Semi-Supervised Bootstrapping Relationship Extractors with Distributional Semantics

David Soares Batista

Berlin, 2nd July 2017

PyData Berlin 2017

Semantic Relationships

Noam Chomsky was born in the East Oak Lane neighbourhood of Philadelphia, Pennsylvania. In 1955, Chomsky become an assistant professor at The Massachusetts Institute of Technology (MIT), a private research university in Cambridge, Massachusetts.

Buzz Aldrin earned a Doctor of Science degree in Astronautics from Massachusetts Institute of Technology.

Barack Obama graduate with a JD degree magna cum laude from Harvard University, a private research university in Cambridge, Massachusetts.

Semantic Relationships

Noam Chomsky was born in the **East Oak Lane** neighbourhood of **Philadelphia**, **Pennsylvania**. In 1955, **Chomsky** become an assistant professor at **The Massachusetts Institute of Technology (MIT)**, a private research university in **Cambridge**, **Massachusetts**.

Buzz Aldrin earned a Doctor of Science degree in Astronautics from **Massachusetts Institute of Technology**.

Barack Obama graduate with a JD degree magna cum laude from **Harvard University**, a private research university in **Cambridge, Massachusetts**.

Semantic Relationships

<Noam Chomsky, *born-in*, East Oak Lane>

<East Oak Lane, *part-of*, Philadelphia>

<Philadelphia, *part-of*, Pennsylvania>

<Chomsky, *affiliated-with*, MIT>

<MIT, *located-in*, Cambridge>

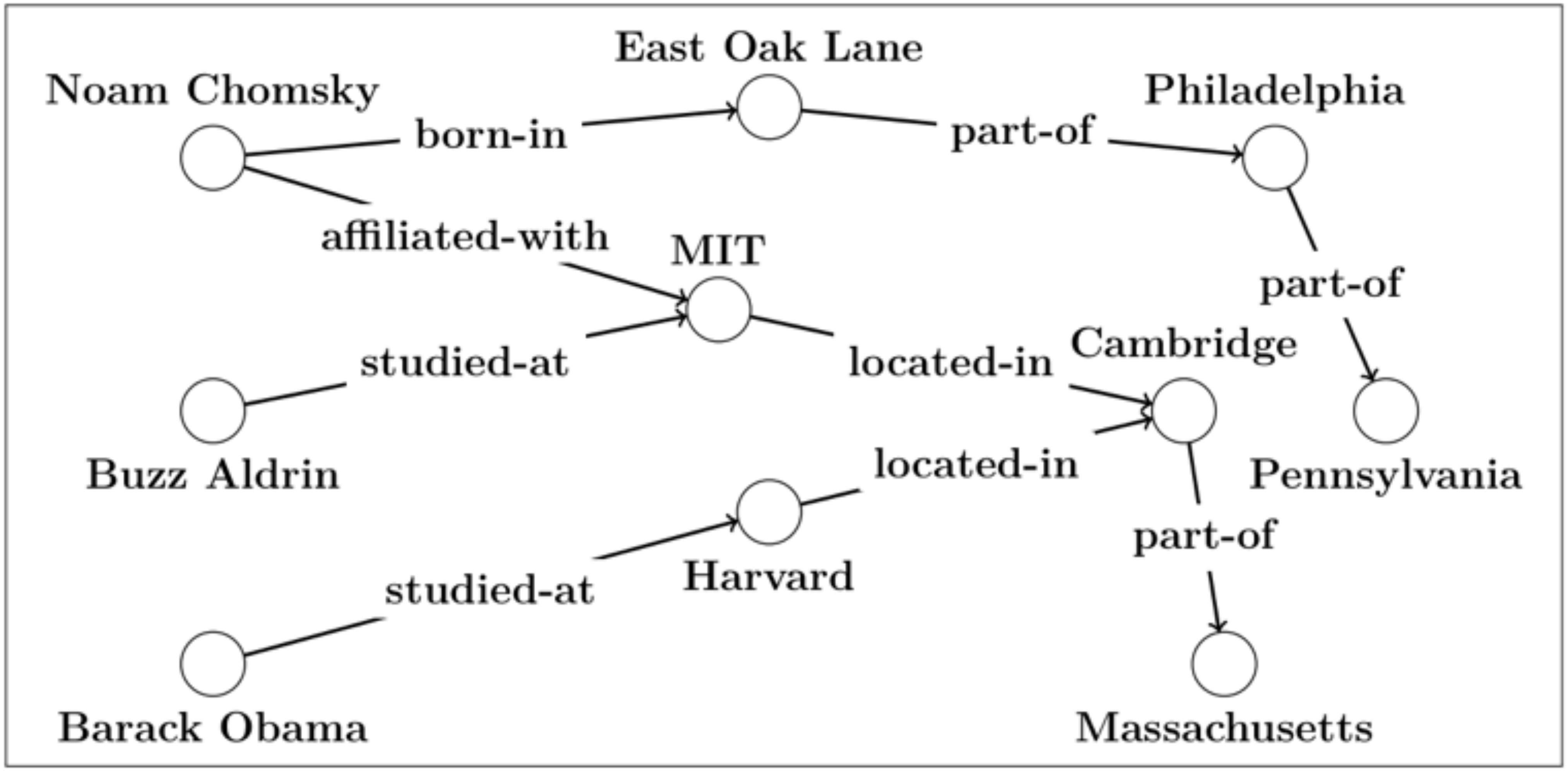
<Cambridge, *part-of*, Massachusetts>

<Buzz Aldrin, *studied-at*, MIT>

<Barack Obama, *studied-at*, Harvard>

<Harvard, *located-in*, Cambridge>

Knowledge Graphs for Question Answering



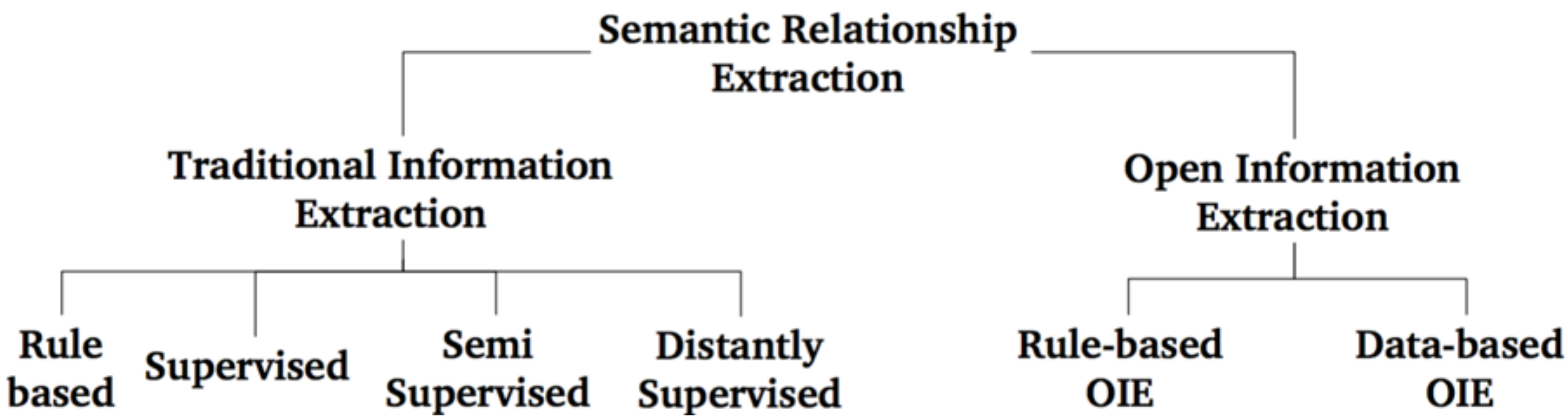
Who studied in Cambridge, Massachusetts ?

