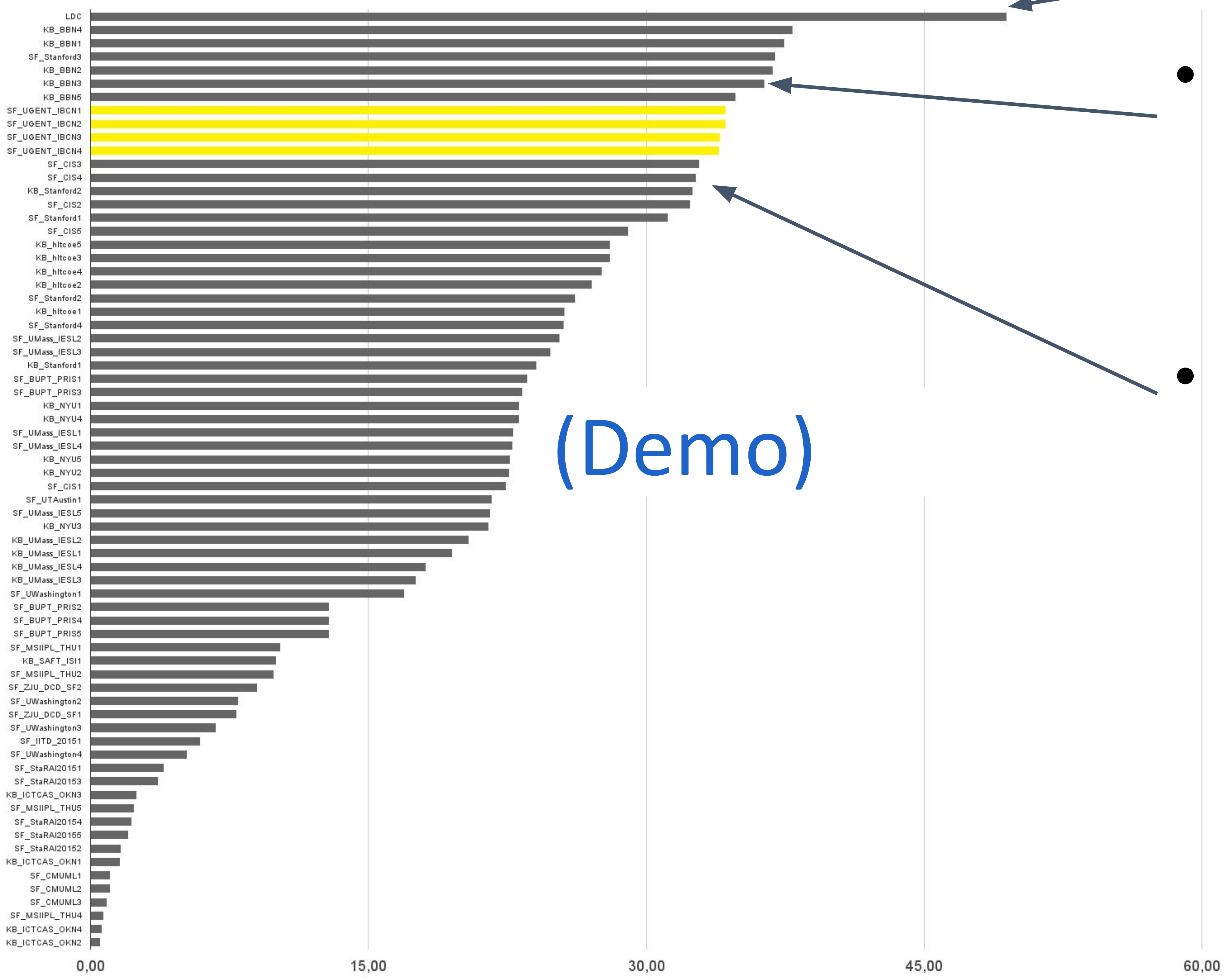


Ghent University at TAC KBP

- **Recall ↑↑**
- **Minimal** supervision (5 min. per relation, 2u30 for TAC-KBP)



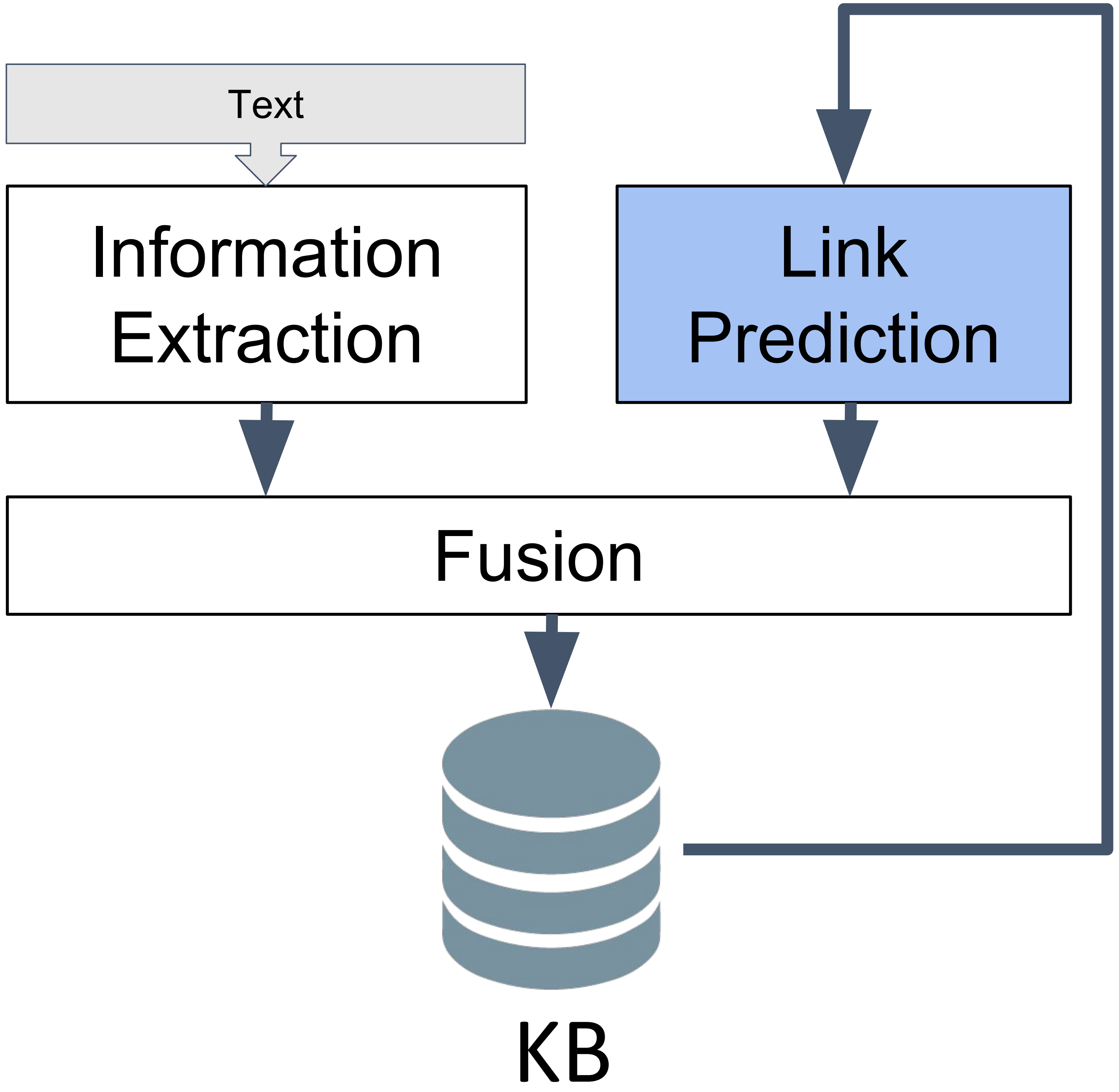
Ghent University at TAC KBP



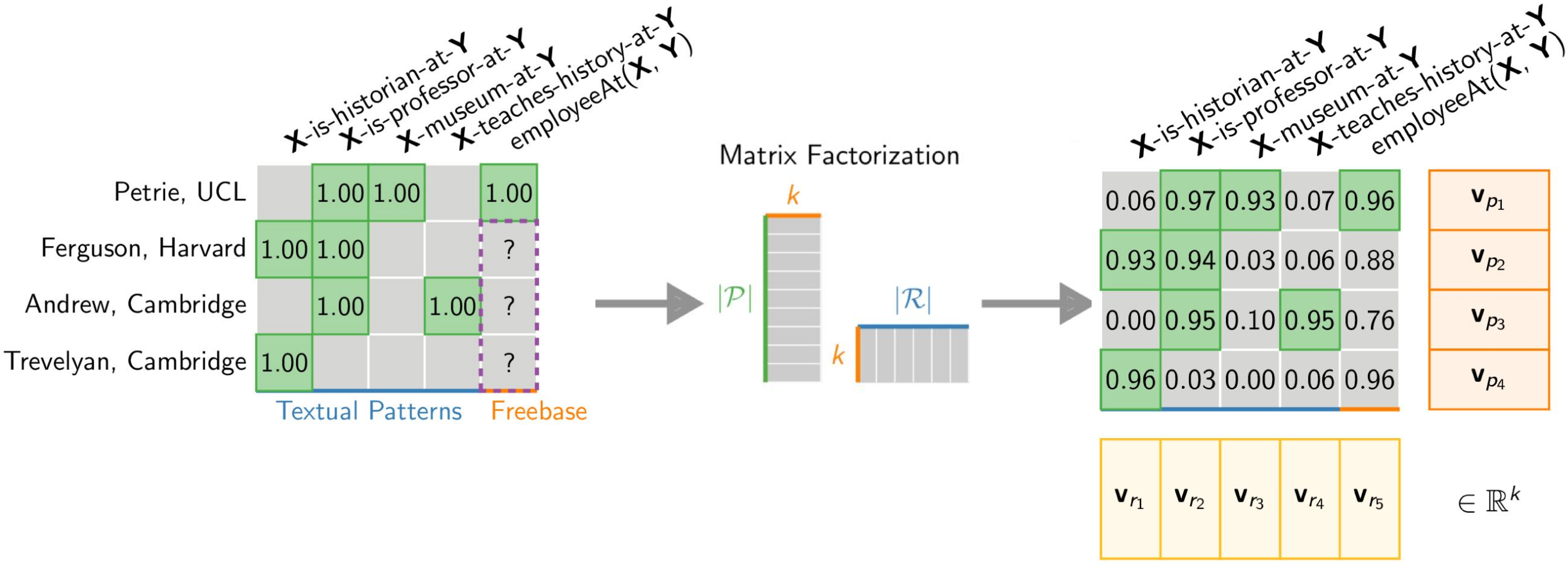
- Human
- DeepDive (Stanford)
  - ~40 hours total writing patterns
  - Supercomputer (786 Gig RAM)
- Ensemble of feature-based classifiers neural architectures, patterns,...

