Large-Scale Semantic Relationship Extraction for Information Discovery

David Soares Batista

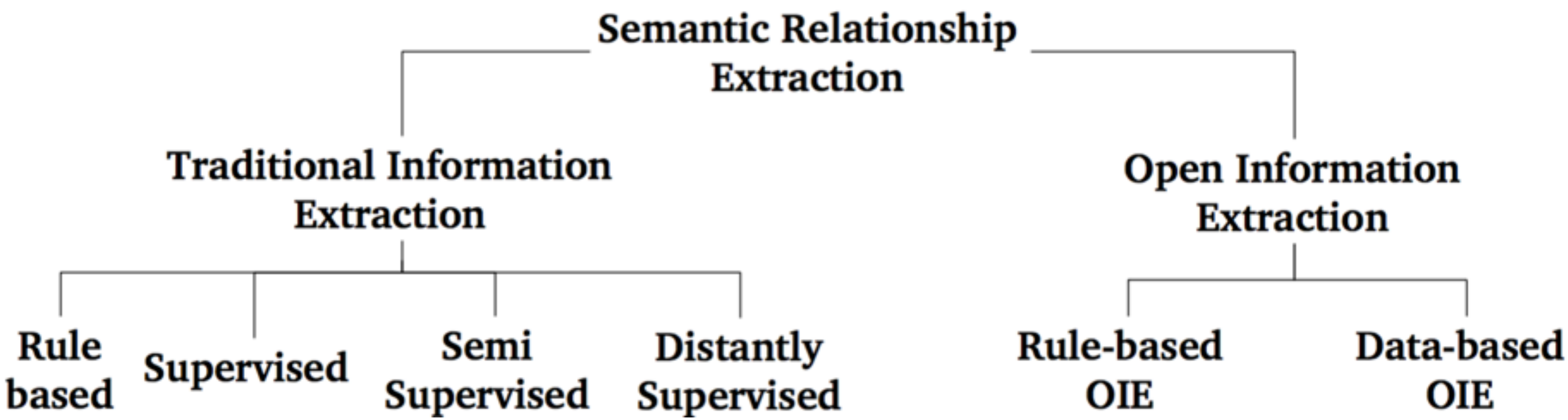
Lisbon, June 22, 2016

Relationship Extraction (RE)

Noam Chomsky was born in the **East Oak Lane** neighbourhood of **Philadelphia**, **Pennsylvania**.

- (Noam Chomsky, East Oak Lane) → born-place
- (East Oak Lane, Philadelphia) → part-of
- (Philadelphia, Pennsylvania) → part-of

Taxonomy


Semantic Relationship Extraction
Traditional Information Extraction
Open Information Extraction
Rule based
Supervised
Semi Supervised
Distantly Supervised
Rule-based OIE
Data-based OIE

Motivation for Large-Scale RE

- Massive scale events trigger bursts of text
  - Disease outbreaks
  - Terrorist attacks
  - Sport Events: Euro 2016
- On-line question answering requires fast and scalable RE. However:
  - Training of Support Vector Machines (SVM) involves a quadratic optimisation problem
  - Multiple binary classifiers needed to extract different relationship types.

Research Question 1

IDEA: Explore the use of a similarity metric, and searching similar relationship examples for RE instead of learning a statistical model

Can supervised large-scale relationship extraction be efficiently performed based on similarity search ?

Motivation for Bootstrapping RE

- Supervised relationship extraction relies on training data
  - Not always available
  - Manual annotation can be prohibitive

- Unlabelled data is vast and abundant
  - Bootstrapping approaches leverage on such data
  - Relying on seed instances and contextual similarity

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

"**Porsche** *has its main headquarters in* **Stuttgart**"

Research Question 2

- Classic approaches use TF-IDF weighted vectors to represent the context

X = "main headquarters in"	1.3	2.3	0	0
Y = "is based in"	0	0	3.3	0
X = "is headquartered in"	0	0	0	2.5

cos_sim(X,Y) = 0
cos_sim(X,Z) = 0
cos_sim(Y,Z) = 0

IDEA: explore word embeddings

"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

Can distributional semantics improve the performance of bootstrapping relationship instances ?

Methodology

Research Question 1

- Develop a new supervised RE approach based on similarity search.
- Identify state-of-the-art approaches for baseline.
- Compare performance against baseline on public datasets.

Research Question 2

- Develop a new approach for bootstrapping relationship instances based on word embeddings.
- Identify baseline approaches based on TF-IDF weighted vectors.
- Compare performance against baseline on public datasets.

Outline

1. ~~Research Questions and Methodology~~
2. Research Question 1:
   Supervised Relationship Extraction as Similarity Search
3. Research Question 2:
   Bootstrapping Relationship Extractions with Distributional Semantics
4. Large-scale Relationship Extraction
5. Conclusions and Future Work

Supervised Relationship Extraction as Similarity Search

- MuSICo - MinHash-based Semantic Relationship Classifier
- Similarity techniques explored:
  - Jaccard similarity between relationship instances
  - Min-Hash to quickly estimate Jaccard similarity
  - Locality Sensitive Hashing (LSH) to identify the most similar instances efficiently

"*A Minwise Hashing Method for Addressing Relationship Extraction from Text*"
David S. Batista, Rui Silva, Bruno Martins, and Mário J. Silva. WISE'13

"*Exploring DBpedia and Wikipedia for Portuguese Semantic Relationship Extraction*"
David Soares Batista, David Forte, Rui Silva, Bruno Martins, and Mário J. Silva. Linguamática, 5(1), 2013

Min-Hash: Jaccard Similarity Estimation

- Given a vocabulary Ω of size *n* and two sets, A and B, where: A,B ⊆ Ω:

$$\text{Jaccard}(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

- Applying a **random permutation** π on the **ordering considered for the elements,** the Jaccard similarity **can be estimated** from the probability of the first values of the random permutation π being equal (Border 1997):

$$\mathrm{P}\left(\min(A) = \min(B)\right) = \frac{|A \cap B|}{|A \cup B|} = \text{Jaccard}(A,B)$$

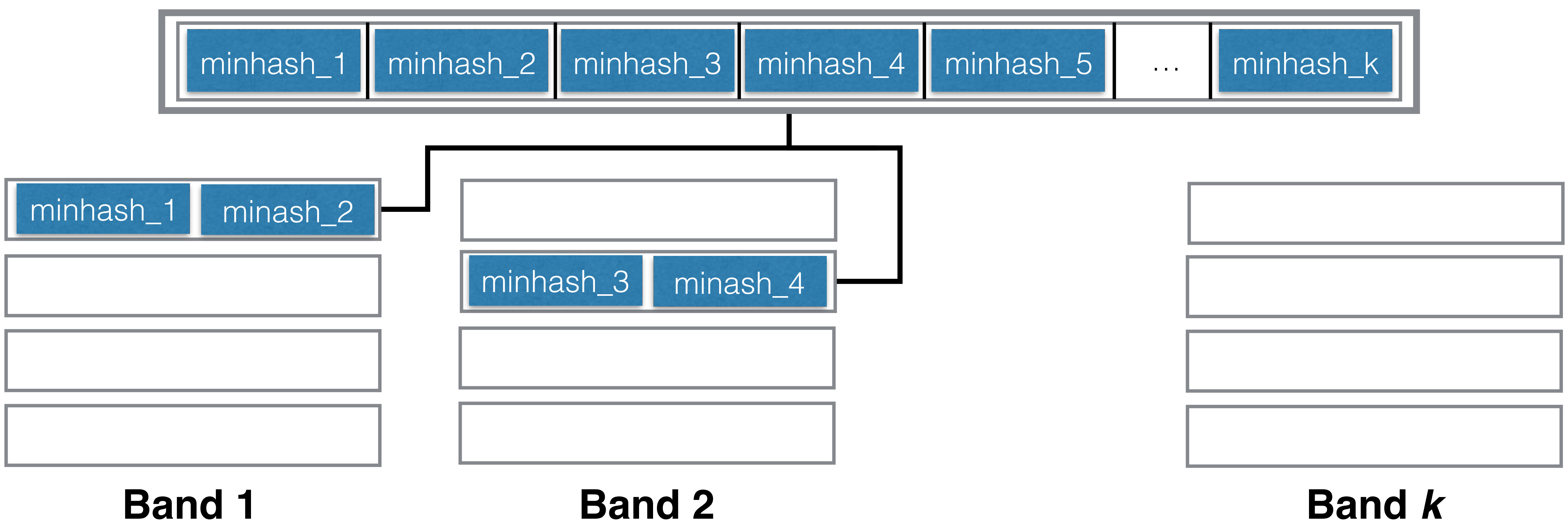
- Having *k* independent permutations one can efficiently estimate Jaccard(*A*, *B)* by applying *k* hashing functions to each element and keeping the minimum

minhash_1	minhash_2	minhash_3	minhash_4	minhash_5	. . .	minhash_k

Locality-Sensitive Hashing

- The minhash signature is split into L different bands (constraint: $k \bmod L = 0$)

- An index is built with L different hash tables, each corresponding to an n-tuple from the min-hash signature.