Which universities are located in Massachusetts ?

Outline

1. Approaches for Semantic Relationship Extraction
2. Semi-Supervised/Bootstrapping
3. Snowball: TF-IDF
4. BREDS: Word Embeddings
5. Experimental Evaluation

Approaches for Relationship Extraction



- **Traditional Information Extraction**: precise and pre-specified relationships
- **Open IE**: extracts all possible relationships with no pre-specified types

Approaches for Relationship Extraction

- **Rule-based:**
  - high precision/low recall
  - hand-made rules hard to maintain
- **Supervised**
  - need training data
  - types of relationships is limited
- **Bootstrapping / Semi-supervised**
  - takes advantage of unlabelled data
  - needs to handle semantic drift
- **Distantly-supervised**
  - generates loads of training data
  - how to filter out noisy sentences ?

Outline

1. ~~Approaches for Semantic Relationship Extraction~~
2. Semi-Supervised/Bootstrapping
3. Snowball: TF-IDF
4. BREDS: Word Embeddings
5. Experimental Evaluation

Bootstrapping

- Unlabelled data is vast and abundant, bootstrapping approaches leverage on such data
- Use just a few seed instances of known relationships, e.g.: company headquarters

Seeds
<Google, Mountain View>
<IKEA, Leiden>
<Soundcloud, Berlin>

Document Collection

Output
<Porsche, Stuttgart>
<Capcom, Osaka>
<Nokia, Espoo>
<AT&T, Dallas>
<BMW, Munich>
<Siemens, Munich>

- Rely only on seed instances and contextual similarity

similarity

"**Google** is *headquartered in* **Mountain View**"

"**Soundcloud** HQ *in* **Berlin**"

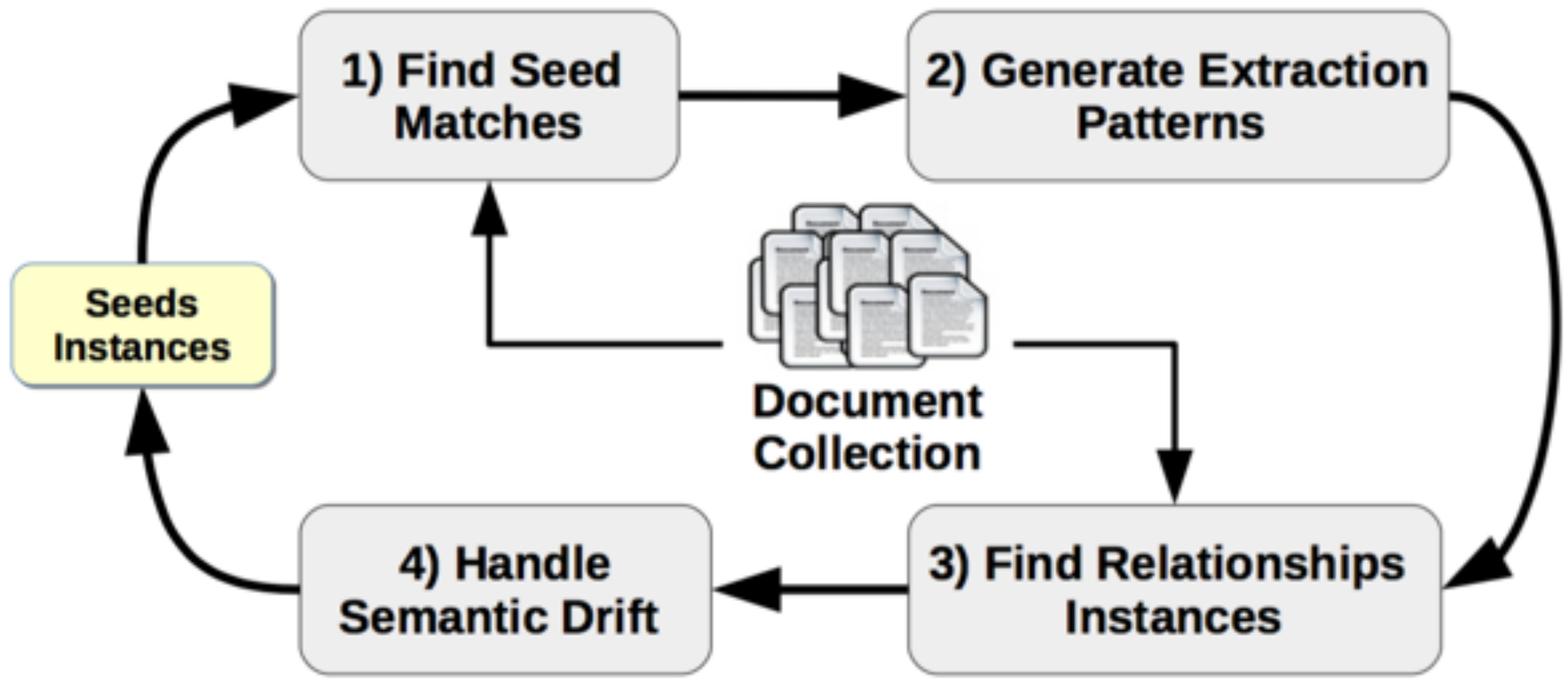
"**Nokia** base campus in **Espoo**"

"**BMW** *main offices in* **Munich**"

"**AT&T** *based in* **Dallas**"

"**Porsche** *main headquarters in* **Stuttgart**"

General Architecture for Bootstrapping



1. Collect occurrence contexts for the seed instances.
2. Based on these contexts, generate extraction patterns.
3. Scan the documents using the patterns to match new relationship instances.
4. Newly extracted instances are then added to the seed set, and the process is repeated until a certain stop criteria is met.

Bootstrapping for Semantic Relationship Extraction

Semantic lexicon acquisition: (late 90s)

- extract concepts or terms and the associated semantic class
- particular case of relationship extraction (i.e., is-a relationships)
- e.g.: biomedical categories for terms found in biomedical journals/papers