Outline

1. Knowledge Bases
2. **Knowledge Base Population**
    a. Knowledge Extraction from Text - TAC KBP
    b. **Link Prediction**
3. Research Topic at Cambridge



Relation Extraction by Matrix Factorization (Riedel, 2013)



$$p(\text{fact}) = p(x_{ij} = 1 \mid \textit{Graph})$$

https://rockt.github.io/slides/2015-naacl.pdf [neural] vector representations,

+ Similarity, approximate inference

- Fails for little alignment, hard to fix mistakes

[symbolic] efficient (“lifted”) injection of prior knowledge “`prof_at` ⇒`works_for`”

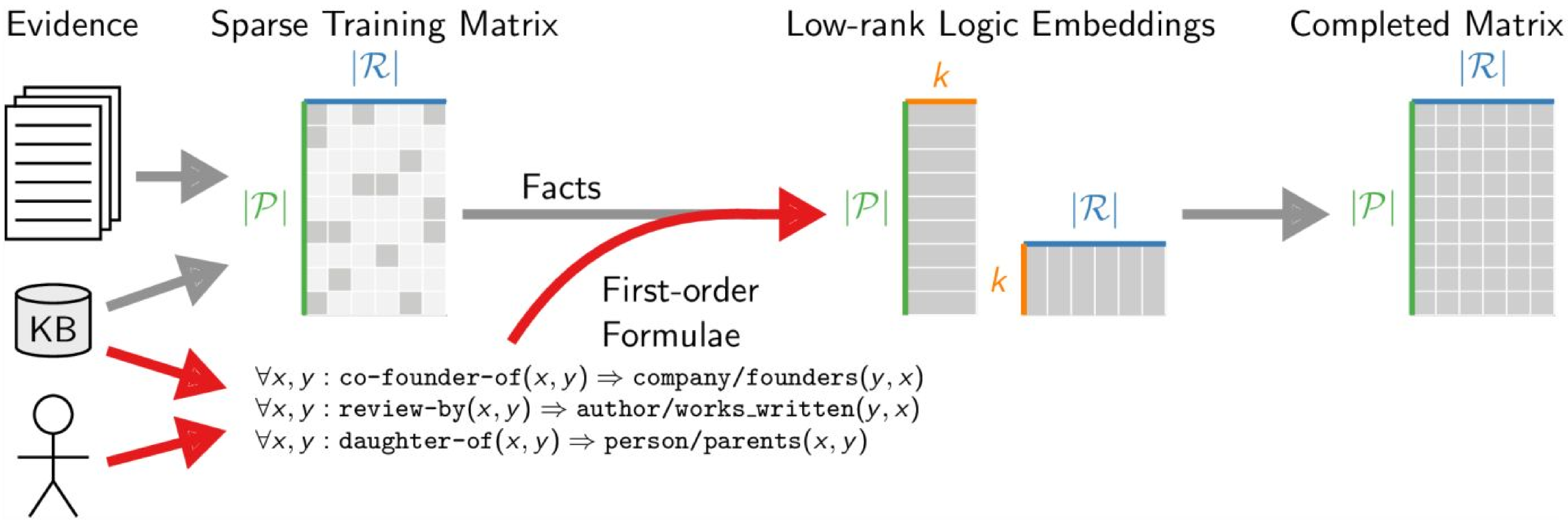
+ Easy to modify

- Brittle, no generalization

combine **neural** and **symbolic** representations

to leverage advantages of both

Injecting Logical Formulae (Rocktäschel, 2015)



- Inject general 1st order formulae
  - expressed in terms of probabilities of all training facts
  - e.g. model for $r_p \Rightarrow r_q$: by grounding over entities

$$p\big((r_p, e) \Rightarrow (r_q, e)\big) \approx 1 - p(r_p, e)\big(1 - p(r_q, e)\big)$$

- Lessons learned:

  + joint training of facts and rules works best

  - due to grounding, only practical for few rules

https://rockt.github.io/slides/2015-naacl.pdf

Lifted implication rules (Demeester, 2016)

When is rule "$\texttt{prof\_at} \Rightarrow \texttt{works\_for}$" satisfied?

$$\forall e \in \mathcal{E} : p\big(\texttt{prof\_at}(e)\big) \leq p\big(\texttt{works\_for}(e)\big)$$

$$\sigma\left(\boldsymbol{v}_{\texttt{prof\_at}} \cdot \boldsymbol{v}_{\boldsymbol{e}}\right) \leq \sigma\left(\boldsymbol{v}_{\texttt{works\_for}} \cdot \boldsymbol{v}_{\boldsymbol{e}}\right)$$

$$\boldsymbol{v}_{\texttt{prof\_at}} \cdot \boldsymbol{v}_{\boldsymbol{e}} \leq \boldsymbol{v}_{\texttt{works\_for}} \cdot \boldsymbol{v}_{\boldsymbol{e}}$$

"compatibility"

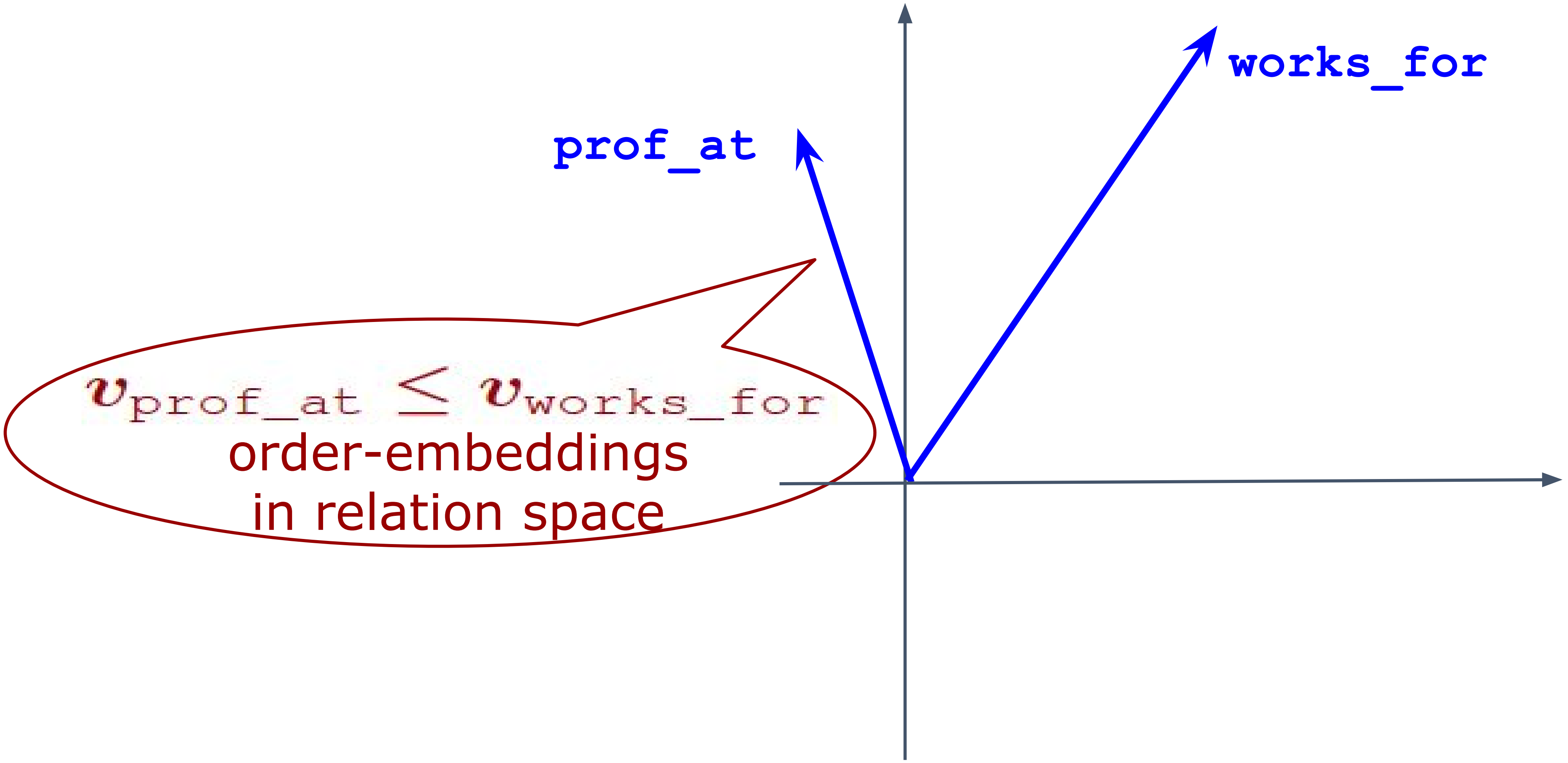
Sufficient (even stricter) condition:

$$\begin{cases} \boldsymbol{v}_{\texttt{prof\_at}} \leq \boldsymbol{v}_{\texttt{works\_for}} & \text{ordered relation embeddings} \\ \forall e \in \mathcal{E} : \boldsymbol{v}_{\boldsymbol{e}} \in \mathbb{R}^{k,+} & \text{non-negative tuple embeddings} \end{cases}$$

Lifted implication rules - illustration

rule $\texttt{prof\_at} \Rightarrow \texttt{works\_for}$

becomes: $\forall \boldsymbol{v_e} : \boldsymbol{v}_{\texttt{prof\_at}} \cdot \boldsymbol{v_e} \leq \boldsymbol{v}_{\texttt{works\_for}} \cdot \boldsymbol{v_e}$



Lifted implication rules - illustration

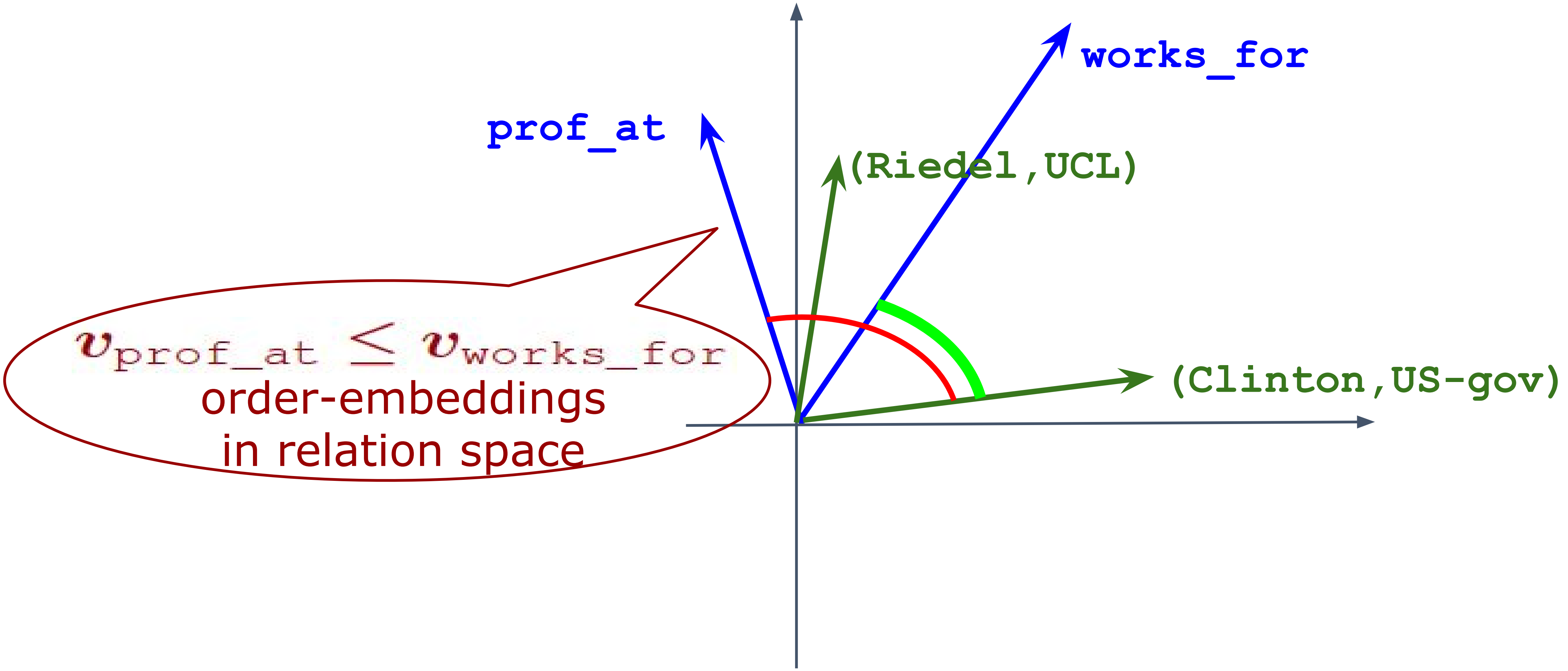
rule `prof_at` ⇒ `works_for`

becomes: $$\forall v_e : v_{\texttt{prof\_at}} \cdot v_e \leq v_{\texttt{works\_for}} \cdot v_e$$