Feature Extraction

“The tech company ***Soundcloud*** *is based in* ***Berlin,*** *the capital of Germany.“*

BEFORE **BETWEEN** **AFTER**

- Characters n-grams of size 4
- Root forms of verbs (except auxiliary verbs)
- Prepositions: *between*, *above*, *within*, etc.;
- Passive Voice Detection: indicate direction of relation
  - “*Harry ate six shrimps at dinner.*” (active voice)
  - “*Six shrimps were eaten by Harry.*” (passive voice)
- Identify and normalise ReVerb Patterns:

  “*Jack White is the guitar player of the White Stripes*”

  “*is the guitar player of*”

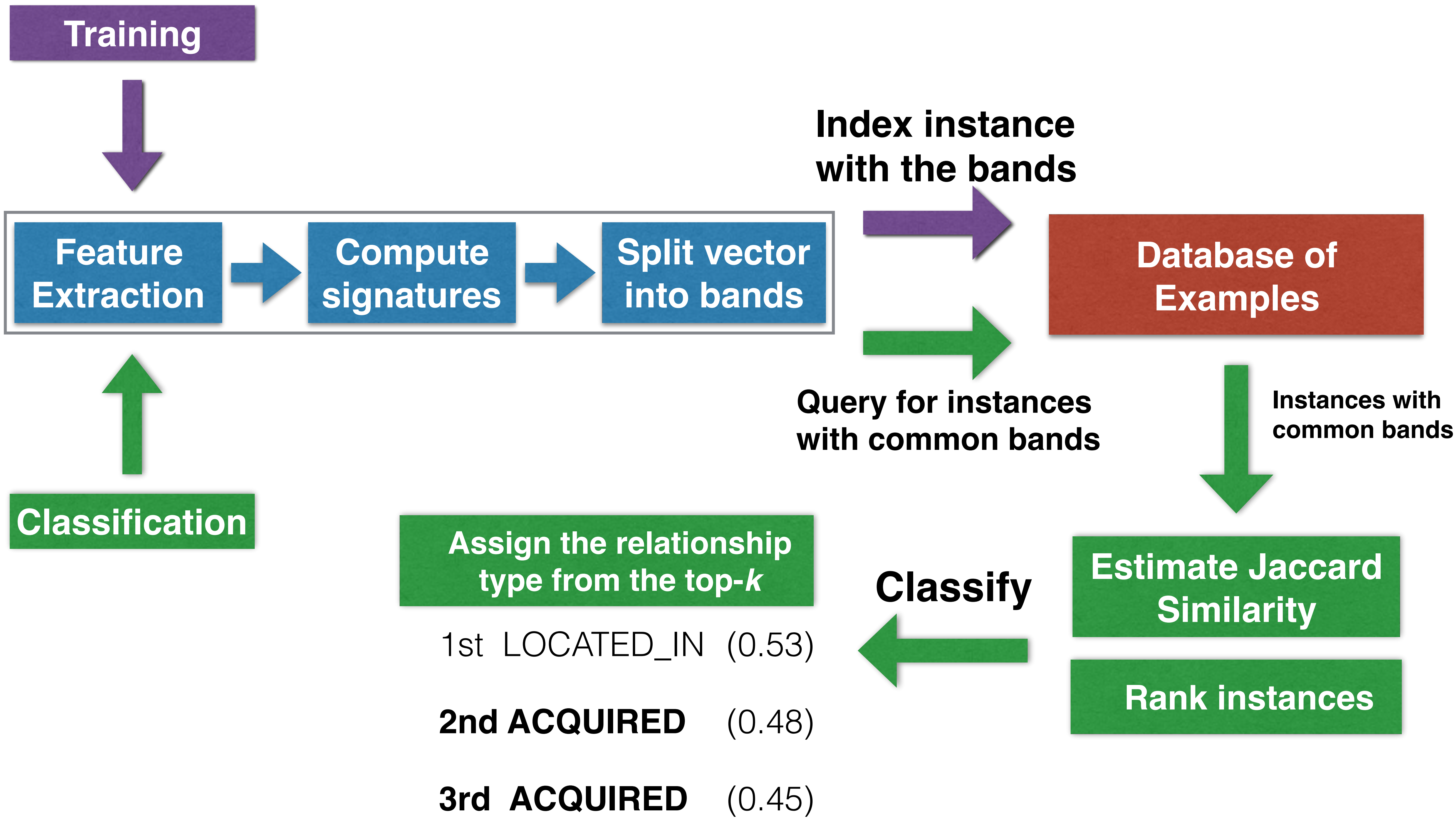
Passive Voice

BE VBD “by”
BE = any form of “to be”
VBD = verb in past tense

ReVerb

V | V P | V W* P
V= verb particle? adv?
W = (noun | adj | adv | pron | det)
P = (prep | particle | inf. marker)

Architecture: Indexing and Classification



Evaluation

- **SemEval 2010 Task 8** (Hendrickx et al., 2010)
  - 10 717 sentences
  - 19 classes
  - Generic web text
- **Wikipedia** (Culotta et al., 2006):
  - 3 125 sentences
  - 47 classes (highly skewed dataset)
  - Wikipedia articles (English)
- **Aimed** (Bunescu and Mooney, 2005a):
  - 2 202 sentences
  - 2 classes
  - Protein interactions from MEDLINE abstracts
- **DBPediaRelations-PT** (Batista et al., 2013b)
  - 97 988 sentences
  - 10 classes
  - Wikipedia articles (Portuguese)

- **Configuration parameters:**
  - min-hash signatures: 200, 400, 600, 800;
  - LSH bands: 25, 50;
  - *k* nearest neighbours: 1, 3, 5, 7;

Evaluation Results

Aimed

- k-NN = 3
- Min-Hash = 800
- Bands = 50

F_1	Kernel Type	Syntactic Dependencies	PoS-tags
0.56	All-Paths Graph Kernel	YES	NO
0.55	Shallow Linguistic Kernel	NO	YES
0.52	MuSICo	NO	YES