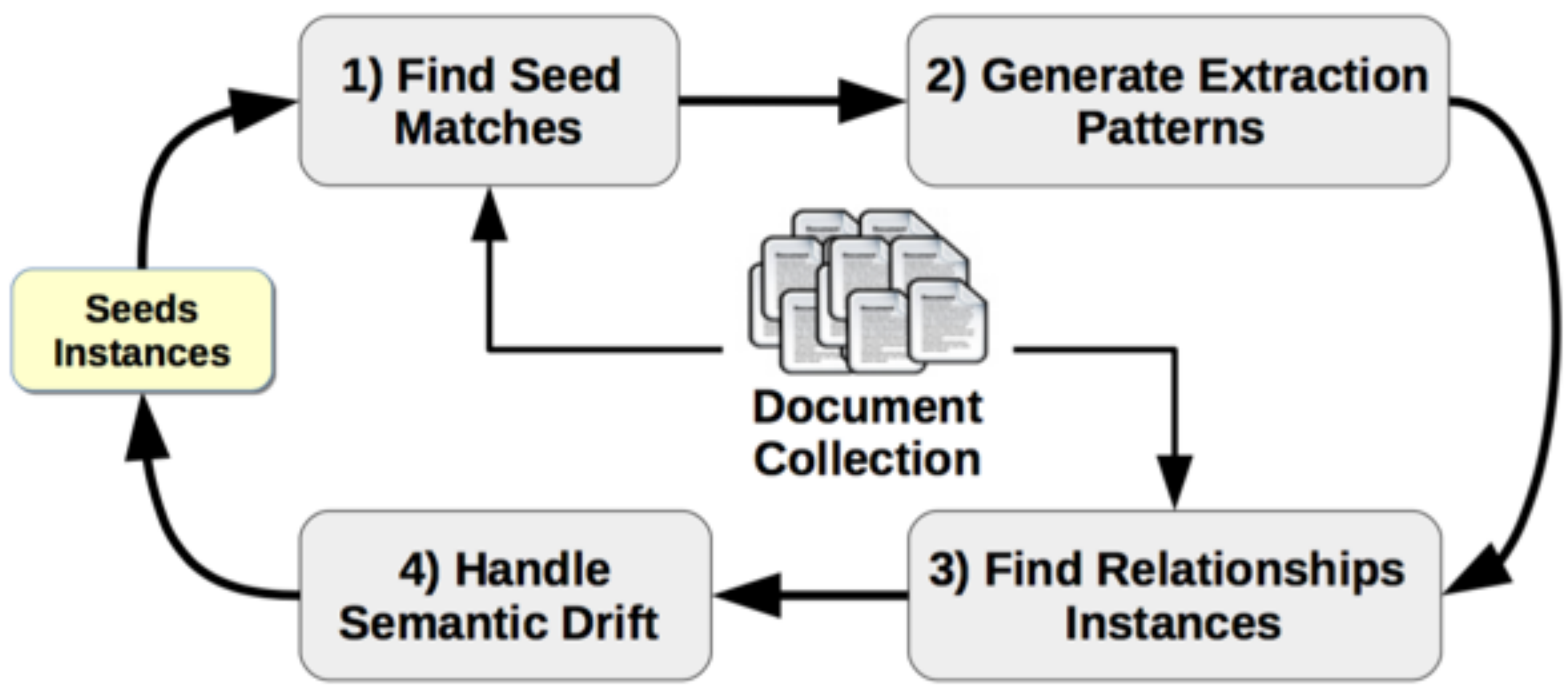
Semantic Relationship Extraction

- DIPRE: Dual Iterative Pattern Relation Expansion, (Brin, 1999)
- Snowball: Extracting Relations from Large Plain-Text Collections (Agichtein and Gravano, 2000; Yu and Agichtein, 2003)
- Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations (Pantel and Pennacchiotti, 2006)

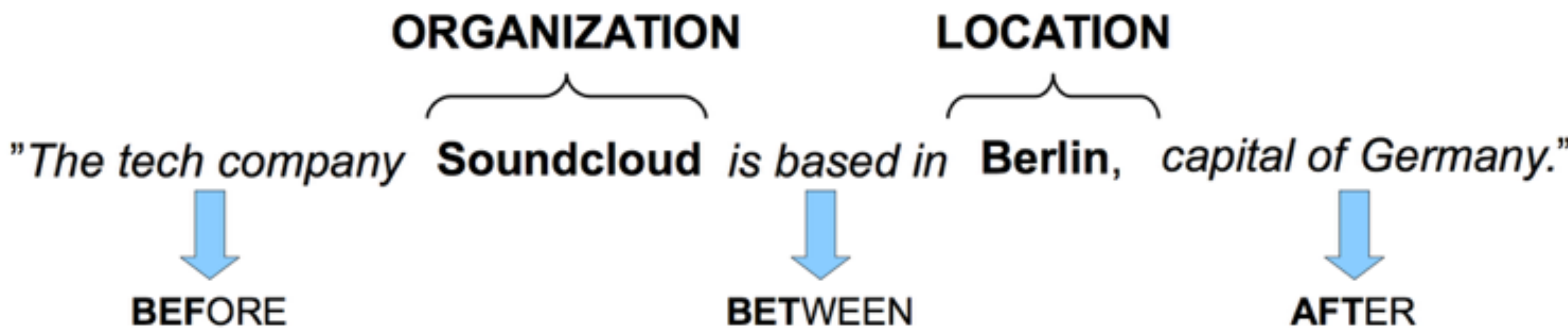
Outline

1. ~~Approaches for Semantic Relationship Extraction~~
2. ~~Semi-Supervised/Bootstrapping~~
3. Snowball: TF-IDF
4. BREDS: Word Embeddings
5. Experimental Evaluation

Snowball


1) Find Seed Matches
2) Generate Extraction Patterns
Seeds Instances
Document Collection
4) Handle Semantic Drift
3) Find Relationships Instances

Snowball: Find Seed Matches



Build a TF-IDF vector for each context

"The tech company"	BEF =	0	0	2.3	0	0	1.1	0	0	0	0	0
"is based in"	BET =	0	0	0	3.3	0	0	0	3.3	0	1.1	0
"capital of Germany"	AFT =	0	0	0	0	2.5	0	0	3.3	0	0	0

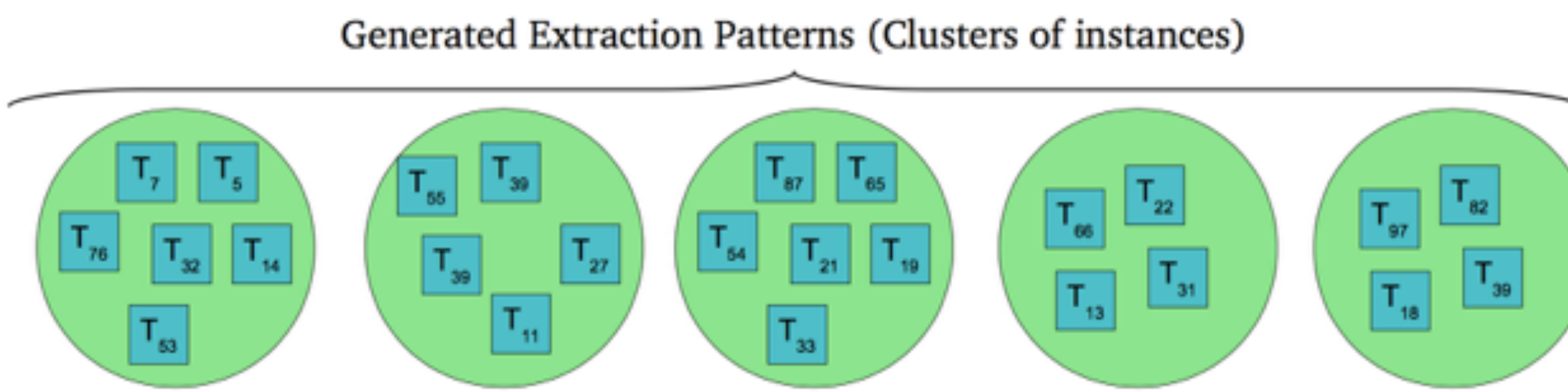
$$T_n = < BEF,\ e1_{ORG},\ BET,\ e2_{LOC},\ AFT >$$

Snowball: Generating Extraction Patterns

- Single-pass clustering over all the collected tuples

$$Sim(T_i, T_j) = \alpha \cdot cos(BEF_i, BEF_j) + \beta \cdot cos(BET_i, BET_j) + \gamma \cdot cos(AFT_i, AFT_j)$$

- Similarity threshold: τ_{sim}



- Compute mean for each context (BEF, BET, AFT) of all vectors in a cluster

$$< \overline{BEF},\ e1_{ORG},\ \overline{BET},\ e1_{LOC},\ \overline{AFT} >$$

Finding new relationships instances

- Collect text segments containing entity pairs whose semantic types match the seeds

<**Google, Mountain View**>

<**Soundcloud, Berlin**>

Document Collection

[..text..] **ORG** [..text..] **LOC** [..text..]