Given: training facts

`works_for(Clinton,US-Gov)`

`prof_at(Riedel,UCL)`



Lifted implication rules - illustration

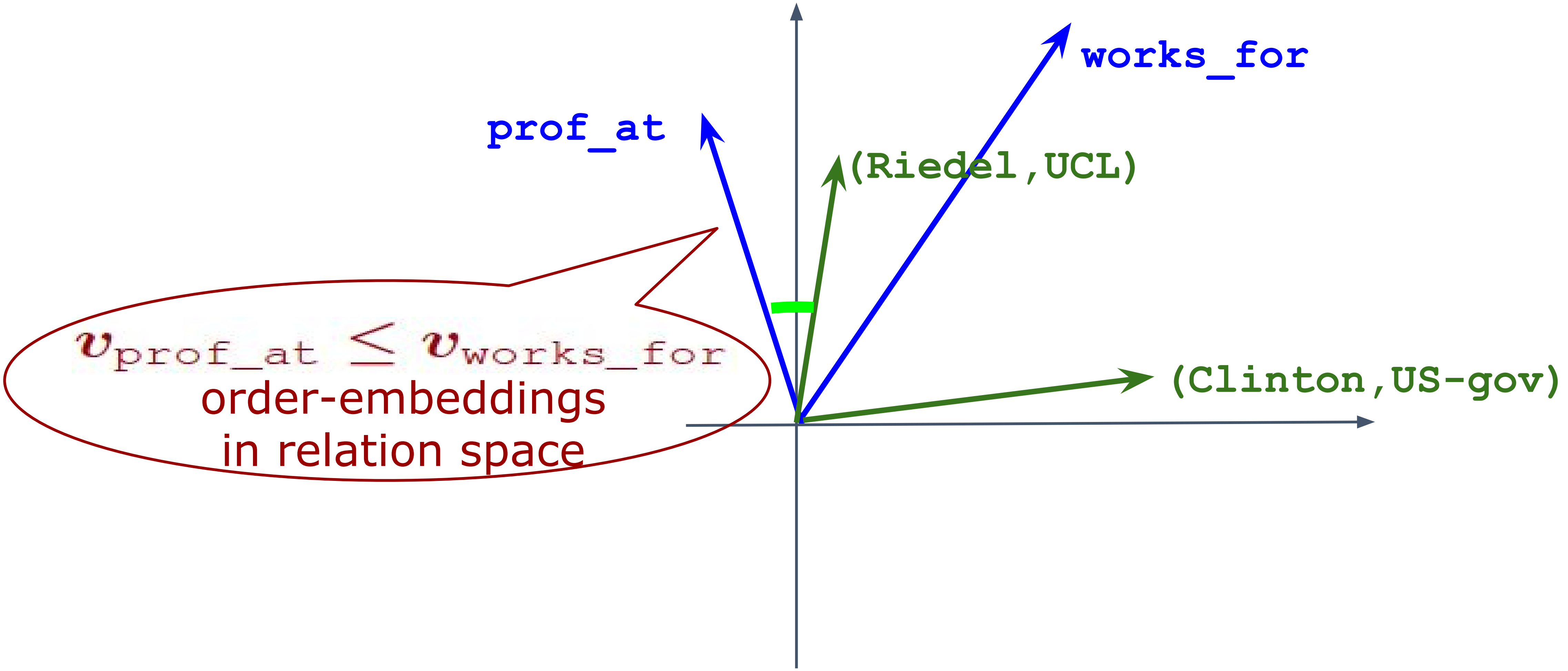
rule `prof_at` ⇒ `works_for`

becomes: $\forall \boldsymbol{v}_e : \boldsymbol{v}_{\mathtt{prof\_at}} \cdot \boldsymbol{v}_e \leq \boldsymbol{v}_{\mathtt{works\_for}} \cdot \boldsymbol{v}_e$

Given: training facts

`works_for(Clinton,US-Gov)`

`prof_at(Riedel,UCL)`



Lifted implication rules - illustration

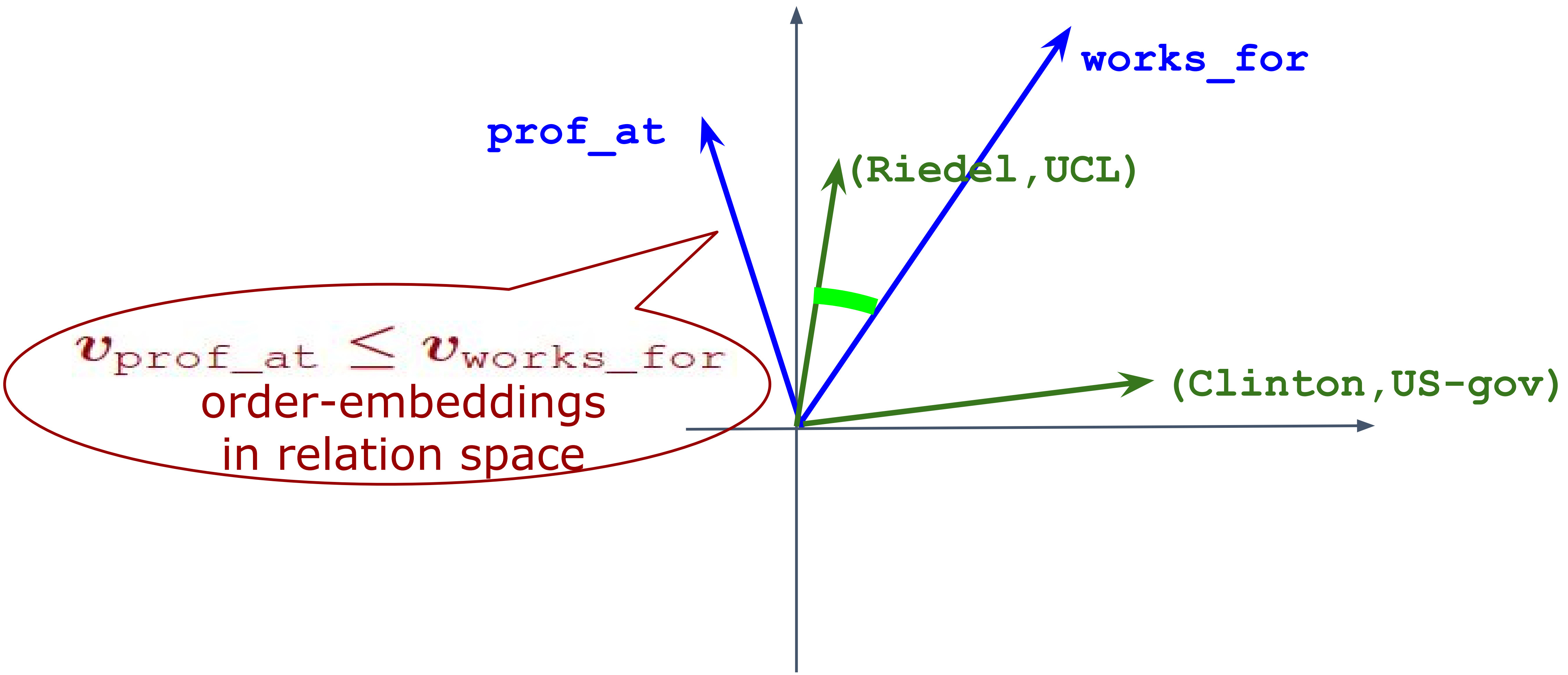
rule `prof_at` ⇒ `works_for`

becomes: $\forall \boldsymbol{v_e} : \boldsymbol{v}_{\text{prof\_at}} \cdot \boldsymbol{v_e} \leq \boldsymbol{v}_{\text{works\_for}} \cdot \boldsymbol{v_e}$