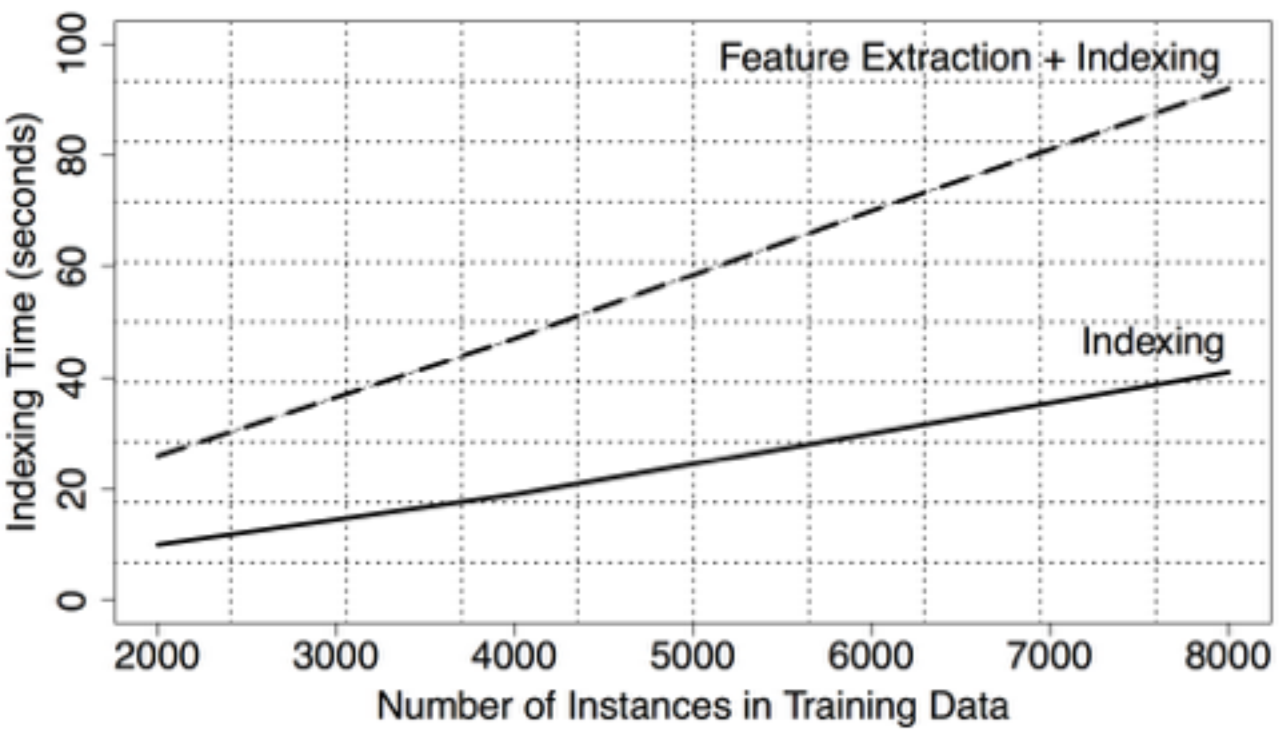
All-Paths Kernel (Train+Testing): 4 524 seconds
Shallow Linguistic Kernel (Train+Testing): 77.2 seconds
MuSICo (FE + Index + Classification): 161 seconds

SemEval 2010 Task 8

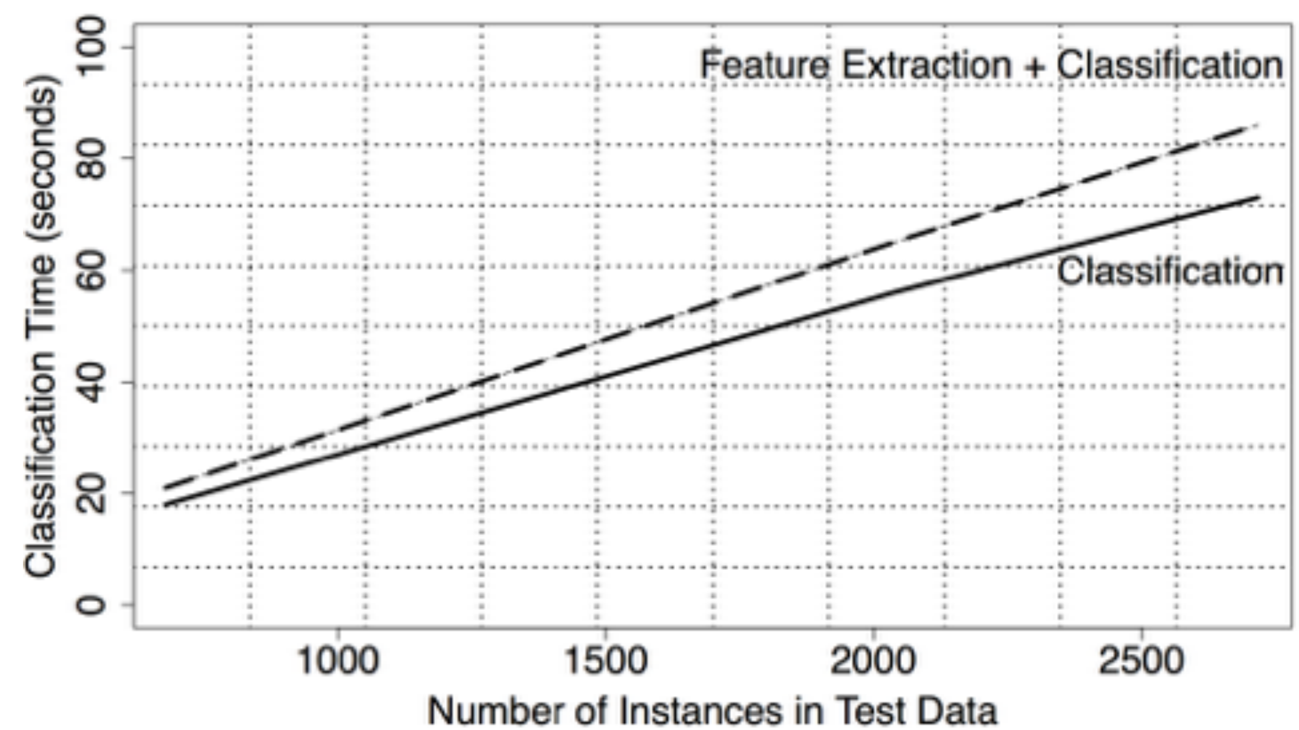
- k-NN = 5
- Min-Hash = 400
- Bands = 50
- Total Time: 172 seconds

F_1	Approach	Syntactic Dependencies	PoS-tags	External Resources
0.82	2 SVM classifers	YES	YES	YES
0.77	4 Kernels (SVM)	NO	YES	YES
0.77	Logistic Regression	NO	NO	YES
0.75	SVM	YES	YES	YES
0.69	MuSICo	NO	YES	NO

Scalability on SemEval 2010 Task 8



Indexing: Training set (25%, 50%, 75%,100%)



Classification: Test set (25%, 50%, 75%,100%)

Feature extraction: compute quadgrams of characters + PoS tagging

Indexing: calculating the min-hash signatures + splitting and indexing in the LSH

Classification: estimate Jaccard similarity + Ranking + assign the relationship type from the top-*k*

Results Analysis

MuSICo:

- Simple set of features common across 3 different domains
  - Character *n*-grams
  - PoS-tagging
- Does not rely on any kind of external resources
- Addresses multi-class classification directly

Baseline Systems:

- WordNet, VerbNet, etc.
- Syntactic Dependencies
- Kernel-based approaches use SVM
  1. Compute features from syntactic dependencies tree and external resources.
  2. Compute pairwise similarities.
  3. Apply the SVM algorithm.
- One-Versus-All classification

MuSICo summary

Accuracy trade-off for:

- **Scalability:** processing time grows linearly with data size.
- **On-Line Learning:** to incorporate new training instances, compute their min-hash signatures and store them.
- **Multi-Class Classification**

Outline

1. ~~Research Questions and Methodology~~
2. ~~Research Question 1:~~
   ~~Supervised Relationship Extraction as Similarity Search~~
3. Research Question 2:
   Bootstrapping Relationship Extractions with Distributional Semantics
4. Large-scale Relationship Extraction
5. Conclusions and Future Work

Bootstrapping Relationship Instances

Rely on seed instances and contextual similarity with seeds

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

"**Porsche** *has its main headquarters in* **Stuttgart**"