BEF **BET** **AFT**

- Compute similarity of each with extraction patterns (centroids)

$$Sim(T_i, T_j) = \alpha \cdot cos(BEF_i, BEF_j) + \beta \cdot cos(BET_i, BET_j) + \gamma \cdot cos(AFT_i, AFT_j)$$

- Extract new instance if above threshold τ_{sim}

Finding new relationships instances

- **Missing related matches due to TF-IDF limitation**

X = “main headquarters in”	0	0	2.3	0	0	1.1	0	0	0	0	0
Y = “is based in”	0	0	0	3.3	0	0	0	3.1	0	0	0
X = “has offices in”	0	1.1	0	0	2.5	0	0	0	0	0	0

$$\text{cos\_sim}(X, Y) = 0$$
$$\text{cos\_sim}(X, Z) = 0$$
$$\text{cos\_sim}(Y, Z) = 0$$

- **Unless there is a common dimension cosine similarity will always be 0**

Word Embeddings

- **Distributional semantics:** based on co-occurrence contexts

"headquarters"	0.18	0.22	0.82	0.65	0.33	0.23
"based"	0.16	0.76	0.81	0.63	0.31	0.33
"headquartered"	0.22	0.81	0.81	0.64	0.36	0.33

cos_sim("headquarters", "based") = 0.76
cos_sim("based", "headquartered") = 0.70
cos_sim("headquarters", "headquartered") = 0.80

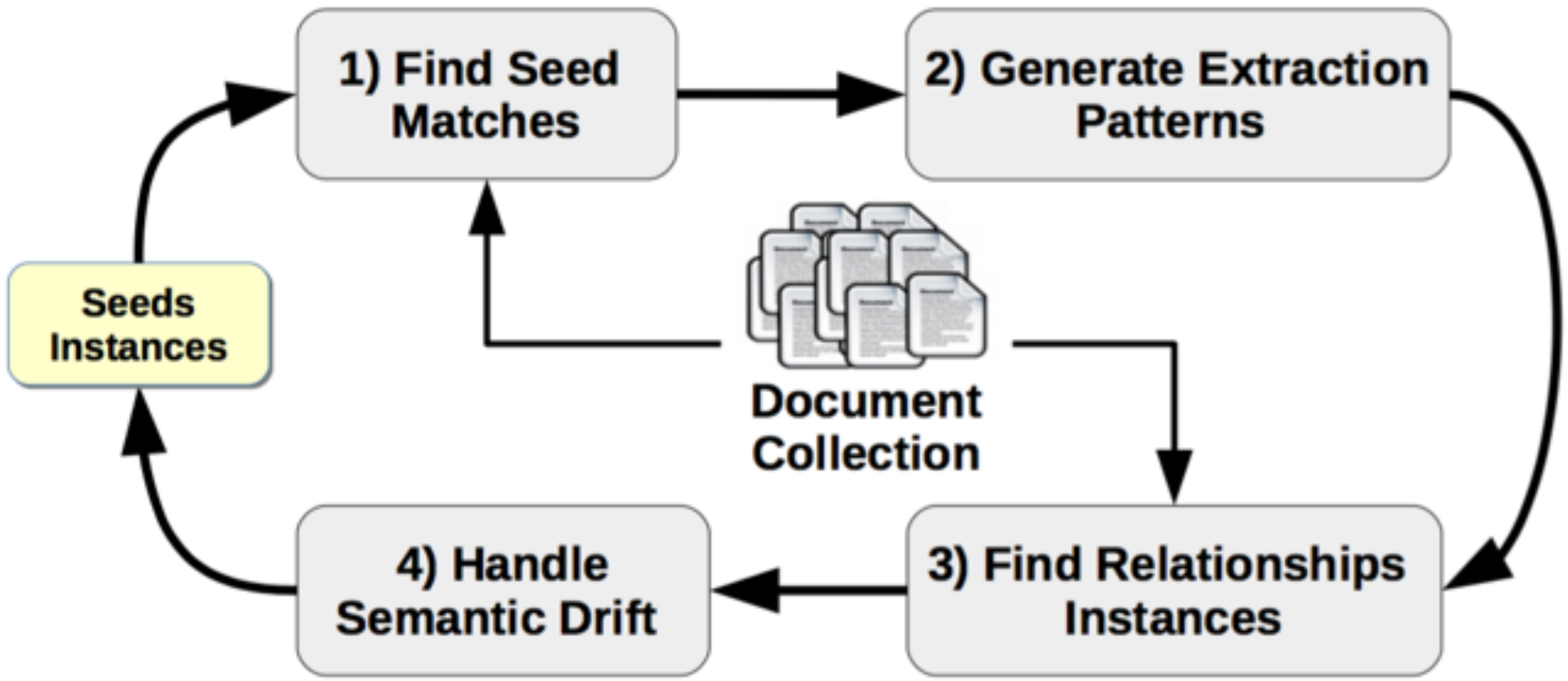
- Snowball architecture expects a single vector per context.

- How to represent each context as a single vector?

Outline

1. ~~Approaches for Semantic Relationship Extraction~~
2. ~~Semi-Supervised/Bootstrapping~~
3. ~~Snowball: TF-IDF~~
4. BREDS: Word Embeddings
5. Experimental Evaluation

BREDS: Bootstrapping Relationship Instances with Distributional Semantics



keep te same architecture

BREDS: Find Seed Matches

Try to find in **BET** context:

- a verb (e.g., invented)
- a verb followed by a preposition (e.g., located in)
- a verb followed by nouns, adjectives, or adverbs ending in a preposition (e.g., has atomic weight of)

$$V \mid VP \mid VW^*P$$
$$V = \text{verb particle? adv?}$$
$$W = (\text{noun} \mid \text{adj} \mid \text{adv} \mid \text{pron} \mid \text{det})$$
$$P = (\text{prep} \mid \text{particle} \mid \text{inf. marker})$$

ReVerb (Fader et al. 2011)

"**Soundcloud** online audio platform is based in **Berlin**, Germany"

BREDS: Find Seed Matches

Try to find in **BET** context:

- a verb (e.g., invented)
- a verb followed by a preposition (e.g., located in)
- a verb followed by nouns, adjectives, or adverbs ending in a preposition (e.g., has atomic weight of)

$$V \mid VP \mid VW^*P$$
V = verb particle? adv?
W = (noun | adj | adv | pron | det)
P = (prep | particle | inf. marker)

ReVerb (Fader et al. 2011)

"**Soundcloud** online audio platform **is based in** **Berlin**, Germany"

BREDS: Find Seed Matches

$$V \mid VP \mid VW^*P$$

$$V = \text{verb particle? adv?}$$

$$W = (\text{noun} \mid \text{adj} \mid \text{adv} \mid \text{pron} \mid \text{det})$$

$$P = (\text{prep} \mid \text{particle} \mid \text{inf. marker})$$

ReVerb (Fader et al. 2011)

Try to find in **BET** context:

- a verb (e.g., invented)
- a verb followed by a preposition (e.g., located in)
- a verb followed by nouns, adjectives, or adverbs ending in a preposition (e.g., has atomic weight of)

“**Soundcloud** online audio platform **is based in** **Berlin**, Germany”

“Today, **John Flower**, the new CEO of **Coffee Inc.**, announced that ...”

BREDS: Find Seed Matches

Try to find in **BET** context:

- a verb (e.g., invented)
- a verb followed by a preposition (e.g., located in)
- a verb followed by nouns, adjectives, or adverbs ending in a preposition (e.g., has atomic weight of)

$$V \mid VP \mid VW^{*}P$$
$$V = \text{verb particle? adv?}$$
$$W = (\text{noun} \mid \text{adj} \mid \text{adv} \mid \text{pron} \mid \text{det})$$
$$P = (\text{prep} \mid \text{particle} \mid \text{inf. marker})$$

ReVerb (Fader et al. 2011)

"**Soundcloud** online audio platform **is based in** **Berlin**, Germany"

"Today, **John Flower** **the new CEO of** **Coffee Inc.**, announced that ..."