Given: training facts

`works_for(Clinton,US-Gov)`

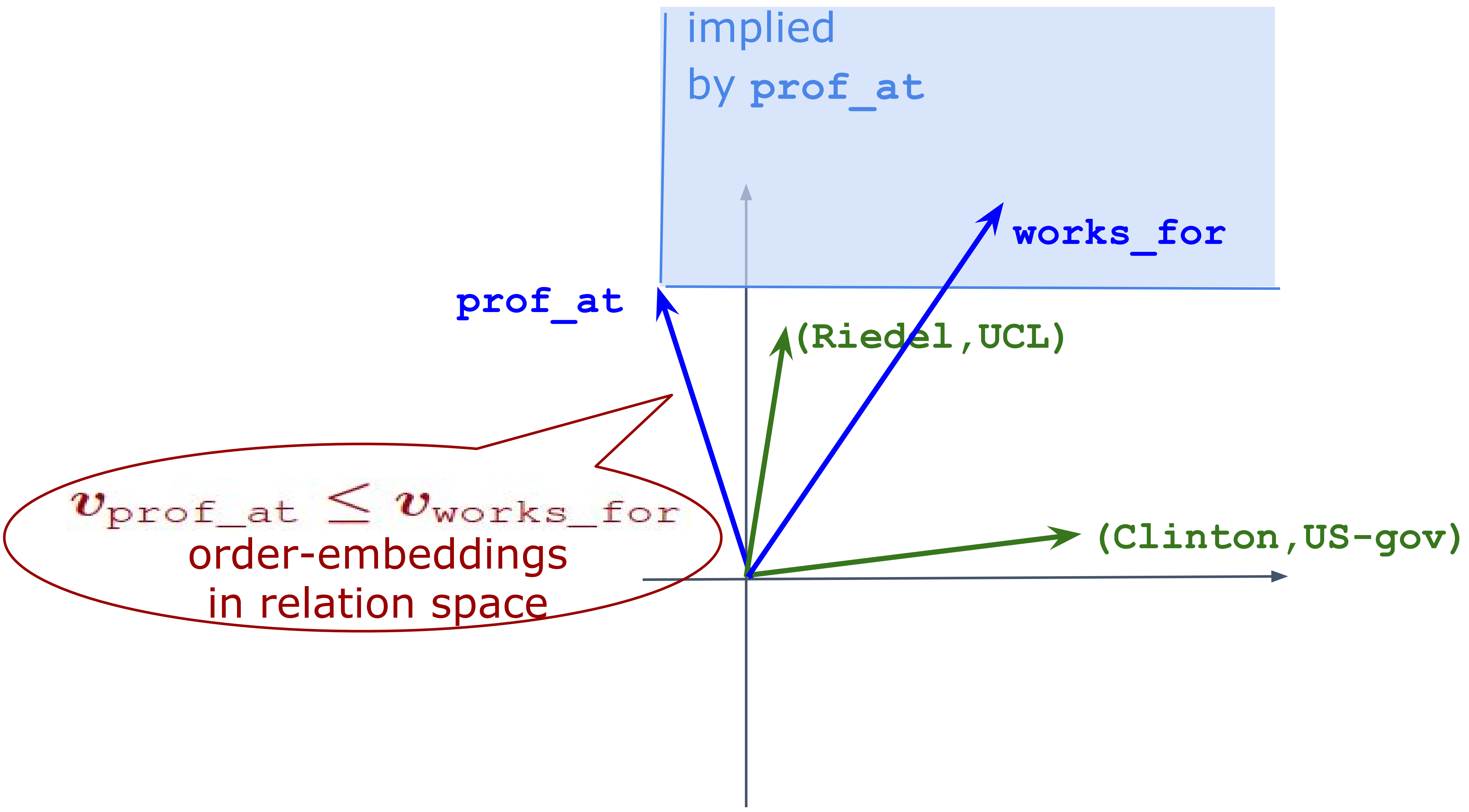
`prof_at(Riedel,UCL)`



Lifted implication rules - illustration

rule `prof_at` $\Rightarrow$ `works_for`

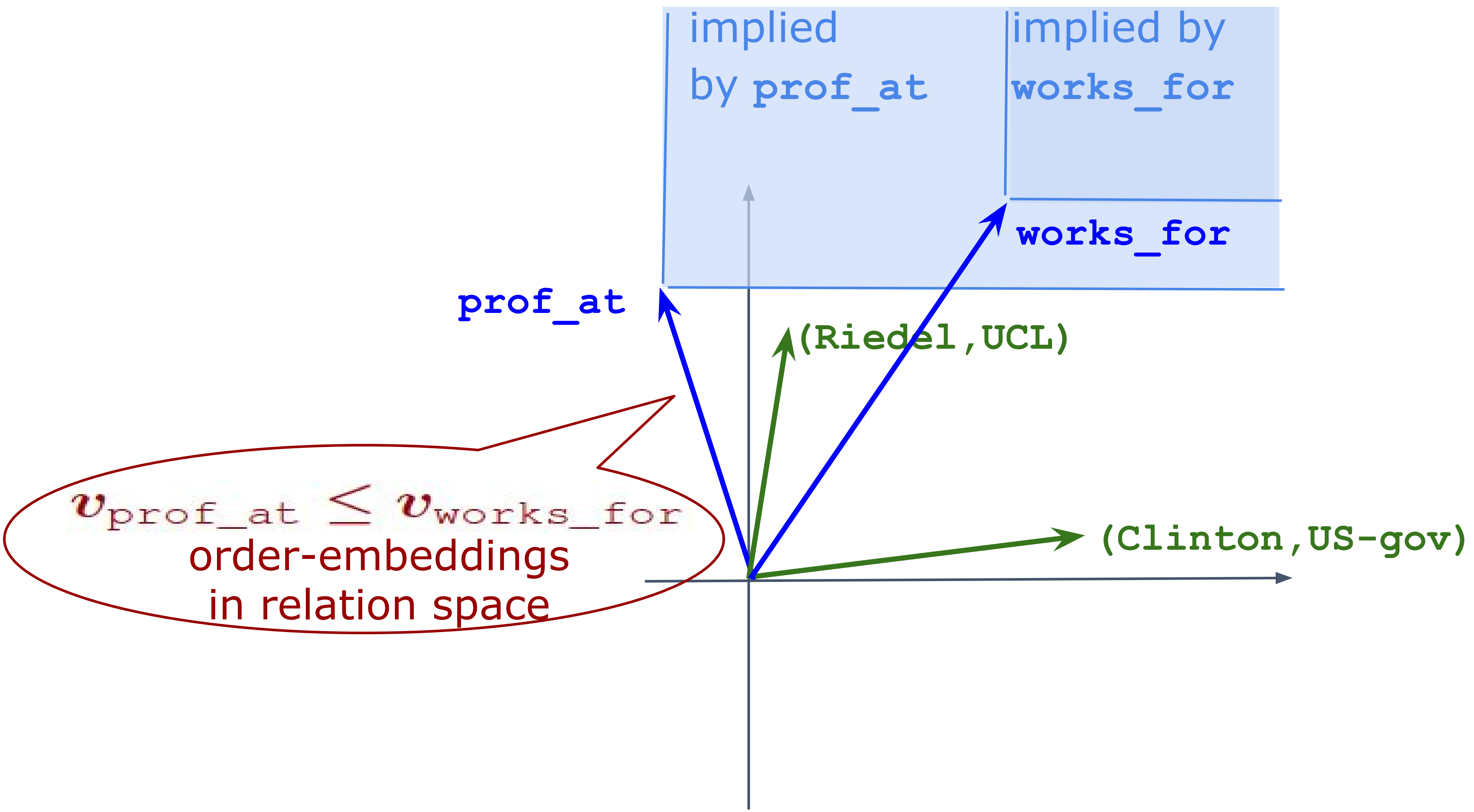
becomes: $\forall \boldsymbol{v}_e : \boldsymbol{v}_{\texttt{prof\_at}} \cdot \boldsymbol{v}_e \leq \boldsymbol{v}_{\texttt{works\_for}} \cdot \boldsymbol{v}_e$



Lifted implication rules - illustration

rule `prof_at` $\Rightarrow$ `works_for`

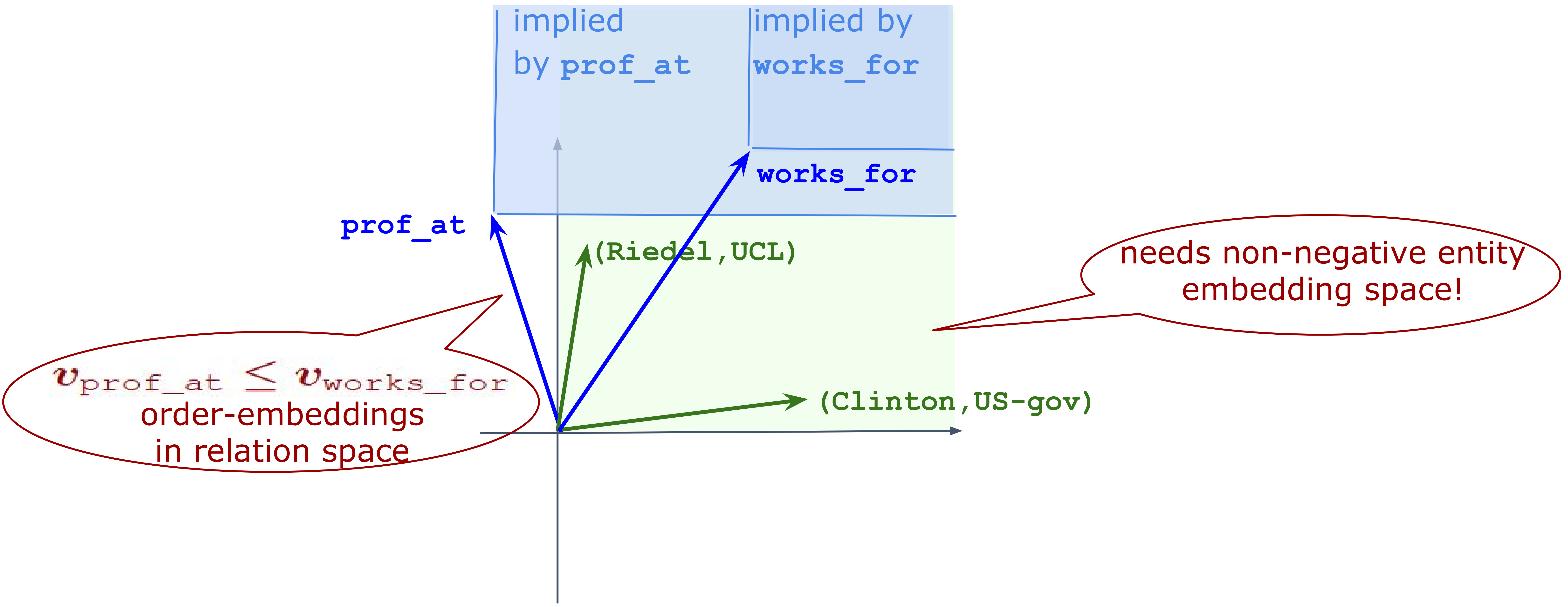
becomes: $\forall \boldsymbol{v}_e : \boldsymbol{v}_{\texttt{prof\_at}} \cdot \boldsymbol{v}_e \leq \boldsymbol{v}_{\texttt{works\_for}} \cdot \boldsymbol{v}_e$



Lifted implication rules - illustration

rule `prof_at` ⇒ `works_for`

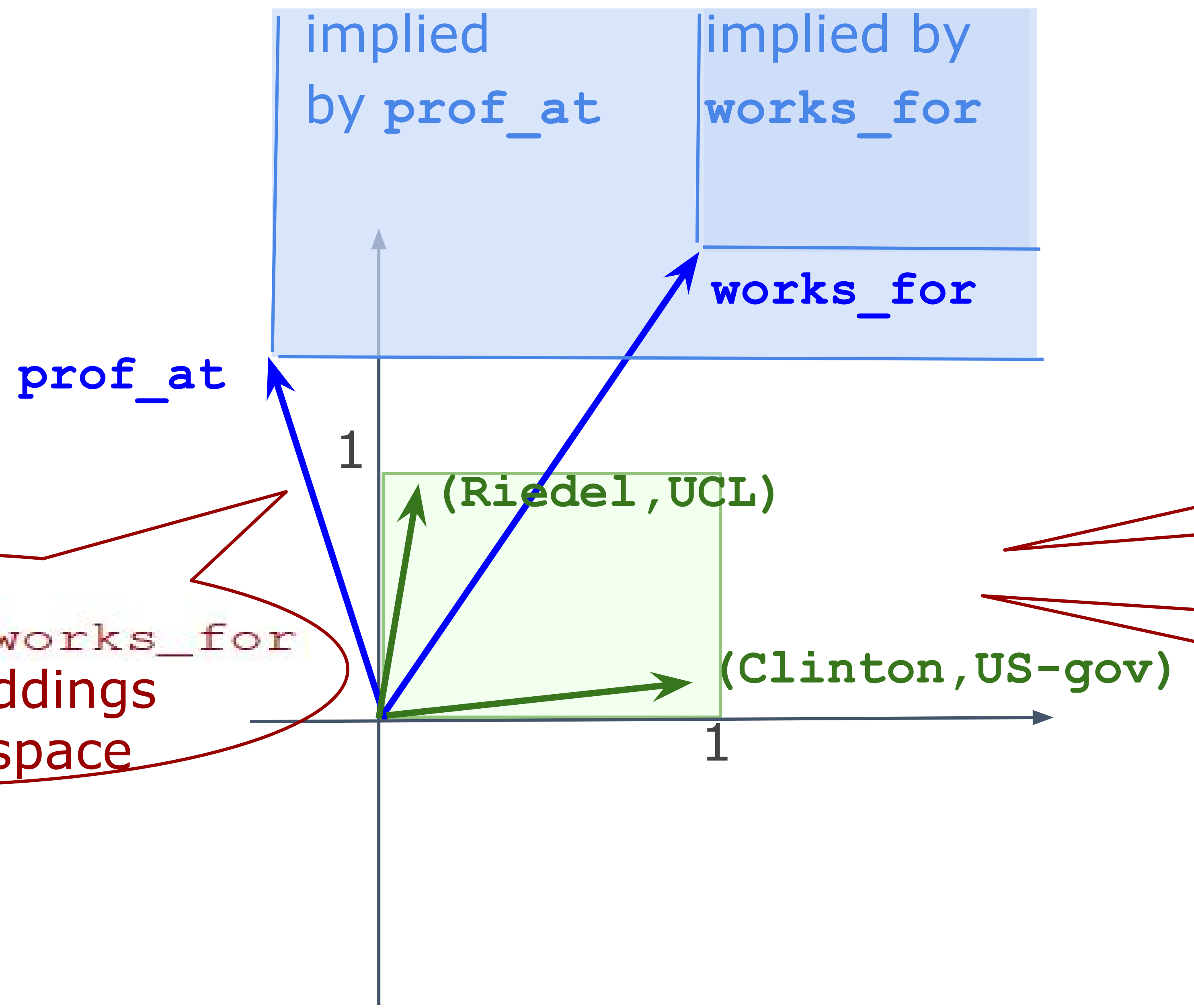
becomes: $\forall \boldsymbol{v}_e : \boldsymbol{v}_{\texttt{prof\_at}} \cdot \boldsymbol{v}_e \leq \boldsymbol{v}_{\texttt{works\_for}} \cdot \boldsymbol{v}_e$



Lifted implication rules - illustration

rule `prof_at` ⇒ `works_for`

becomes: $\forall v_e : v_{\texttt{prof\_at}} \cdot v_e \leq v_{\texttt{works\_for}} \cdot v_e$



$v_{\texttt{prof\_at}} \leq v_{\texttt{works\_for}}$
order-embeddings in relation space

needs non-negative entity embedding space!

in practice: even better with **approximately boolean embeddings**

Lifted implication rules - in practice

Non-negative entity embeddings?

Differentiable mapping of $\boldsymbol{e} \in \mathbb{R}^k$ to $\tilde{\boldsymbol{e}} \in \mathbb{R}^{k,+}$

Options:

$$\tilde{\boldsymbol{e}} := \exp(\boldsymbol{e}) \in \mathbb{R}^{k,+}$$

$$\tilde{\boldsymbol{e}} := \mathrm{ReLU}(\boldsymbol{e}) \in \mathbb{R}^{k,+}$$

$$\tilde{\boldsymbol{e}} := \sigma(\boldsymbol{e}) \in (0,1)^k$$

strongest restriction, but works best!
"Approximately Boolean embeddings"

Ordered relation embeddings?