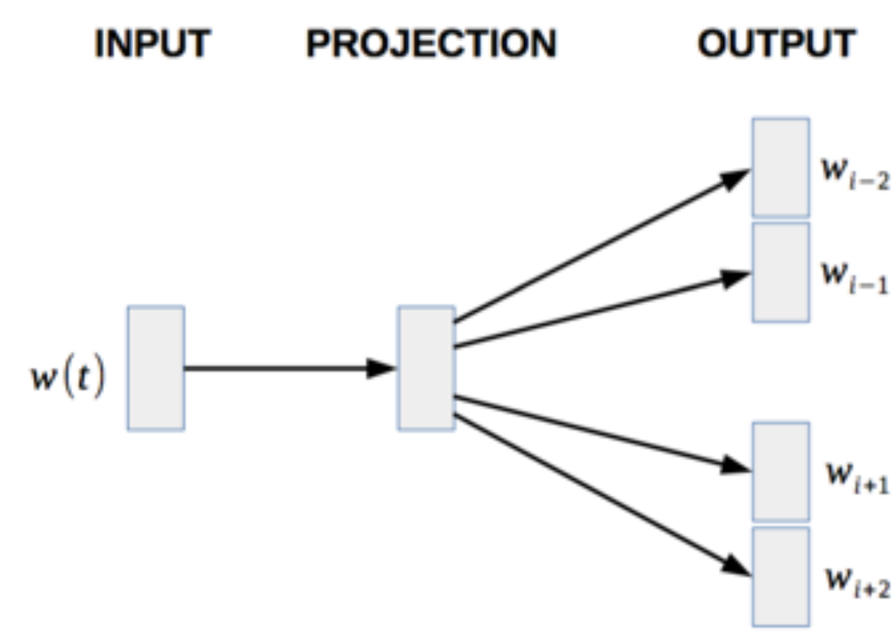
Previous approaches use TF-IDF weighted vectors

Distributional Semantics

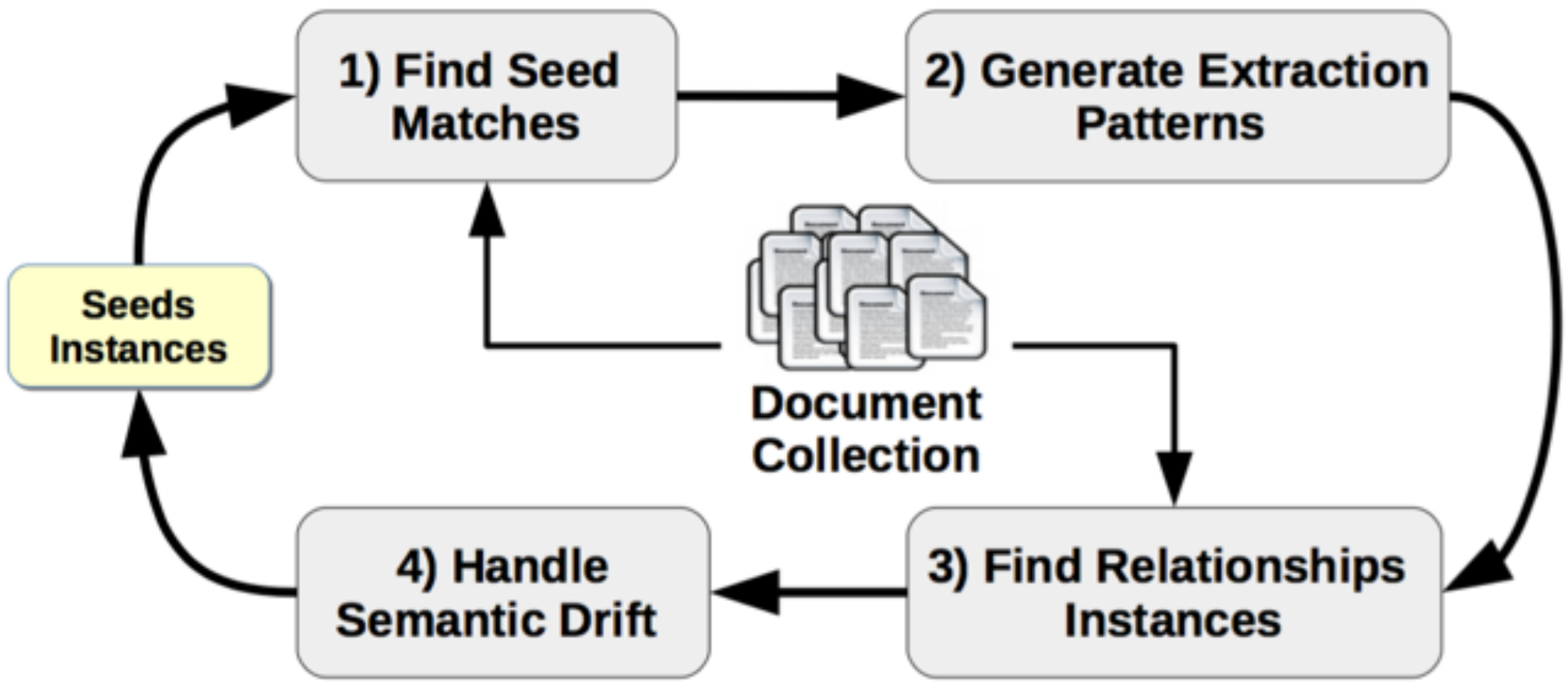
"*You shall know a word by the company it keeps*" (Firth, 1957)

- Brown Clustering (Brown et al., 1992)
- Latent Semantic Analysis (Landauer and Dunais, 1997)
- Neural Probabilistic Language Model (Bengio et al. 2003)

- **Skip-Gram** (Mikolov et al. 2013a,b)
  - Given a word, predict the most probable surrounding words in a context window.
  - In the process of estimating model parameters, the network learns **word embeddings**: word representations by real-valued vectors of low dimensions.



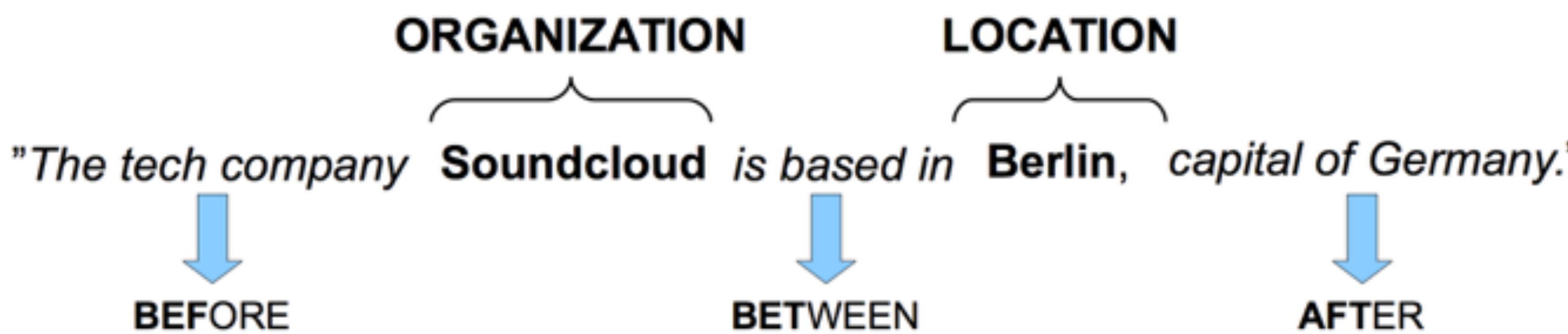
BREDS: Bootstrapping Relationship Instances with Distributional Semantics



BREDS follows the same architecture and metrics of Snowball (Agichtein et al., 2000) but relies on word embeddings instead of TF-IDF.