BREDS: Find Seed Matches

Transform each context into a single embedding vector:

- Removes stop-words and adjectives
- Sum the embeddings of each word

BREDS: Find Seed Matches

Transform each context into a single embedding vector:

- Removes stop-words and adjectives
- Sum the embeddings of each word

<NULL, **Soundcloud**, “is based in”, **Berlin**, “Germany”>

<Today, **John Flower,** “the new CEO of”, **Coffee Inc.**, “announced that”>

BREDS: Find Seed Matches

Transform each context into a single embedding vector:

- Removes stop-words and adjectives
- Sum the embeddings of each word

<NULL, **Soundcloud**, "~~is~~ based ~~in~~", **Berlin**, "Germany">

<Today, **John Flower,** "~~the~~ ~~new~~ CEO ~~of~~", **Coffee Inc.**, "announced that">

BREDS: Find Seed Matches

Transform each context into a single embedding vector:

- Removes stop-words and adjectives
- Sum the embeddings of each word

<NULL, **Soundcloud**, “based”, **Berlin**, “Germany”>

<Today, **John Flower**, “CEO”, **Coffee Inc.**, “announced that”>

BREDS: Find Seed Matches

Transform each context into a single embedding vector:

- Removes stop-words and adjectives
- Sum the embeddings of each word

<NULL, **Soundcloud**, “based”, **Berlin**, “Germany”>

<Today, **John Flower**, “CEO”, **Coffee Inc.**, “announced that”>

BEF = NULL

BET = E(“based”)

AFT = E(“Germany”)

BEF = E(“Today”)

BET = E(“CEO”)

AFT = E(“announced) + E(“that”)

$$T_n = < BEF,\ e1_{ORG},\ BET,\ e2_{LOC},\ AFT >$$

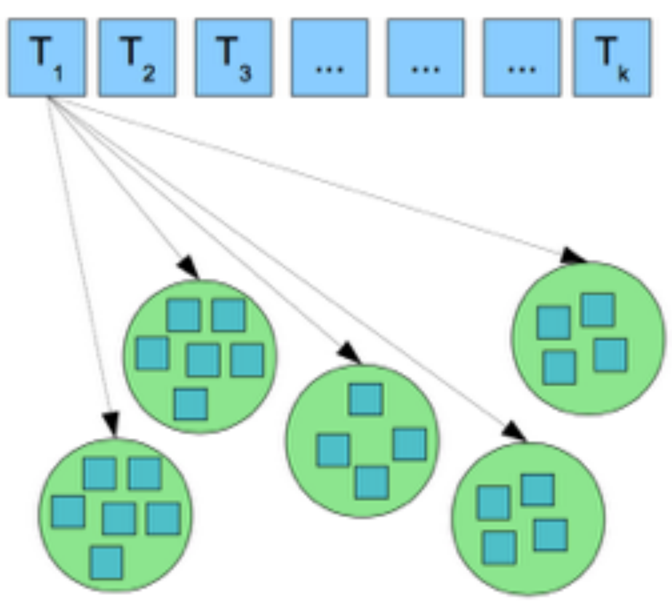
BREDS: Generating Extraction Patterns

Single-pass clustering over all the collected tuples

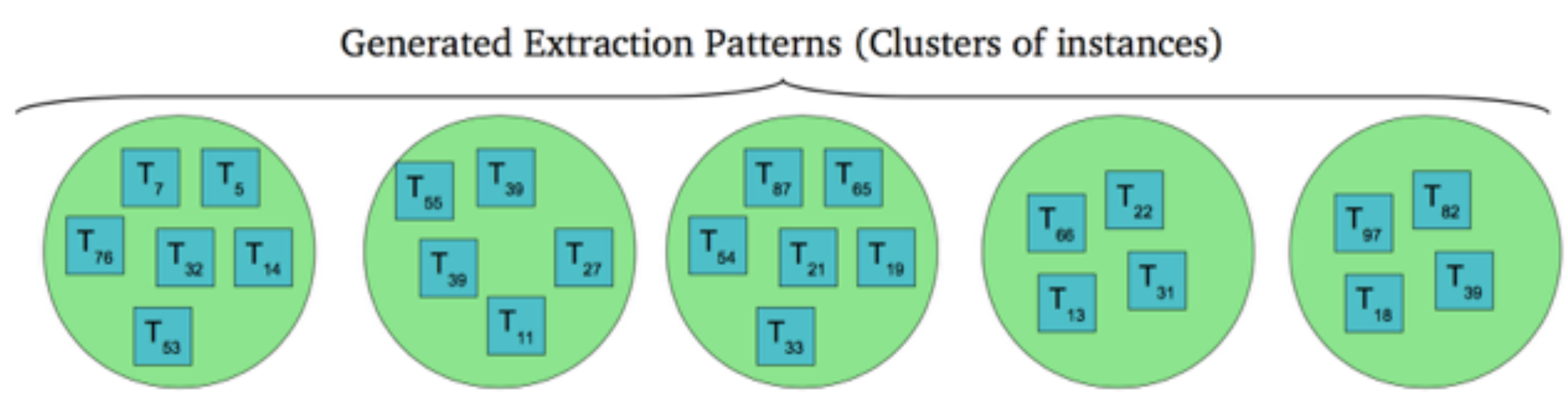
$$Sim(T_i, T_j) = \alpha \cdot cos(BEF_i, BEF_j) + \beta \cdot cos(BET_i, BET_j) + \gamma \cdot cos(AFT_i, AFT_j)$$

Similarity between an instance and a cluster:

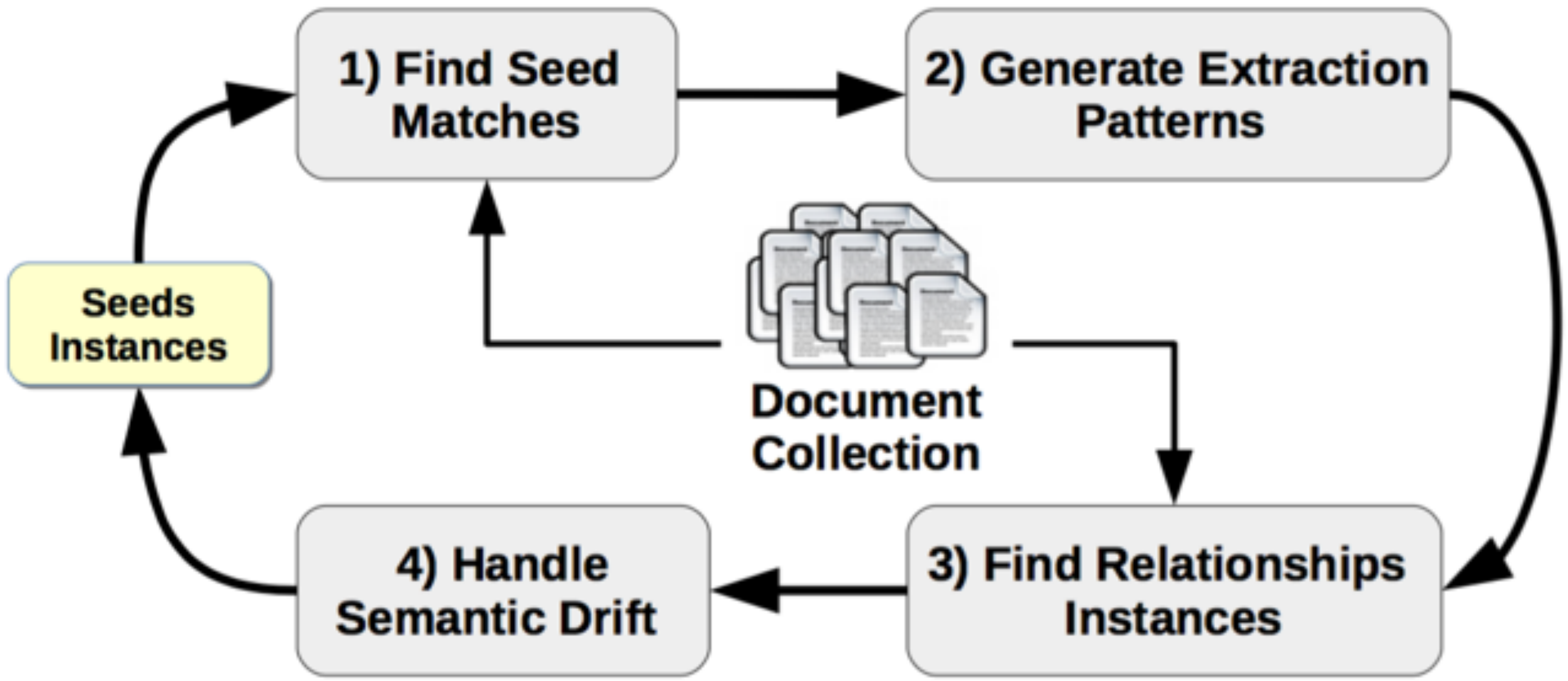
- maximum of all the similarities between any of the instances in a cluster, if the majority of the similarity scores is higher than τ_{sim}
- 0 otherwise