1 additional "lifted" loss term per implication rule:

minimize $$\mathcal{L}_{\text{rule}} = \sum_i \max\left(0, [\boldsymbol{v}_{\texttt{prof\_at}} - \boldsymbol{v}_{\texttt{prof\_at}}]_i\right)$$

upper bound to "grounded" loss

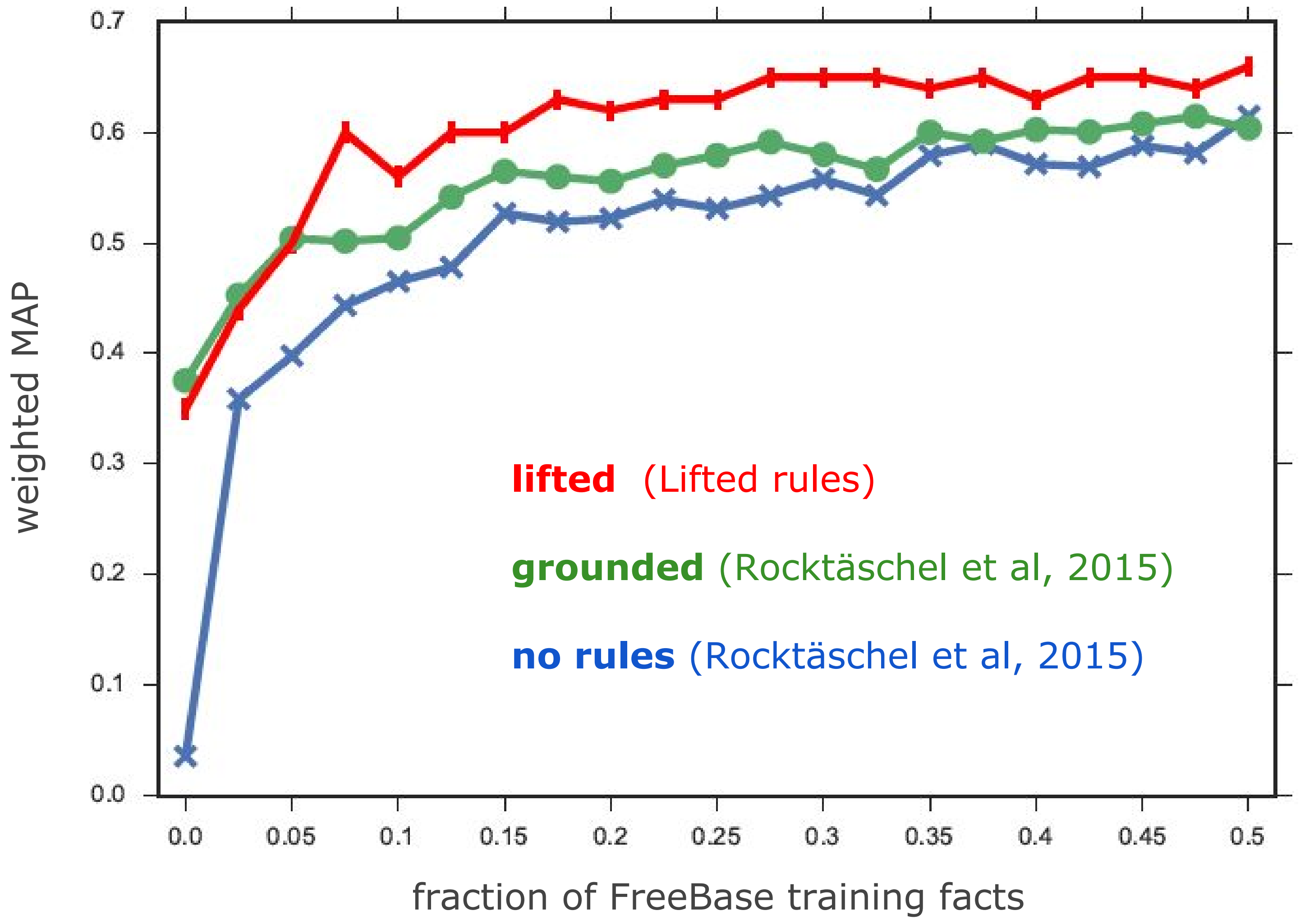
Experiments: Grounded versus Lifted?

More efficient:

	0 rules	36 rules	427 rules
1 epoch (single CPU)	**6.33s**	**6.76**	**6.97s**

only
10% overhead
due to rules

Higher precision:



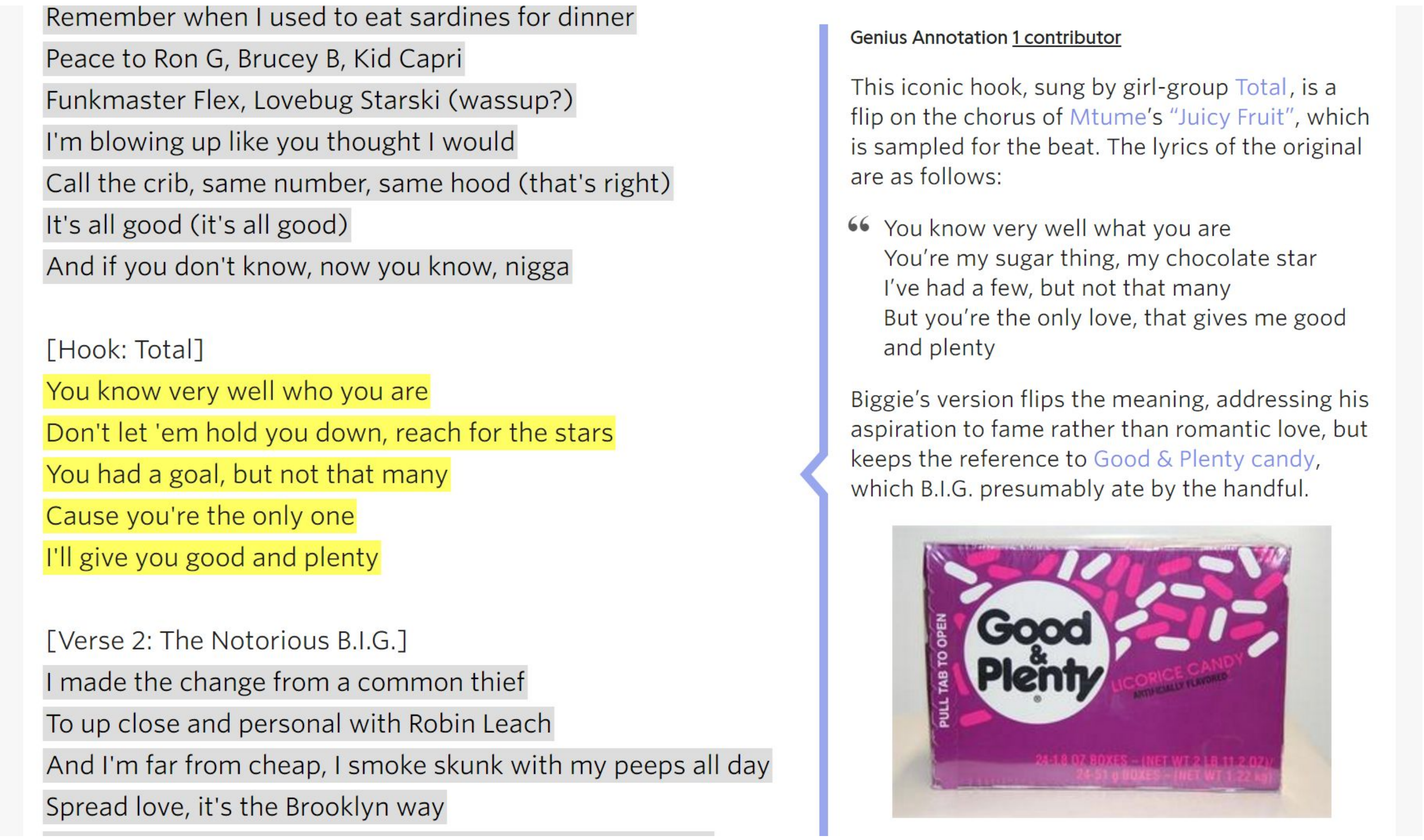
Outline

1. Knowledge Bases
2. Knowledge Base Population
   a. Knowledge Extraction from Text
   b. Link Prediction
3. **Research Topic at Cambridge**

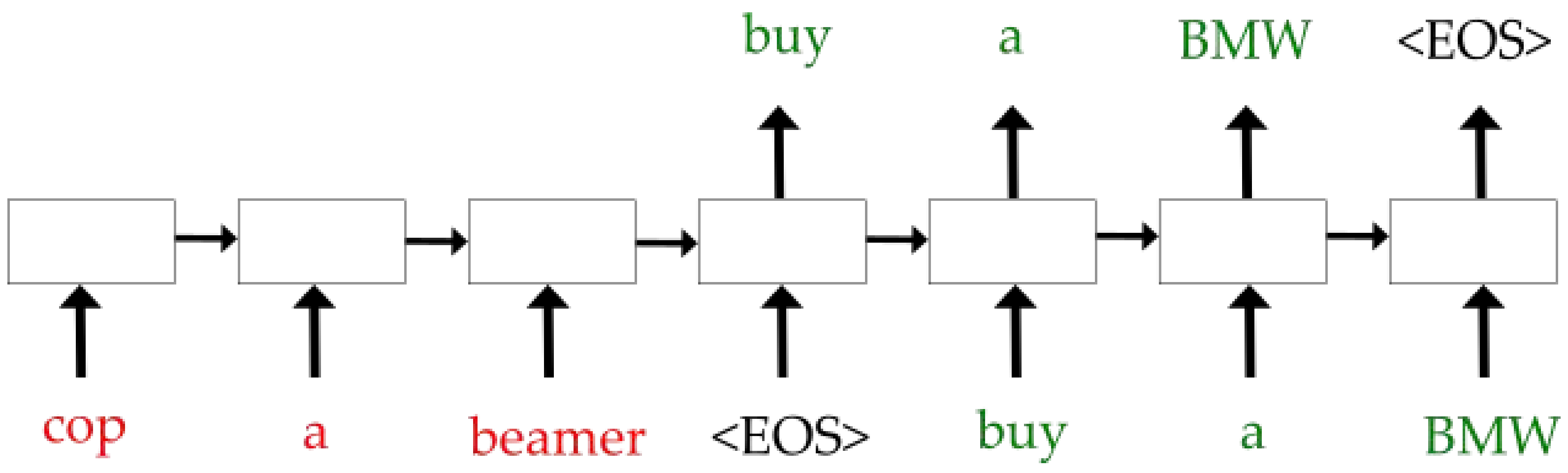
At Cambridge: **Learning to Annotate**

Learn to explain hard-to-interpret text

- Genius.com
- 950.000 lyrics-annotation pairs
- English-to-English text generation
    - (Summarizing, Simplification, Paraphrasing,...)