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

Find Seed Matches



1. BET: extract ReVerb patterns or all words if no verbs are found

“*Soundcloud is based in Berlin*”: **is based in**

“*Soundcloud headquarters in Berlin*”: **headquarters in**

2. Detect if passive voice is present

3. Transform each context into a single vector

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

$$T_n \begin{cases} Vector_{BEFORE} = E(''tech'') + E(''company'') \\ Vector_{BETWEEN} = E(''is'') + E(''based'') \\ Vector_{AFTER} = E(''capital'') \end{cases}$$

Generate Extraction Patterns

- Cluster all collected seed instances

$$
\begin{aligned}
\mathrm{Sim}(T_i, T_j) &= \alpha \cdot \cos(BEF_i, BEF_j) \\
&+ \beta \cdot \cos(BET_i, BET_j) \\
&+ \gamma \cdot \cos(AFT_i, AFT_j)
\end{aligned}
$$

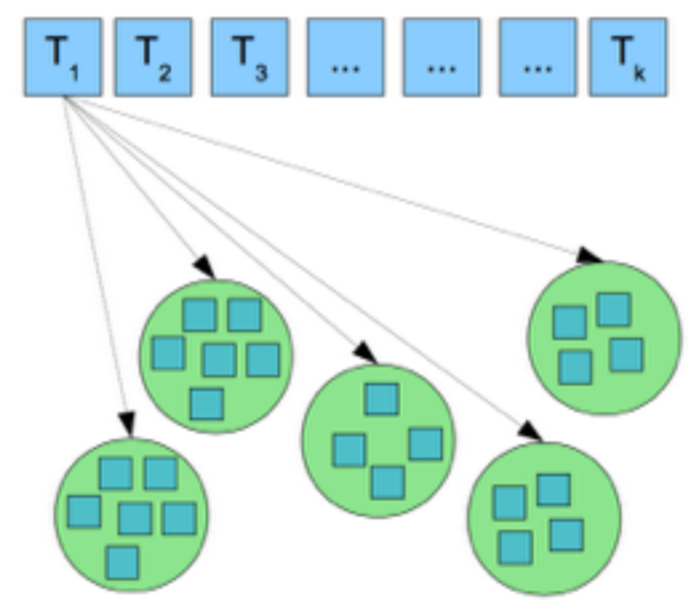
Similarity threshold parameter: τ_{sim}

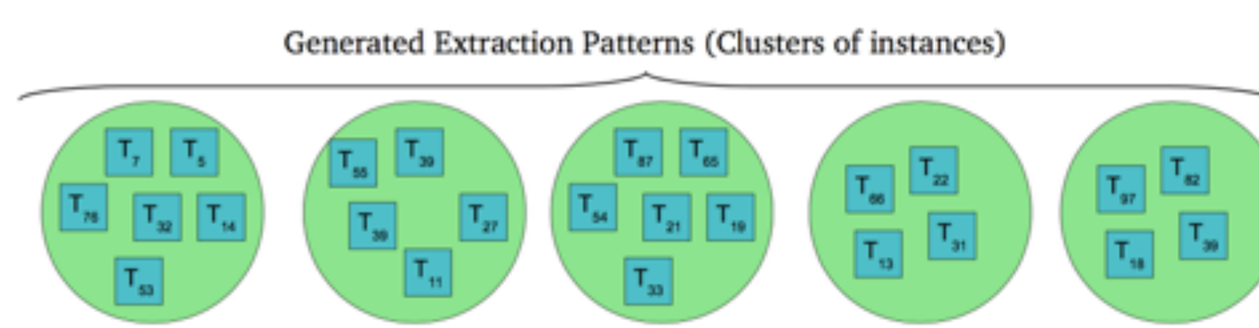
Algorithm 1: Single-Pass Clustering.

Input: $Instances = \{i_1, i_2, i_3, ..., i_n\}$
Output: $Patterns = \{\}$
$Cl_1 = \{i_1\}$
$Patterns = \{Cl_1\}$
for $i_n \in Instances$ **do**
  for $Cl_j \in Patterns$ **do**
    if $Sim(i_n, Cl_j) >= \tau_{sim}$ **then**
      $Cl_j = Cl_j \cup \{i_n\}$
    else
      $Cl_m = \{i_n\}$
      $Patterns = Patterns \cup \{Cl_m\}$

Similarity between an instance and a cluster:

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





Find Relationship Instances

Collect all segments of text containing entity pairs whose semantic types match the types of the seeds, e.g:

- **<Google, Mountain View>**
- Collect all <ORG,LOC> text segments
- Generate 3 vectors

Algorithm 2: Find Relationship Instances.

Input: $Sentences = \{s_1, s_2, s_3, ..., s_n\}$

Input: $Patterns = \{Cl_1, Cl_2, ..., Cl_n\}$

Output: $Candidates$

```
for s_i ∈ Sentences do
    i = create_instance(s_i)
    sim_best = 0
    p_best = None
    for Cl_i ∈ Patterns do
        sim = Sim(i, Cl_i)
        if sim >= τ_sim then
            Conf_ρ(C_i)
            if sim >= sim_best then
                sim_best = sim
                P_best = Cl_i
    Candidates[i].patterns[p_best] = sim_best
```

- Calculate similarity with every extraction pattern
- If the similarity between an instance and an extraction pattern is equal or above τ_{sim}
- Extract the instance and update the confidence score of the pattern

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

Handle Semantic Drift

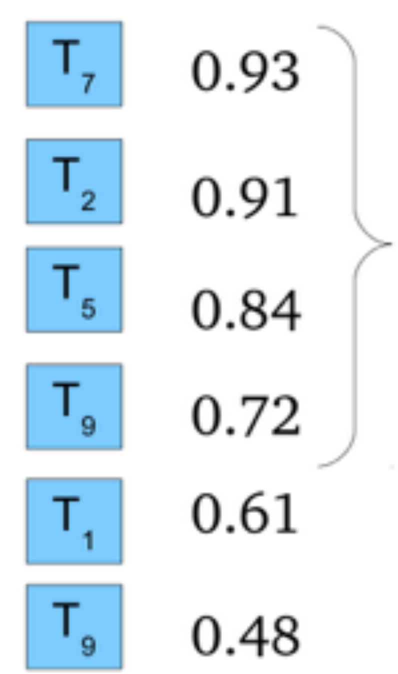
- Rank the extracted instances according to a confidence metric:

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

    - 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}

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



Experimental Evaluation

- **Dataset**: 5.5 million news articles
  - Selected 1.2 million sentences with at least 2 named-entities
  - Word embeddings
  - TF-IDF vector weights

4 Relationship Types

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>

- **Baseline systems**
  - Snowball-Classic (Agichtein et al., 2000)
  - Snowball-ReVerb (selects words for BET)

- **Thresholds**
  - τ_{sim} :[0.5,1.0]
  - τ_{min} :[0.5,1.0]
  - 36 x 4 (relationship types) x 2 (weighting schema)

2 Weighting Context Vectors Schema

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

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 achieves the 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.

  - BEF and AFT contexts are sparse, containing many different words which do not contribute to the capture the relationship.

- For the 3 evaluated systems different relationship types require different threshold parameters configuration to achieve the best results.

Outline

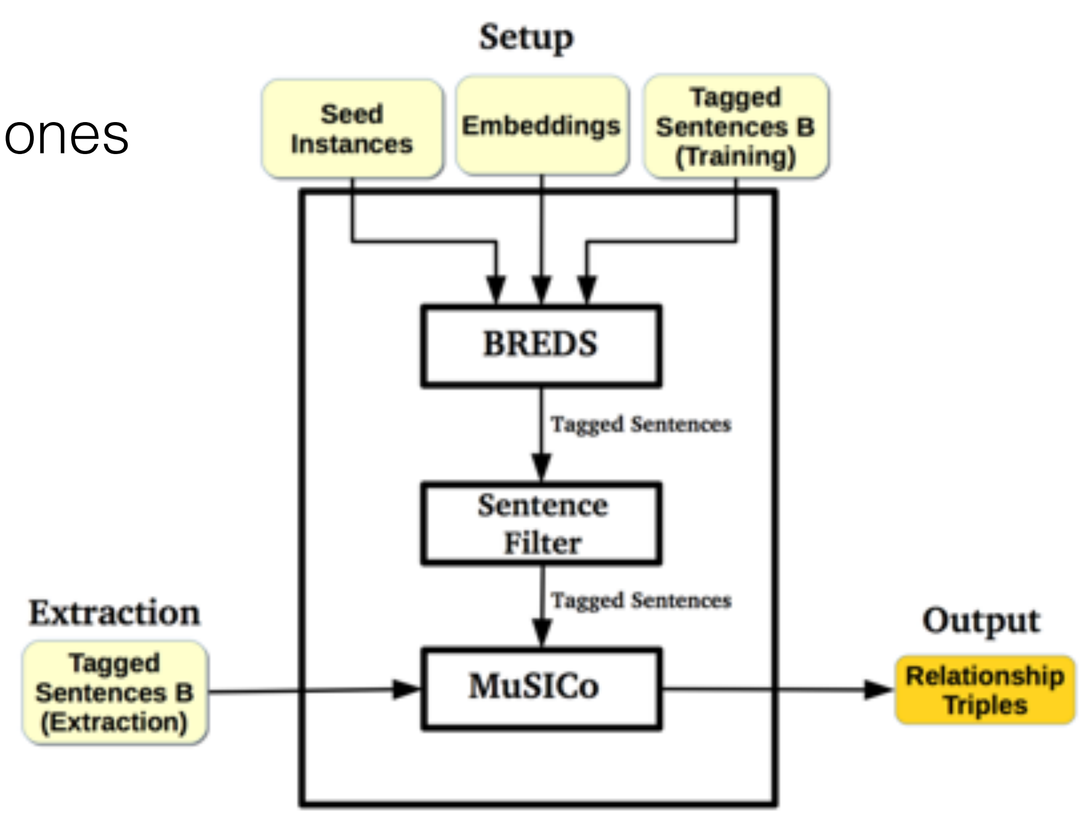
1. ~~Research Questions and Methodology~~
2. ~~Research Question 1:~~
   ~~Supervised Relationship Extraction as Similarity Search~~
3. ~~Research Question 2:~~
   ~~Bootstrapping Relationship Extractions with Distributional Semantics~~
4. Large-scale Relationship Extraction
5. Conclusions and Future Work