No means are computed



BREDS: Bootstrapping Relationship Instances with Distributional Semantics



keep te same architecture

Find Relationship Instances

- Collect all segments of text containing entity pairs whose semantic types match the seeds

<Google, Mountain View>

<Soundcloud, Berlin>

Document Collection

[..text..] **ORG** [..text..] **LOC** [..text..]

BEF **BET** **AFT**

- If similarity between a tuple and an extraction pattern is equal or above τ_{sim}

- Extract the instance and update the confidence score of the pattern

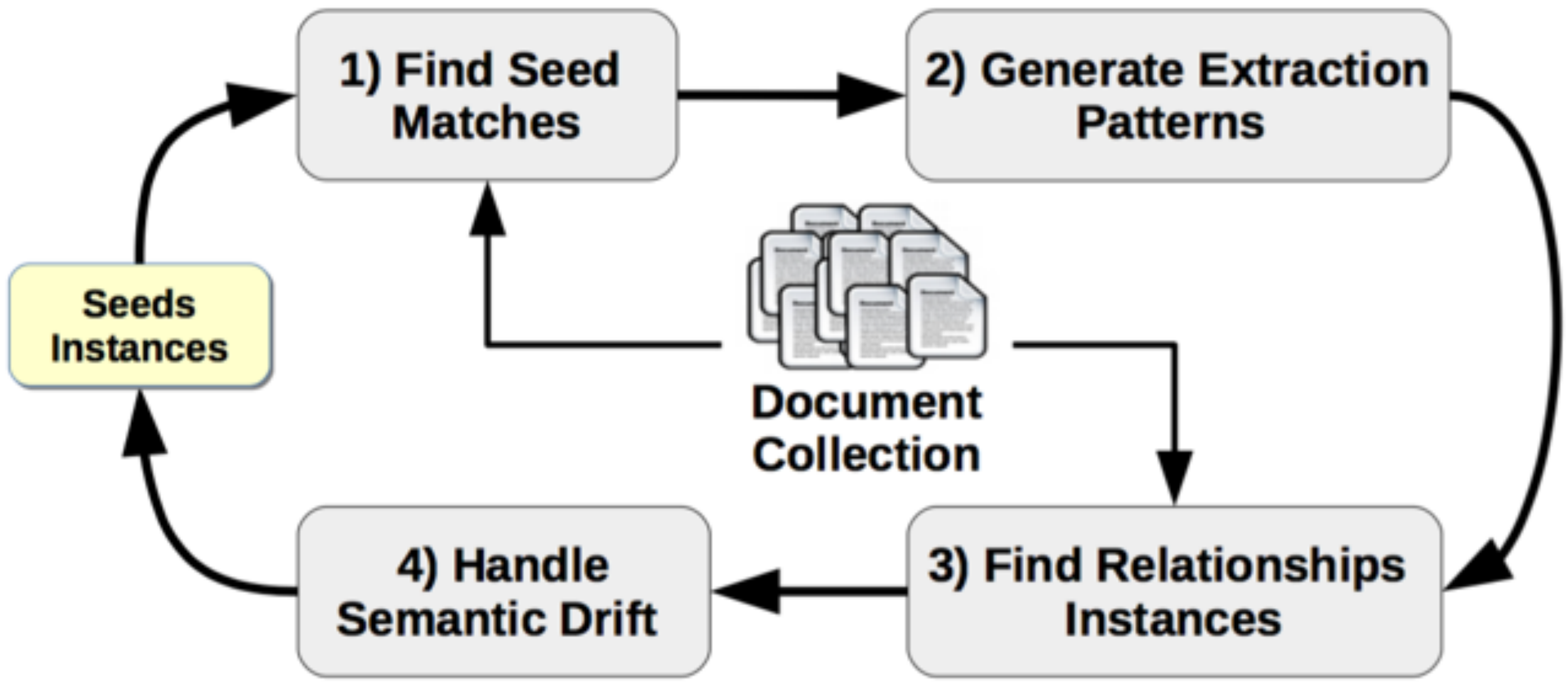
$$\text{Conf}_\rho(p) = \frac{|P|}{|P| + W_{ngt} \cdot |N| + W_{unk} \cdot |U|}$$

Find Relationship Instances: scoring patterns

$$\mathrm{Conf}_{\rho}(p) = \frac{|P|}{|P| + W_{ngt} \cdot |N| + W_{unk} \cdot |U|}$$

- For each extracted instance, <e1, e2>:
  - **NEGATIVE:** if e1 is in the seed set, and the associated e2 does not correspond to the e2 in the extracted relationship:
  - **POSITIVE:** if e1 is in the seed set, and the associated e2 correspond to the e2 in the extracted relationship:
  - **UNKNWON:** e1 is not in the seed set
- For each extracted instance, keep track of the pattern(s) that extracted it and the similarity score(s) - used ahead to compute instance confidence score

BREDS: Bootstrapping Relationship Instances with Distributional Semantics



keep te same architecture

Semantic Drift

Happens when relationships instances, where seed occurs, but with different semantics are added to the seed set:

<**Google, Mountain View>**

"**Google**'s headquarters in **Mountain View**"

"**Google**, based in **Mountain View**"

"**Google**'s shareholders meeting in **Mountain View**"

- Leads to generating extraction patterns that target other relationship types.
- Errors propagate, the semantics of the extracted relationships rapidly drifts away from the original.