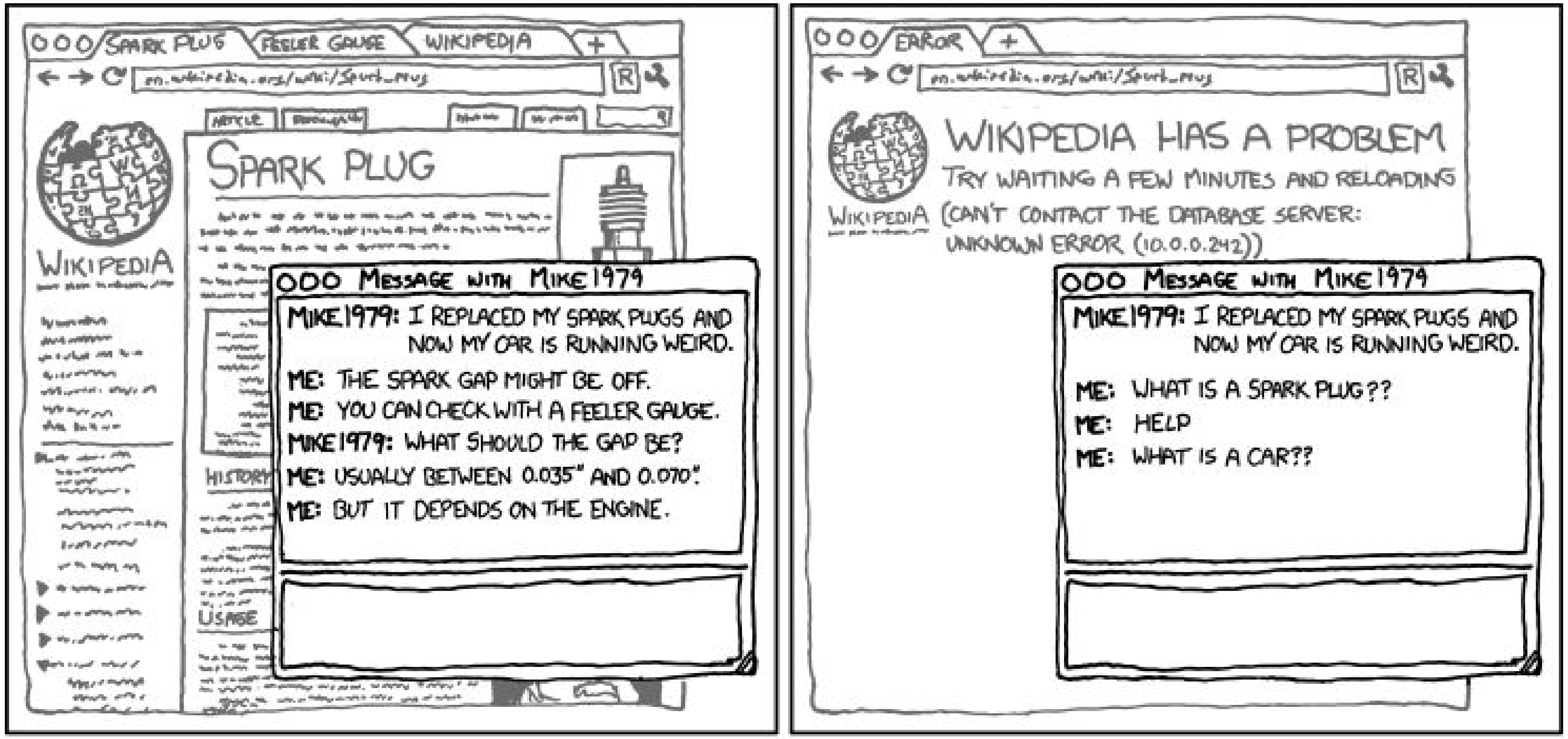
- Statistical and Neural Machine Translation?
- Translation vs. Retrieval based?
- Evaluation ?
- Paraphrasing vs. External Knowledge ?



Learning to Annotate

Lyrics:	`rolly on my arm`
True:	he's always rocking a rolex
Retr.:	(5) he wears a rolex, rolex manufactures expensive wristwatches.
LSTM:	(4) he wears his rolex
Lyrics:	`i blast ya cabbage`
True:	think "head of cabbage". he's gonna shoot you in the head.
Retr.:	(3) brains, post-headshot
LSTM:	(4) he ' s gonna shoot you .
Lyrics:	`i used to have to pack a mack in back of the ac`
True	skits called "packin a mac in the back of the ac" appeared on pun's first two albums
Retr.:	(5) he's "packin a mac [pakinamac] in the back of the ac [meaning acura] vinnie is saying that he'll put you in the back of an accura right next to big pun's mac.
LSTM:	(4) he used to have a gun in the back of the car
Lyrics:	`behold the flyest; bentley drivers louis vuitton buyers, jet fuel abusers driving bentleys,`
True:	buying louis vuitton swag, and burning fuel in private jets are ways to show the cash that biggie and jay-z have to spend. (they are the flyest)
Retr.:	(2) for rich people/ballers only, prestigious (expensive as hell) name brand.
LSTM:	(4) he ' s got a lot of money and expensive cars , and he ' s got a lot of expensive brands .

Thank you ! Questions, Comments?

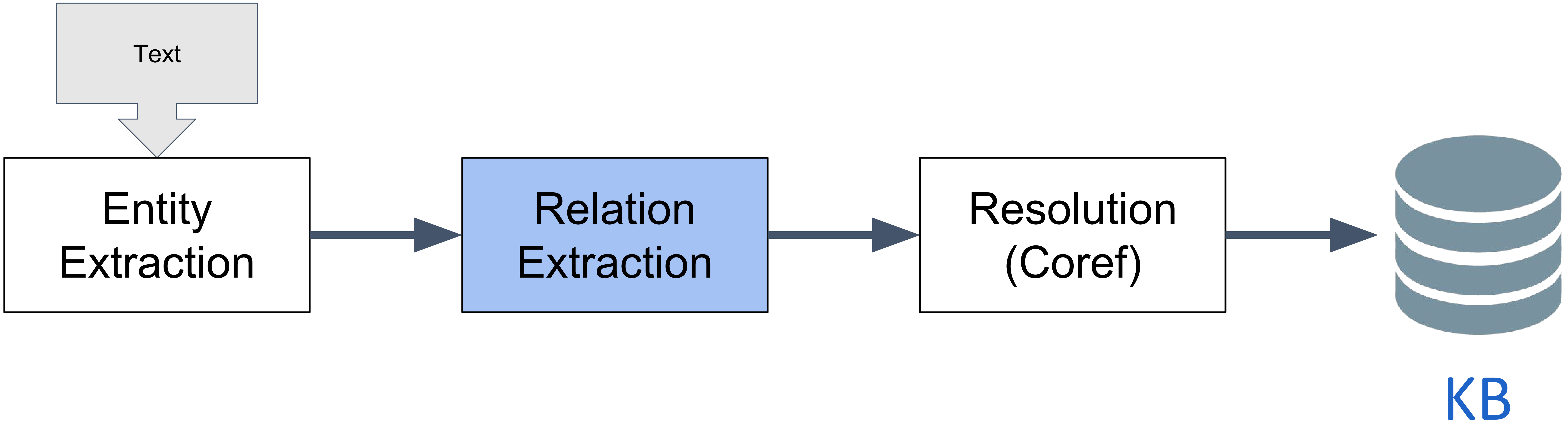


WHEN WIKIPEDIA HAS A SERVER OUTAGE, MY APPARENT IQ DROPS BY ABOUT 30 POINTS.

Xkcd extended miind

Lucas Sterckx
lusterck.github.io
@lusterck

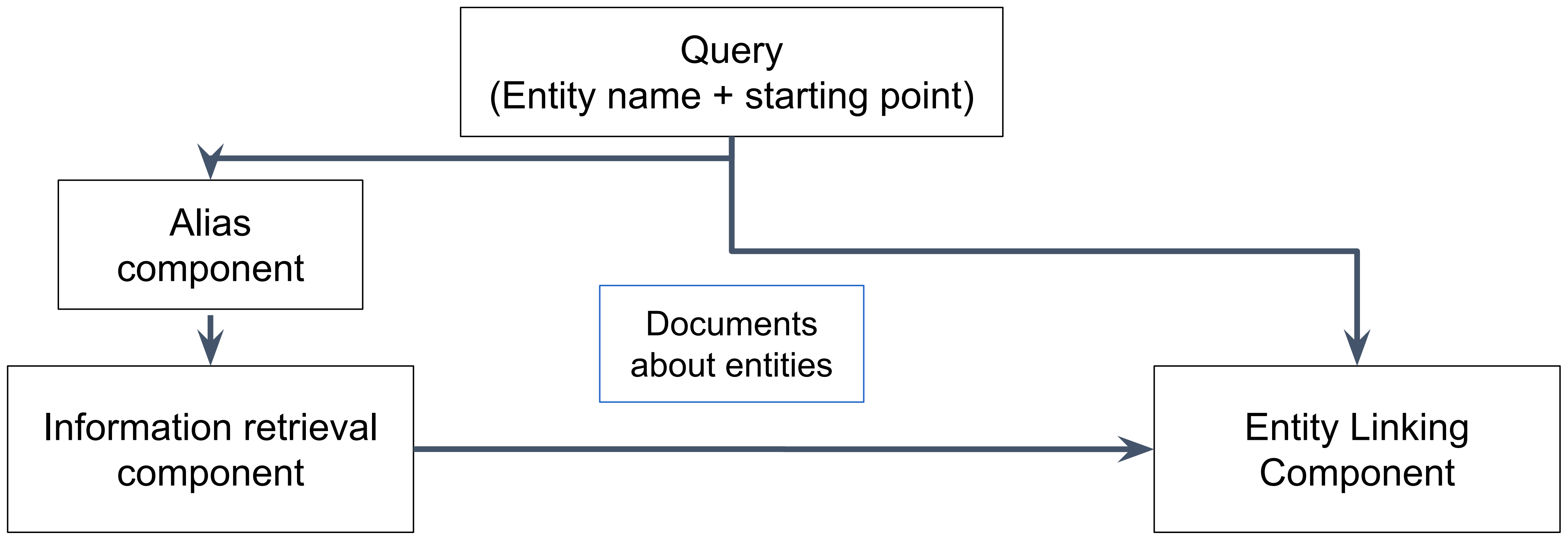
Information Extraction



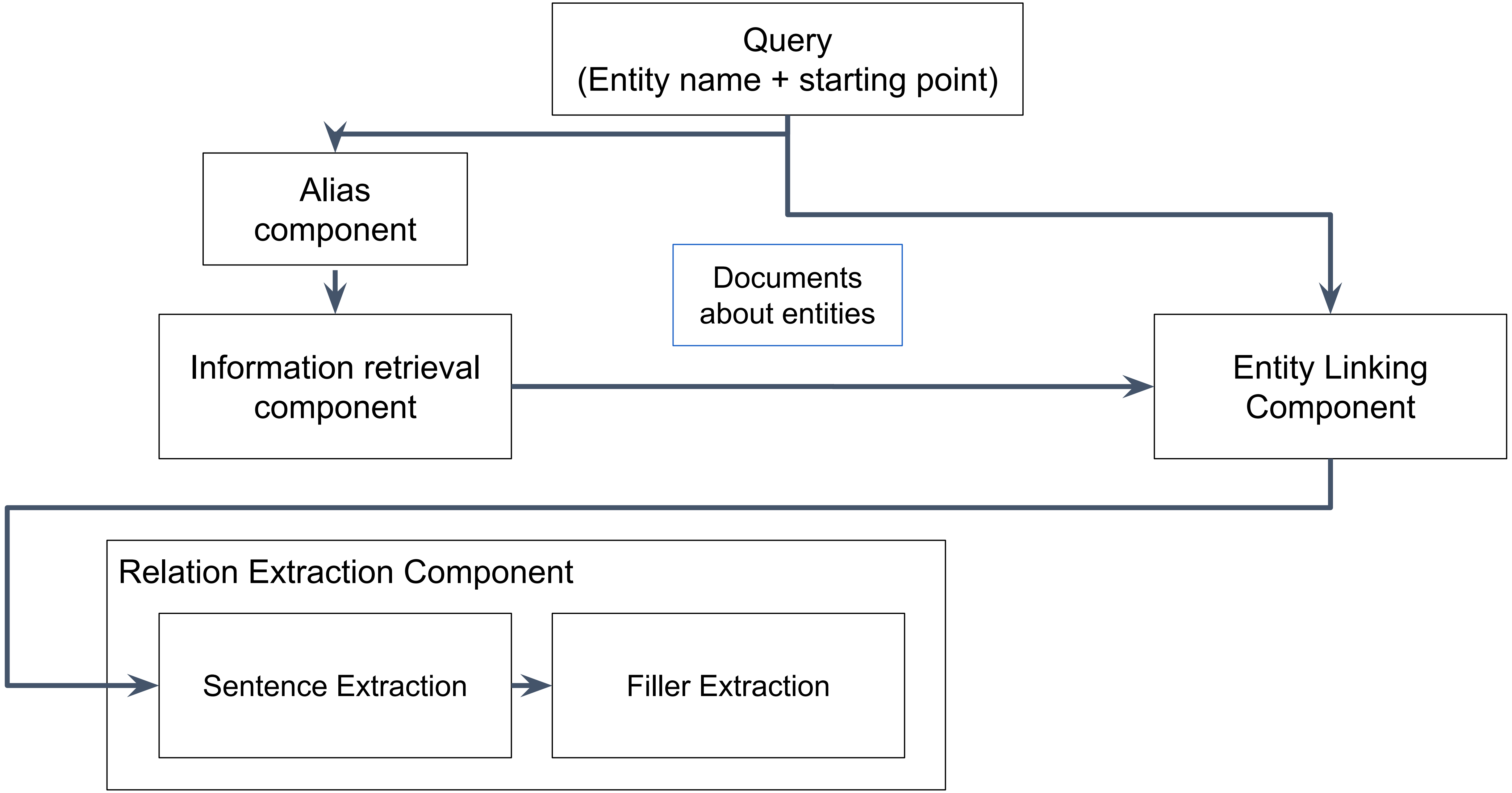
Ghent University at TAC KBP

Query
(Entity name + starting point)