TREMoSSo - Triples Extraction with Min-Hash and diStributed Semantics

- Framework integrating MuSICo and BREDS along with other NLP tools
- Extraction of different relationship types with a single-pass over the documents

- **Setup (BREDS)**
  1. Bootstrap relationship instances and filter correct ones
  2. Index the relationship instances

  - **Input Data:**
    - Seed instances
    - Word embeddings
    - A set of sentences tagged with named-entities



- **Extraction (MuSICo)**
  - Extract relationship instances based index examples

TREMoSSo: setup (BREDS)

- 11 relationship types
- 40 seed instances

Relationship	Direction	Seeds
affiliated-with	(ORG,PER)	<Google, Eric Schmidt> <OPEC, Edmund Daukoru> <UEFA, Michel Platini> <WikiLeaks, Julian Assange>
	(PER,ORG)	<Dominique Strauss, IMF> <Henning Kagermann, SAP> <Gianni Agnelli, Fiat> <John Sauven, Greenpeace>
owns/has-parts-in	(ORG_1,ORG_2)	<Adidas, Reebok > <Volkswagen, Audi>
	(ORG_2,ORG_1)	<Mercedes-Benz, Daimler AG> <Airbus, EADS> <Audi, Volkswagen>
founded-by	(ORG,PER)	<CNN, Ted Turner> <Google, Sergey Brin>
	(PER,ORG)	<Dietmar Hopp, SAP AG> <Chung Ju-yung, Hyundai>
has-installations-in	(ORG,LOC)	<Opel, Spain> <Nokia, Espoo> <Volkswagen, Portugal> <Siemens, Munich>
	(LOC,ORG)	<Berlin, Deutsche Welle> <New York, NBC News> <Miami, National Hurricane Center> <Seoul, Samsung Group> <San Jose, Cisco> <London, Unilever>
spouse	(PER,PER)	<George W. Bush, Laura Bush> <Jennifer Lopez, Marc Anthony> <Britney Spears, Kevin Federline>
studied-at	(PER,ORG)	<Barack Obama;Columbia University> <Barack Obama;Harvard University> <Al Gore;Vanderbilt University> <Al Gore;Harvard University>
	(ORG,PER)	<Stanford, Larry Page> <Harvard, Barack Obama> <Harvard, Mark Zuckerberg> <Harvard, Steve Ballmer>

Results

Relationship	Direction	Precision	Recall	F_1
affiliated-with	(ORG,PER)	0.97	0.82	0.89
	(PER,ORG)	0.52	0.53	0.53
owns	(ORG_1,ORG_2)	0.51	0.71	0.60
	(ORG_2,ORG_1)	0.41	0.47	0.44
founded-by	(ORG,PER)	1.00	0.76	0.86
	(PER,ORG)	0.87	0.33	0.48
has-installations-in	(ORG,LOC)	0.82	0.55	0.66
	(LOC,ORG)	0.93	0.58	0.71
spouse	(PER,PER)	0.59	0.59	0.59
studied-at	(PER,ORG)	0.89	0.74	0.81
	(ORG,PER)	0.88	0.41	0.56

Number of Instances per type

Relationship	Direction	# Relationship Instances
affiliated-with	(PER,ORG)	2 708 (13.9%)
	(ORG,PER)	9 775 (50.2%)
owns/has-parts-in	(ORG_1,ORG_2)	501 (2.6%)
	(ORG_2,ORG_1)	100 (0.5%)
founded-by	(ORG,PER)	802 (4.1%)
	(PER,ORG)	92 (0.5%)
has-installations-in	(ORG,LOC)	4 259 (21.9%)
	(LOC,ORG)	362 (1.9%)
spouse	(PER,PER)	725 (3.7%)
studied-at	(PER,ORG)	104 (0.5%)
	(ORG,PER)	36 (0.2%)
Total		19 464 (100%)

TREMoSSo: extraction (MuSICo)

Relationship	Direction	Precision	Recall	F_1
affiliated-with	(ORG,PER)	0.490	0.736	0.588
	(PER,ORG)	0.070	0.293	0.113
owns/has-parts-in	(ORG_1,ORG_2)	0.423	0.194	0.265
	(ORG_2,ORG_1)	0.233	0.095	0.135
founded-by	(ORG,PER)	0.327	0.191	0.241
	(PER,ORG)	0.036	0.020	0.026
has-installations-in	(ORG,LOC)	0.836	0.655	0.734
	(LOC,ORG)	0.386	0.182	0.248
spouse	(PER,PER)	0.486	0.139	0.217
studied-at	(PER,ORG)	0.096	0.394	0.154
	(ORG,PER)	0.250	0.067	0.105

- ca. 4,700 correct relationship
- skewed training set
- relationship types with the lowest number of examples have the most incorrect extractions

- **Setup**: ca. 20 000 sentences (single relationship per sentence)
  - Feature Extraction + Computing Signatures + Indexing = 572 seconds
  - Average: 34.1 sentences per second
- **Extraction:** ca. 850 000 sentences (multi-relationships per sentence)
  - Feature Extraction + Computing Signatures + Computing Similarity = 6 050 seconds
  - Average: 3.2 sentences per second

Outline

1. ~~Relationship Extraction~~
2. ~~Research Questions and Methodology~~
3. ~~Supervised Relationship Extraction as Similarity Search~~
4. ~~Bootstrapping Relationship Extractions with Distributional Semantics~~
5. ~~Large-scale Relationship Extraction~~
6. Conclusions and Future Work

Conclusions

Can supervised large-scale relationship extraction be efficiently performed based on similarity search ?

- New supervised classifier levering on min-hash and locality sensitive hashing
- Empirically evaluated through experiments with datasets from different domains
- Scalable, on-line, address multi-class classification

Can distributional semantics improve the performance of bootstrapping relationship instances ?

- New bootstrapping approach for relationship extraction, based word embeddings
- Evaluated and compared against baseline systems relying on TF-IDF weighted vectors.
- Increase in performance is due to the high recall, which is caused by the relaxed semantic matching enabled by computing similarities based on word embeddings

Future Work

MuSICo:

- Only PoS-tags, fast to compute, but do not capture long distance relationships.
- Teixeira et al. (2012) proposed an algorithm for graph fingerprints based on min-hash, allows to perform similarity search by relying on graph-based representations of syntactic dependencies.