Handle Semantic Drift: scoring instances

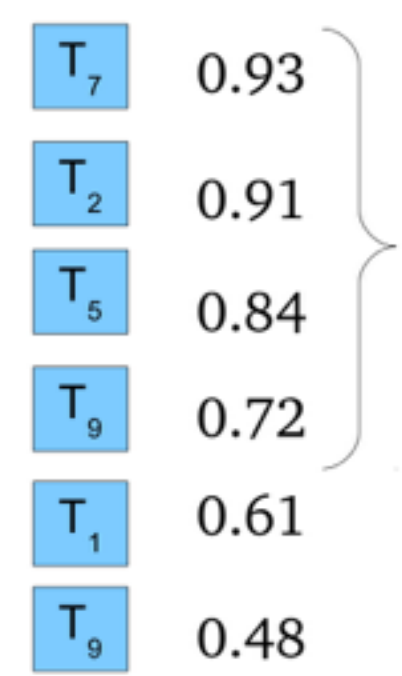
- Rank the extracted instances according to a confidence metric:

$$\mathrm{Conf}_\iota(i) = 1 - \prod_{j=0}^{|\xi|} \left(1 - \mathrm{Conf}_\rho(\xi_j) \times \mathrm{Sim}(C_i, \xi_j)\right)$$

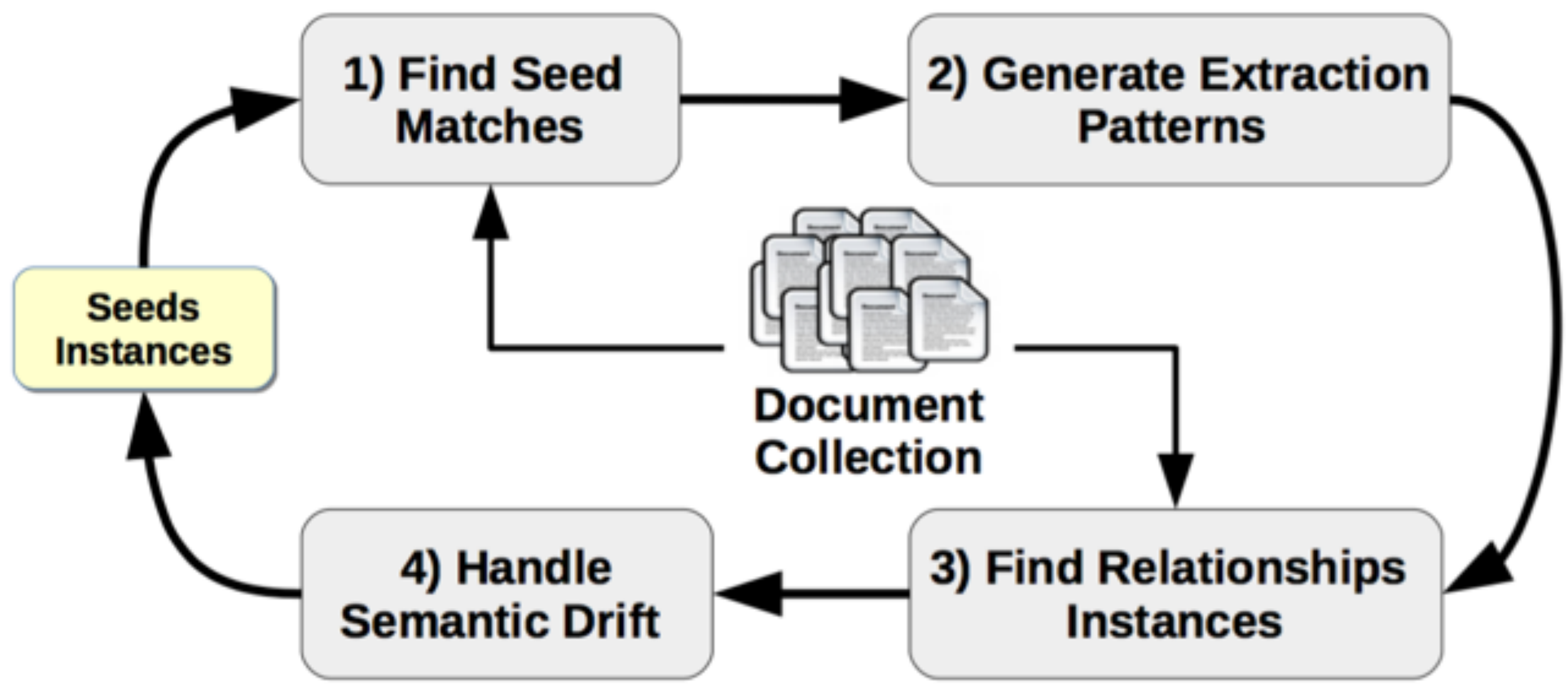
  - ξ is the set of patterns that extracted a relationship *i*
  - *C* is the textual context of an instance

- Add to the seed set all instances with a confidence score above a certain threshold τ_{min}

$$\mathrm{Conf}_\iota(i) \geq \tau_{min}$$



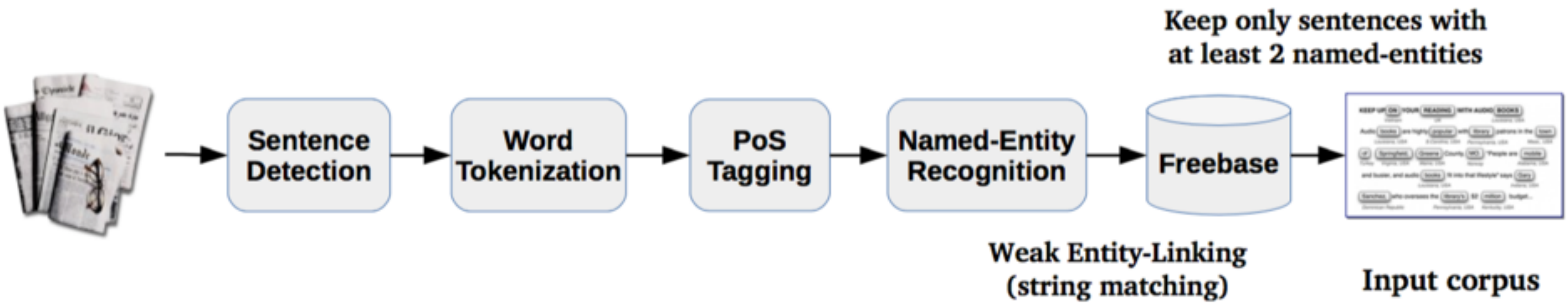
BREDS: Bootstrapping Relationship Instances with Distributional Semantics



Outline

1. ~~Approaches for Semantic Relationship Extraction~~
2. ~~Semi-Supervised/Bootstrapping~~
3. ~~Snowball: TF-IDF~~
4. ~~BREDS: Word Embeddings~~
5. Experimental Evaluation

Experimental Evaluation: setup



- **Document collection:** 5.5 millions news articles (1994-2010)
- **Pre-processing:** Python NLTK and Stanford NER (PER, LOC, ORG)
- **Skip-gram Embeddings**: skip_length=5 and vectors_dim=200
- **Freebase (Knowledge Base):** keep only the sentences containing at least two entities mentioned in Freebase (1.2 million sentences)

Experimental Evaluation: systems and seeds

- **BREDS:** Embeddings + selected words
- **Snowball (ReVerb):** TF-IDF w/ selected words
- **Snowball (Classic):** TF-IDF

- **Parameters**
  - τ_{sim}: [0.5,…,1.0]
  - τ_{min}: [0.5,…,1.0]

Configuration	Context Weighting
Conf_1	$\alpha = 0.0$ $\beta = 1.0$ $\gamma = 0.0$
Conf_2	$\alpha = 0.2$ $\beta = 0.6$ $\gamma = 0.2$

Relationship	Seeds
acquired	<Adidas, Reebok> <Google, DoubleClick>
founder-of	<CNN, Ted Turner> <Amazon, Jeff Bezos>
headquarters	<Nokia, Espoo> <Pfizer, New York>
affiliation	<Google, Marissa Mayer> <Xerox, Ursula Burns>