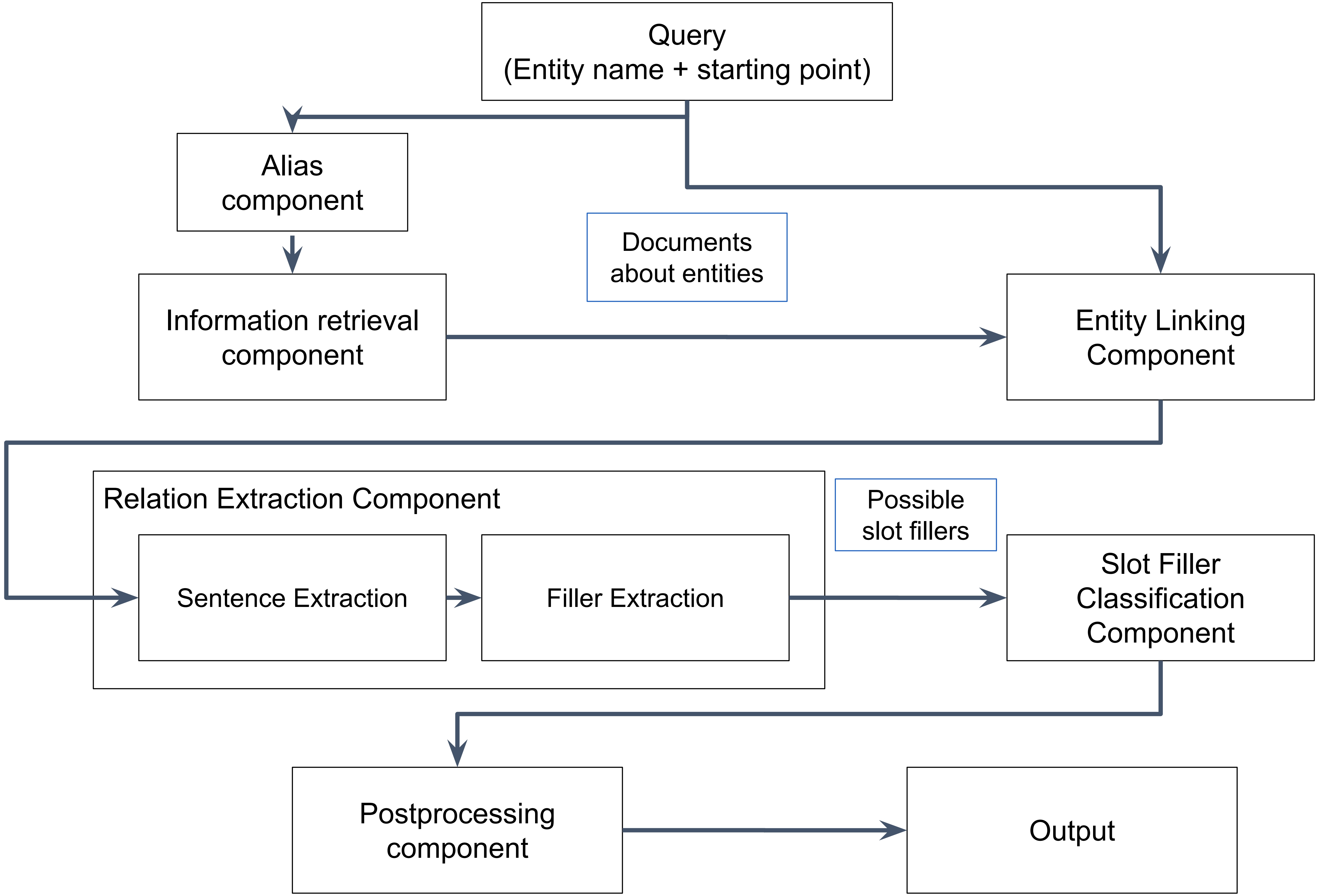
Ghent University at TAC KBP



Ghent University at TAC KBP

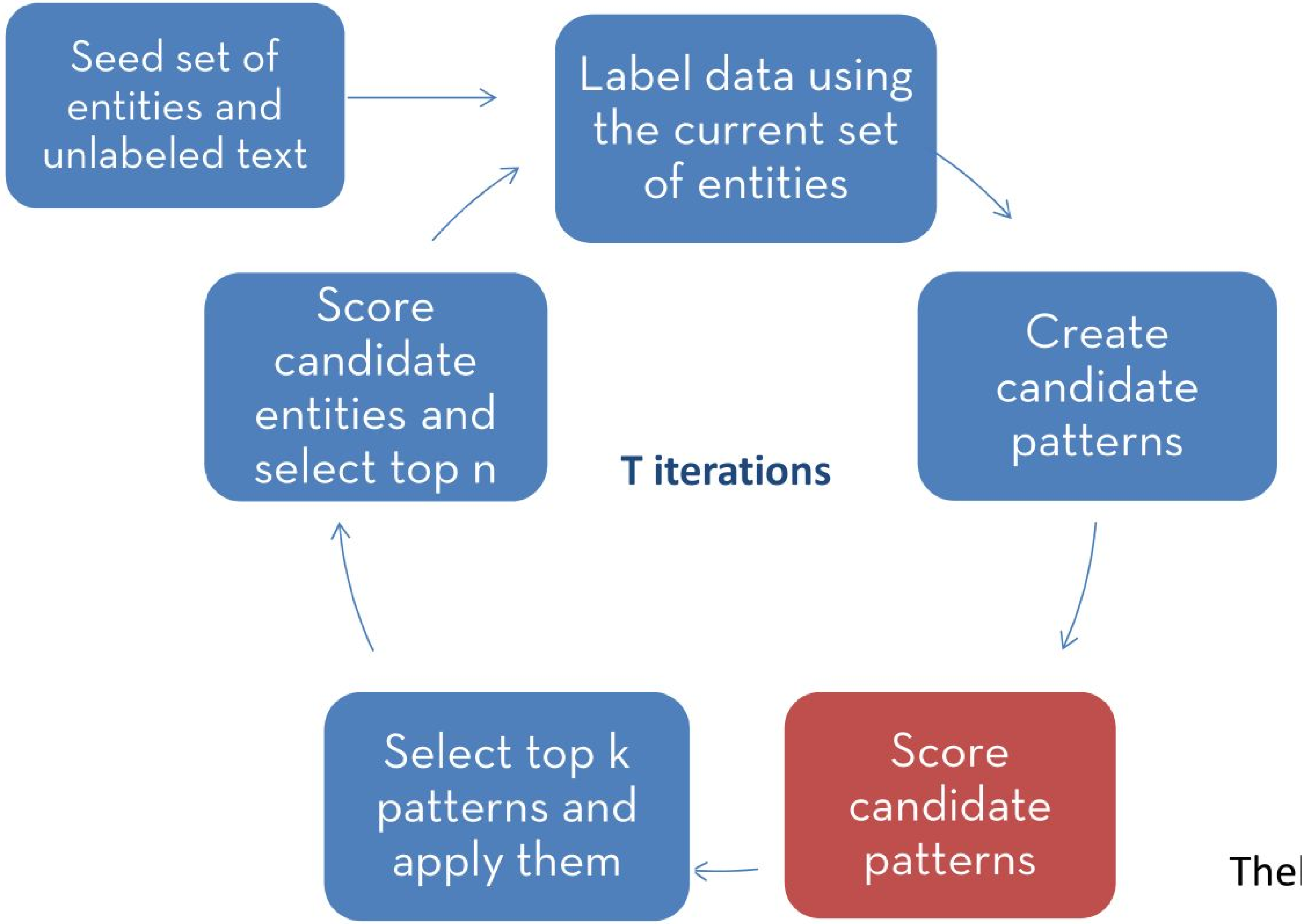


Ghent University at TAC KBP



Semi-Supervised Relation Extraction

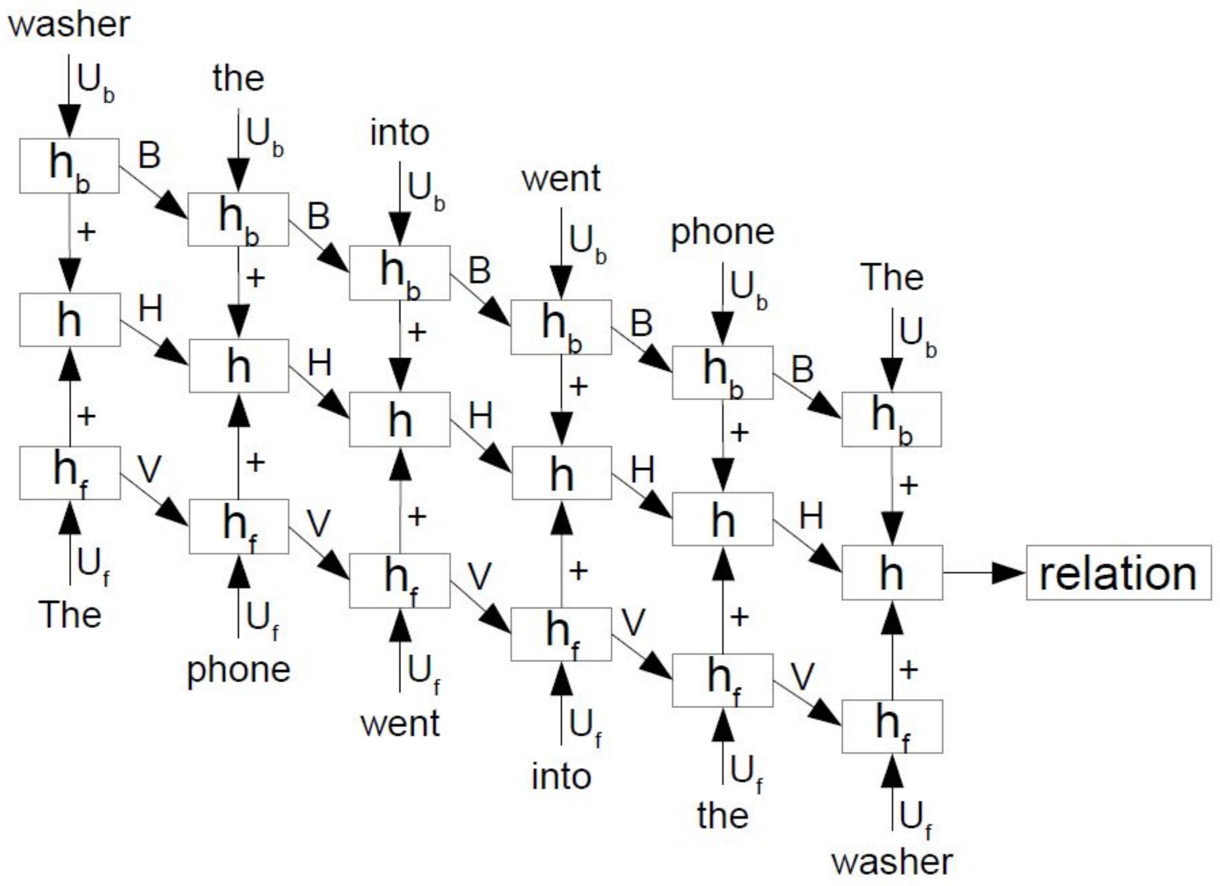
→ **Bootstrapping** (Hearst, DIPRE, Snowball, BRED)