BREDS:

- Only PoS-tags, fast to compute, but do not capture long distance relationships.
- "*semantic drift occurs when a candidate instance is more similar to recently added instances than to the seed instances*" (McIntosh and Curran 2009)
- Entity Linking could alleviate some of the errors generated by simple NER

Final Remarks

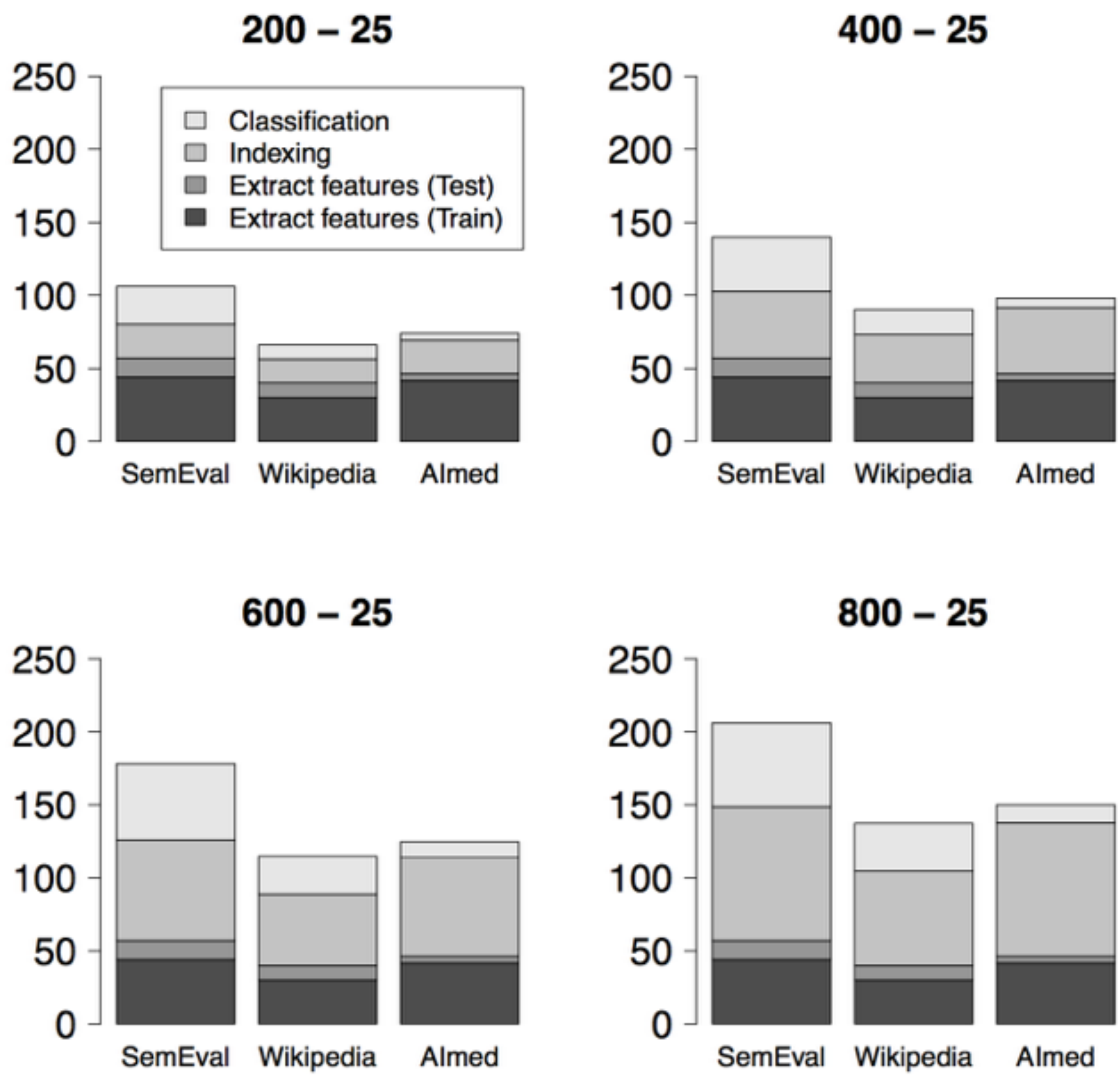
- Currently Deep Learning (DL) techniques dominate most of the research in RE (and in other NLP fields)
- Mostly DL are supervised approaches requiring labeled datasets for training, which is always a bottleneck.
- I believe future RE research needs to explore techniques that combine semi-supervised or distantly supervised methods together with the new Deep Learning approaches.
- Allow to efficiently extract many different types of relationship from large document collections such as the Web.