Results

BREDS

Relationship	Conf$_1$ #Instances	Conf$_1$ (P)recision	Conf$_1$ (R)ecall	Conf$_1$ F$_1$	Conf$_2$ #Instances	Conf$_2$ (P)recision	Conf$_2$ (R)ecall	Conf$_2$ F$_1$
acquired	132 (2.1%)	0.73	**0.77**	**0.75**	5 (0.3%)	**1.00**	0.15	0.26
founder-of	413 (6.6%)	**0.98**	**0.86**	**0.91**	261 (16.2%)	0.97	0.79	0.87
headquartered	870 (14.0%)	0.63	**0.69**	**0.66**	614 (38.1%)	**0.64**	0.61	0.62
affiliation	4806 (77.3%)	**0.85**	**0.91**	**0.88**	730 (45.3%)	0.84	0.60	0.70
Weighted Avg. for P, R and F$_1$		0.83	0.87	0.85	—	0.79	0.63	0.70

(a) Precision, Recall and F$_1$ over the extracted instances with the two different configurations of BREDS

Snowball (ReVerb)

Relationship	Conf$_1$ #Instances	Conf$_1$ (P)recision	Conf$_1$ (R)ecall	Conf$_1$ F$_1$	Conf$_2$ #Instances	Conf$_2$ (P)recision	Conf$_2$ (R)ecall	Conf$_2$ F$_1$
acquired	53 (3.5%)	0.83	0.61	0.70	11 (1.8%)	0.73	0.22	0.34
founder-of	241 (16.1%)	0.96	0.77	0.86	212 (35.3%)	0.97	0.75	0.85
headquartered	891 (59.4%)	0.48	0.63	0.55	322 (53.7%)	0.55	0.42	0.47
affiliation	316 (21.1%)	0.52	0.29	0.37	55 (9.2%)	0.36	0.05	0.08
Weighted Avg. for P, R and F$_1$		0.58	0.58	0.58	—	0.68	0.50	0.57

(b) Precision, Recall and F$_1$ over the extracted instances with the two different configurations of Snowball (ReVerb)

Snowball (Classic)

Relationship	Conf$_1$ #Instances	Conf$_1$ (P)recision	Conf$_1$ (R)ecall	Conf$_1$ F$_1$	Conf$_2$ #Instances	Conf$_2$ (P)recision	Conf$_2$ (R)ecall	Conf$_2$ F$_1$
acquired	38 (2.8%)	0.87	0.54	0.67	43 (5.0%)	0.77	0.54	0.63
founder-of	222 (16.6%)	0.97	0.76	0.85	187 (21.6%)	0.98	0.73	0.84
headquartered	743 (55.7%)	0.52	0.61	0.57	551 (63.8%)	0.53	0.54	0.54
affiliation	332 (24.9%)	0.49	0.29	0.36	83 (9.6%)	0.42	0.08	0.13
Weighted Av for P, R and F$_1$		0.60	0.55	0.57	—	0.63	0.54	0.57

Results Analysis

- BREDS highest F1 scores due to a higher recall caused by the use of embeddings.
- Using only the BET context yields a higher performance than using BEF, BET, AFT.

Relationship	BREDS	Snowball (ReVerb and Classic)
acquired	acquired acquisition purchased by 's purchase of	acquisition acquired
founder-of	founder co-founder co-founders founded	founder
headquartered	based in headquarters in headquartered in offices in	based in headquarters in
affiliation	president chief executive vice-president general manager CEO chairman	president chief executive

Improvements

“The **ICJ** which is part of the **UN** is based in **The Hague**”

Improvements

“The **ICJ** which is part of the **UN** **is based in** **The Hague**”

Improvements

"The **ICJ** which is part of the **UN is based in The Hague**"

<**UN, is based in, The Hague**>

Improvements

“The **ICJ** which is part of the **UN is based in The Hague**”

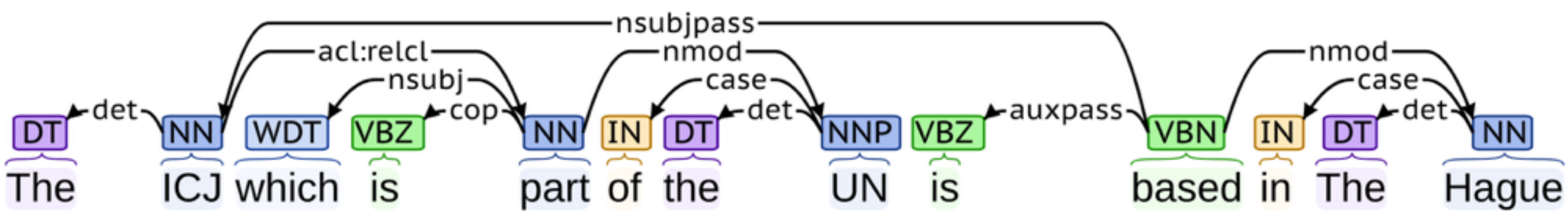
~~<**UN, is based in, The Hague**>~~

Improvements

“The **ICJ** which is part of the **UN** **is based in** **The Hague**”

~~<**UN,** **is based in,** **The Hague**>~~

Compute syntactic dependencies



Improvements

Entity-Linking: disambiguation of an entity according to a knowlege-base

“George Bush”, “Bush”

BUSH

Advantage over NER: can capture more contexts where the same entity is mentioned.

Thank you :-)

https://github.com/davidsbatista/BREDS

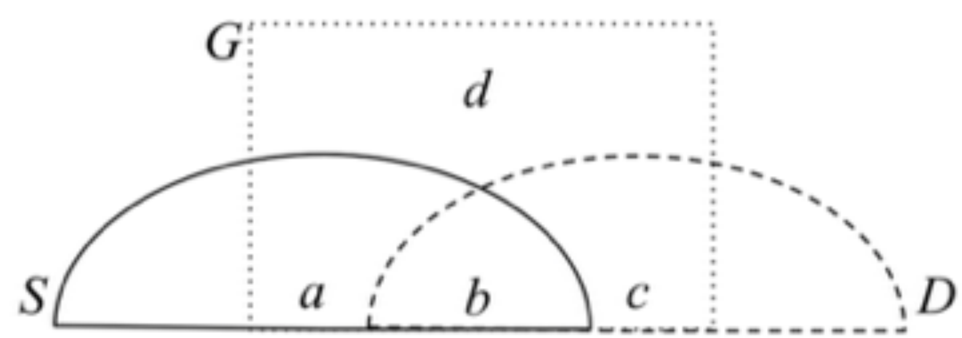
https://github.com/davidsbatista/Snowball

Semi-Supervised Bootstrapping of Relationship Extractors with Distributional Semantics David S. Batista, Bruno Martins, and Mário J. Silva EMNLP'15

http://davidsbatista.net

Addendum

Evaluation Framework



D: Knowledge Base, **G** ground truth,

S: system output

- *a*: correct relationships from system output not in KB
- *b*: intersection between system output and KB
- *c*: KB relationships in the corpus but not extracted by the system
- *d*:relationships in the corpus not extracted by the system nor in the KB

a: relationships only contain entities from the KB, so this intersection is trivial

b: Proximate PMI $\mathrm{PPMI(e_1, rel, e_2)} = \frac{\mathrm{count(e_1\ NEAR}{:}X\ \mathit{rel}\ \mathrm{NEAR}{:}X\ \mathrm{e_2)}}{\mathrm{count(e_1\ AND\ e_2)}}$

c: Generate *G'*, all possible (i.e.: correct and incorrect) relationships at a sentence level and estimate $|G \cap D| = |b| + |c|$, then $|c| = |G \cap D| - |b|$

d: Calculate Proximate PMI for all the relationships not in the database

$G' \setminus D$, then $d = |G \setminus D| - |a|$

$$P = \frac{|a| + |b|}{|S|} \qquad R = \frac{|a| + |b|}{|a| + |b| + |c| + |d|}$$

"Automatic Evaluation of Relation Extraction Systems on Large-scale" (Bronzi et al. 2012)