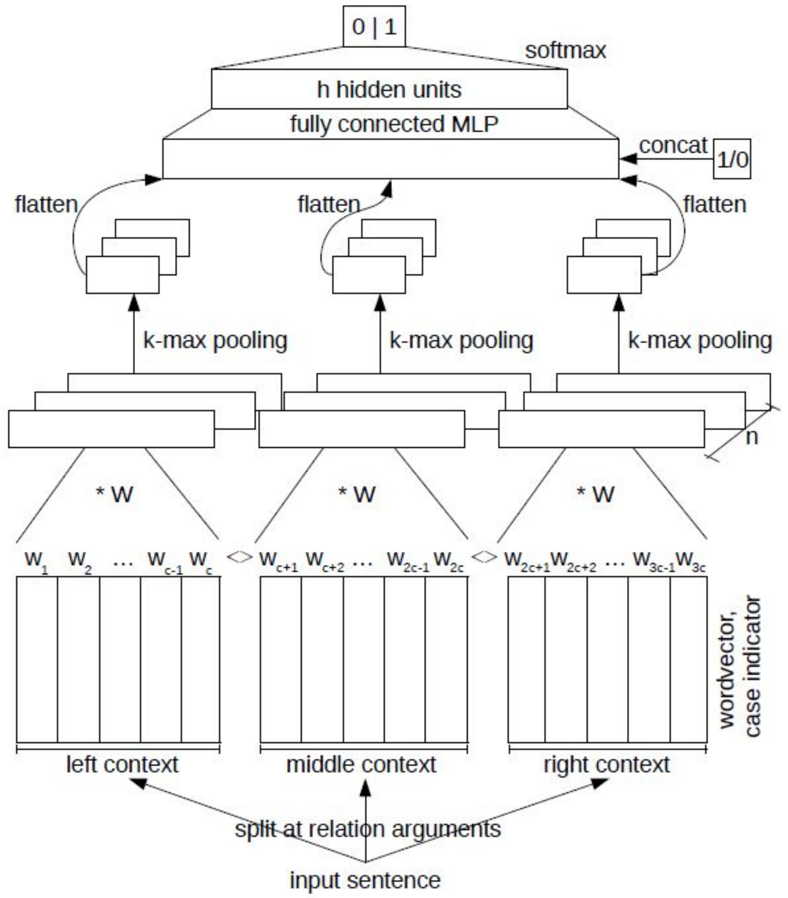


Thelen and Riloff, 2002

Supervised Relation Extraction

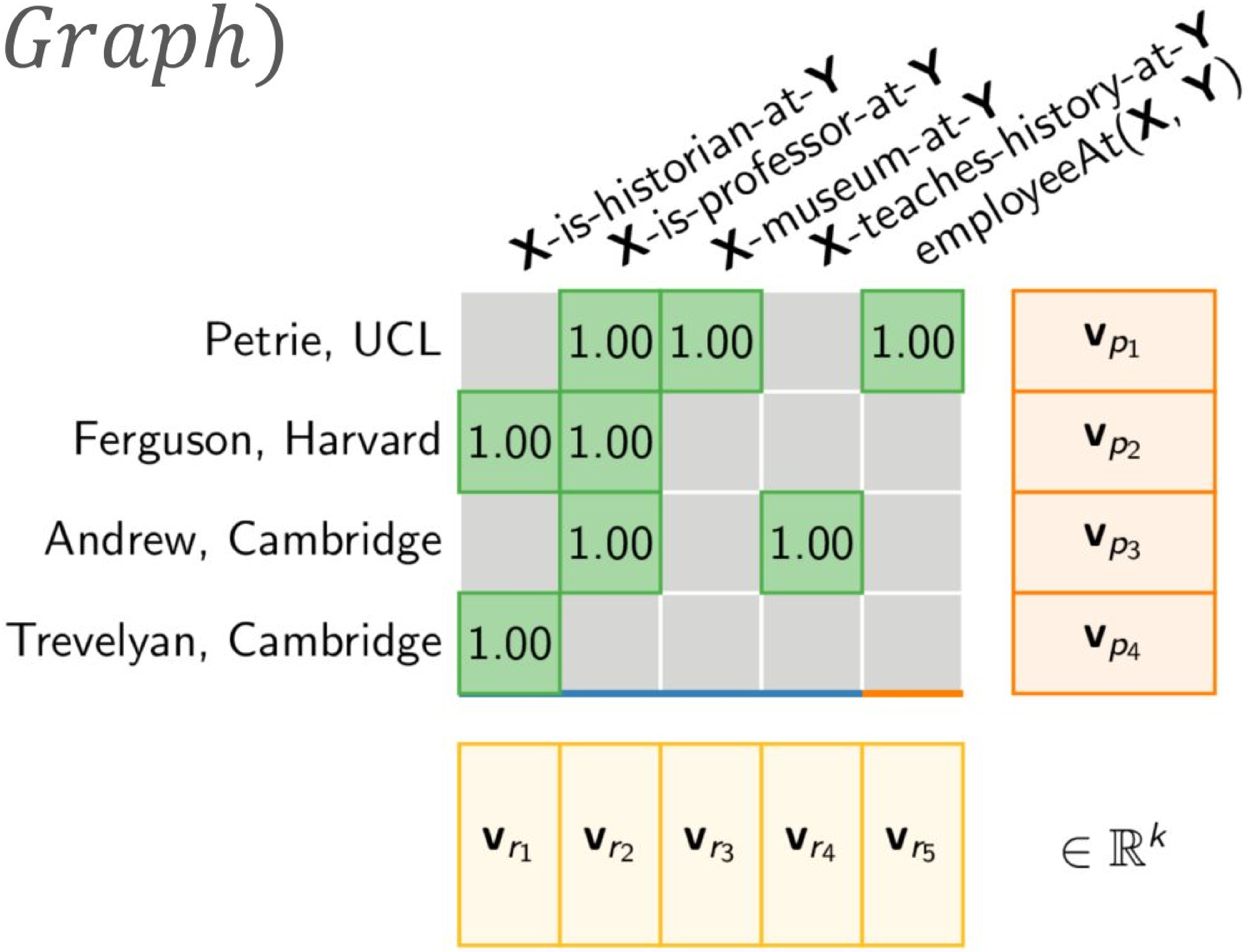
- **LSTM or CNN-based** sentence classifiers





Link Prediction in Knowledge Graphs

$$\Pr(x_{ijk} = 1 \mid Graph)$$

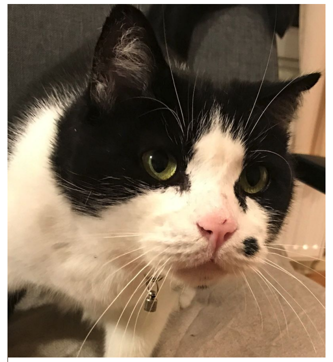


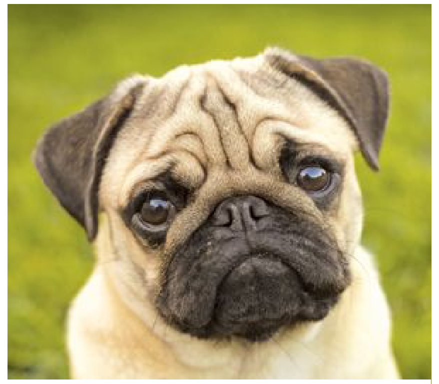
Knowledge Bases enable AI

- “**Knowledge is Power**”
  - “If a program is to perform a complex task well, it must know a great deal about the world in which it operates.”
- **Graphs** can be processed **efficiently** and offer a convenient abstraction
- **Enabling technology** for:
  - Machine reasoning
  - Disambiguation in written and spoken data
  - Semantic search in terms of entities & relations (not keywords & pages)

Knowledge representations


Whisky

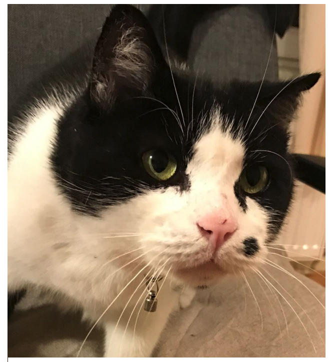

Tarzan

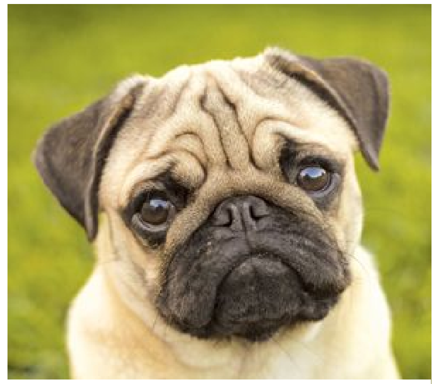

Snoopy

	is_cat	is_dog	is_animal
Whisky	1	0	1
Tarzan	1	0	1
Snoopy	0	1	1

Knowledge representations


Whisky


Tarzan


Snoopy

neural | symbolic

low-dimensional representations $v_{\texttt{Whisky}},\ v_{\texttt{is\_cat}},\ v_{\texttt{is\_animal}} \in \mathbb{R}^k$

- can capture similarity / hierarchy $v_{\texttt{Whisky}} \approx v_{\texttt{Tarzan}}$
- can be trained from raw facts $p(\texttt{fact}) := \sigma\left(v_{\texttt{predicate}} \cdot v_{\texttt{entity}}\right)$

$$\sigma\left(v_{\texttt{is\_cat}} \cdot v_{\texttt{Tarzan}}\right) \approx 1$$
$$\sigma\left(v_{\texttt{is\_cat}} \cdot v_{\texttt{Whisky}}\right) \approx 1$$

- Difficult to incorporate prior knowledge!

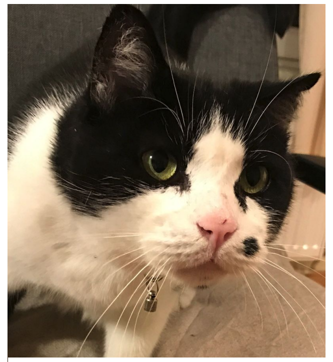
"all cats are animals"

$$\sigma\left(v_{\texttt{is\_animal}} \cdot v_{\texttt{Tarzan}}\right) \approx 1$$
$$\sigma\left(v_{\texttt{is\_animal}} \cdot v_{\texttt{Whisky}}\right) \approx 1$$

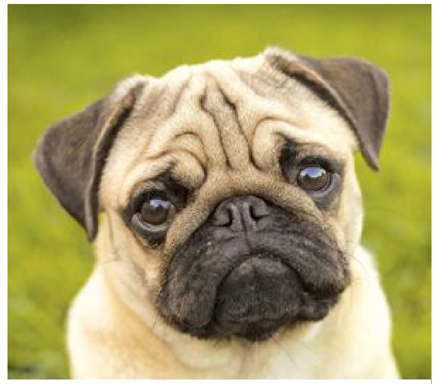
grounded in
all entities!

Knowledge representations

Whisky



Tarzan



Snoopy

neural | **symbolic**

symbols as representations (e.g. entities, predicates)

easy to integrate domain knowledge: add rules

- powerful logic reasoning tools (prolog)

"lifted" formulation

```
is_cat(Tarzan).                 % Tarzan is a cat

is_animal(X) :- is_cat(X)       % rule: all cats are animals

?- is_animal(Tarzan)            % is Tarzan an animal?
   yes
```

- No notion of similarity (Whisky ≠ Tarzan)

```
?- is_animal(Whisky)
   no idea :(
```

not suited for approximate inference

Guiding Bootstrapped Relation Extractors