Addendum

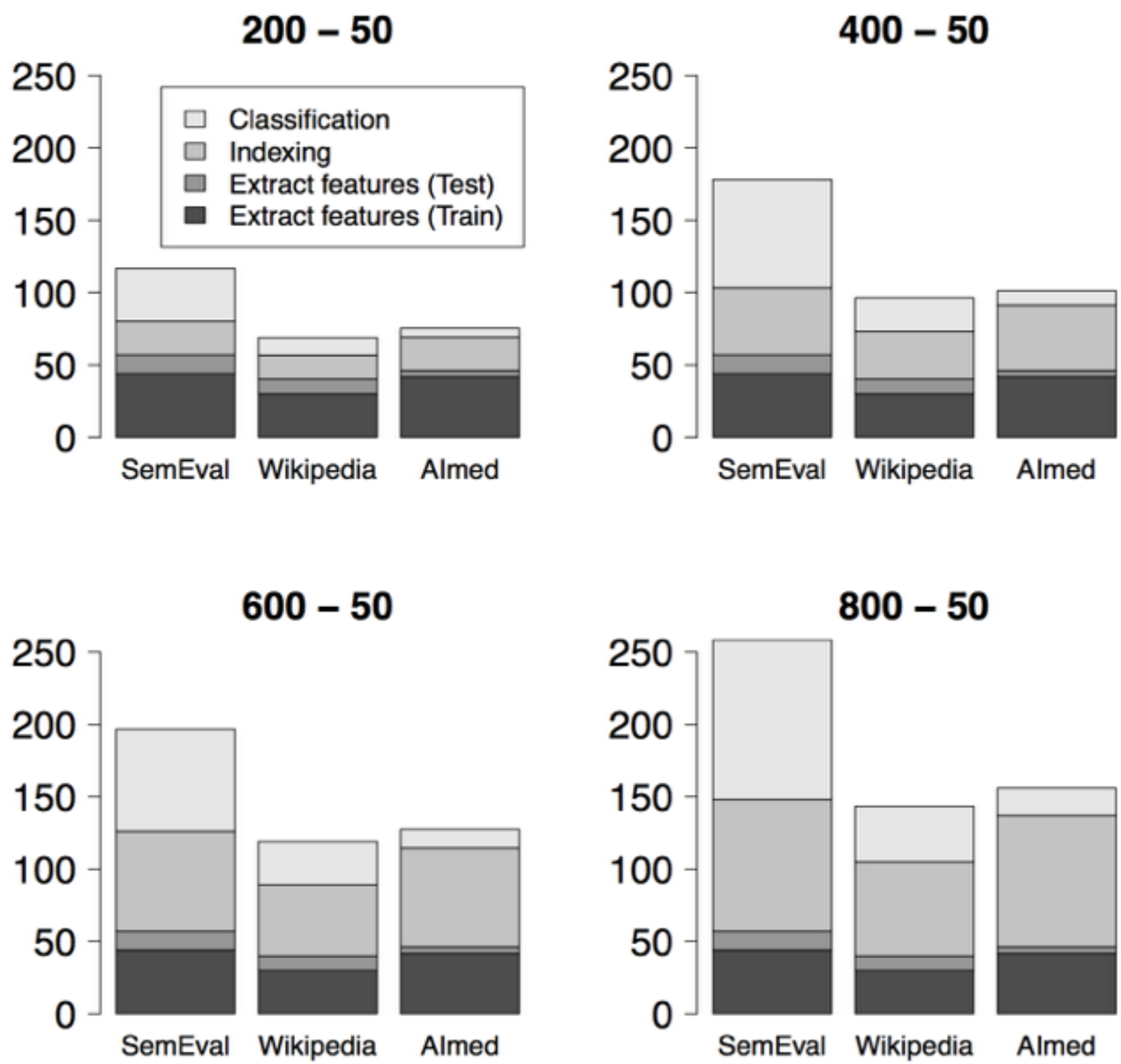
Results for the English datasets

	Sigs./Bands	1 kNN P	1 kNN R	1 kNN F_1	3 kNN P	3 kNN R	3 kNN F_1	5 kNN P	5 kNN R	5 kNN F_1	7 kNN P	7 kNN R	7 kNN F_1
SemEval	200/25	0.662	0.622	0.641	0.683	0.642	0.662	0.698	0.652	0.674	0.698	0.637	0.666
	200/50	0.662	0.621	0.640	0.683	0.643	0.662	0.698	0.651	0.673	0.698	0.636	0.666
	400/25	0.664	0.636	0.650	0.685	0.668	0.676	0.708	0.672	0.690	0.691	0.667	0.679
	400/50	0.663	0.635	0.649	0.684	0.664	0.674	**0.708**	0.674	**0.690**	0.694	0.670	0.682
	600/25	0.657	0.631	0.644	0.677	0.660	0.669	0.697	0.674	0.685	0.695	0.660	0.677
	600/50	0.657	0.631	0.644	0.676	0.658	0.667	0.699	**0.678**	0.688	0.694	0.664	0.678
	800/25	0.654	0.630	0.642	0.675	0.656	0.665	0.694	0.662	0.678	0.696	0.658	0.677
	800/50	0.654	0.632	0.643	0.677	0.658	0.667	0.698	0.665	0.681	0.696	0.658	0.676
Wikipedia	200/25	0.410	0.336	0.369	0.434	0.335	0.378	0.439	0.310	0.363	0.489	0.323	0.389
	200/50	0.409	0.336	0.369	0.435	0.336	0.379	0.440	0.310	0.364	0.489	0.321	0.387
	400/25	0.453	0.350	0.394	0.472	0.354	0.405	0.507	0.348	0.413	0.485	0.323	0.388
	400/50	0.450	0.349	0.393	0.468	0.354	0.403	0.503	0.350	0.412	0.509	0.328	0.399
	600/25	0.419	0.344	0.378	0.439	0.352	0.391	0.492	0.364	0.419	0.522	**0.365**	**0.430**
	600/50	0.419	0.343	0.377	0.444	0.354	0.394	0.485	0.353	0.408	**0.532**	0.353	0.425
	800/20	0.416	0.344	0.377	0.431	0.348	0.385	0.493	0.351	0.410	0.513	0.343	0.411
	800/50	0.419	0.345	0.378	0.433	0.350	0.387	0.515	0.346	0.414	0.517	0.338	0.409
AImed	200/25	0.405	0.545	0.465	0.430	0.509	0.466	0.480	0.484	0.482	0.507	0.460	0.482
	200/50	0.405	0.545	0.465	0.430	0.509	0.466	0.480	0.484	0.482	0.507	0.460	0.482
	400/25	0.420	0.589	0.491	0.451	0.554	0.497	0.481	0.524	0.501	0.516	0.502	0.509
	400/50	0.420	0.588	0.490	0.455	0.561	0.502	0.484	0.529	0.505	**0.519**	0.505	0.512
	600/25	0.409	0.605	0.488	0.445	0.571	0.500	0.475	0.529	0.500	0.511	0.513	0.512
	600/50	0.409	0.605	0.488	0.445	0.571	0.500	0.475	0.530	0.501	0.511	0.513	0.512
	800/25	0.416	0.613	0.496	0.453	0.595	0.514	0.481	0.547	0.512	0.490	0.512	0.501
	800/50	0.418	**0.614**	0.498	0.454	0.596	**0.515**	0.482	0.545	0.511	0.489	0.514	0.501

MuSICo: processing times (seconds)



MuSICo: processing times (seconds)



MuSico: results for SemEval 2010

Relationship	Direction	Instances (train/test)	Asymmetrical Precision	Asymmetrical Recall	Asymmetrical F_1	Symmetrical Precision	Symmetrical Recall	Symmetrical F_1
Cause-Effect	(e_1,e_2)	344/134	0.843	0.843	0.843	0.798	0.902	0.847
	(e_2,e_1)	659/194	0.735	0.902	0.810			
Component-Whole	(e_1,e_2)	470/162	0.572	0.759	0.653	0.628	0.670	0.648
	(e_2,e_1)	150/129	0.609	0.520	0.561			
Entity-Destination	(e_1,e_2)	844/291	0.744	0.911	0.819	0.747	0.901	0.817
	(e_2,e_1)	1/1	1.000	0.000	0.000			
Entity-Origin	(e_1,e_2)	568/211	0.789	0.815	0.802	0.756	0.795	0.775
	(e_2,e_1)	148/47	0.667	0.723	0.694			
Product-Producer	(e_1,e_2)	323/108	0.670	0.602	0.634	0.673	0.589	0.628
	(e_2,e_1)	394/123	0.654	0.569	0.609			
Member-Collection	(e_1,e_2)	78/32	0.778	0.438	0.560	0.767	0.777	0.772
	(e_2,e_1)	612/201	0.776	0.791	0.783			
Message-Topic	(e_1,e_2)	490/210	0.751	0.733	0.742	0.778	0.778	0.778
	(e_2,e_1)	144/51	0.750	0.706	0.727			
Content-Container	(e_1,e_2)	374/153	0.726	0.778	0.751	0.706	0.802	0.751
	(e_2,e_1)	166/39	0.627	0.821	0.711			
Instrument-Agency	(e_1,e_2)	97/22	0.429	0.545	0.480	0.605	0.667	0.634
	(e_2,e_1)	407/134	0.615	0.679	0.645			
Other	—	1 410/454	—	—	—	0.442	0.293	0.352
Macro-average	—	—	0.708	0.674	0.690	0.718	0.764	0.740

Results for DBPediaRelations-PT

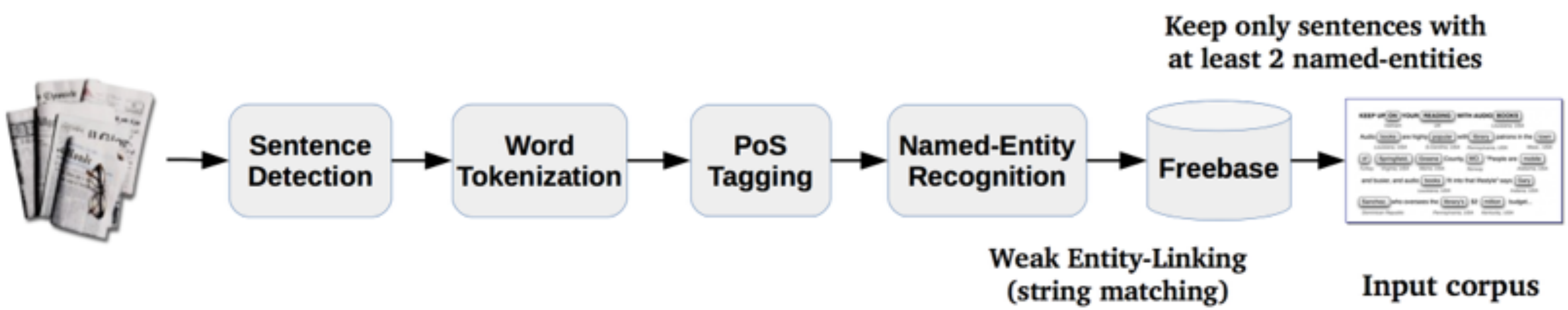
	Sigs./Bands	1 kNN P	1 kNN R	1 kNN F_1	3 kNN P	3 kNN R	3 kNN F_1	5 kNN P	5 kNN R	5 kNN F_1	7 kNN P	7 kNN R	7 kNN F_1
Set I	200/25	0.492	0.400	0.441	0.627	0.426	0.507	0.716	0.423	0.532	0.724	0.429	0.539
	200/50	0.489	0.400	0.440	0.625	0.425	0.506	0.716	0.423	0.532	0.726	0.430	0.540
	400/25	0.476	0.405	0.438	0.559	0.418	0.478	0.724	0.434	0.543	**0.736**	**0.443**	**0.553**
	400/50	0.474	0.405	0.437	0.557	0.423	0.481	0.715	0.434	0.540	0.731	0.441	0.550
	600/25	0.609	0.435	0.508	0.645	0.437	0.521	0.688	0.440	0.537	0.663	0.440	0.529
	600/50	0.583	0.435	0.498	0.646	0.437	0.521	0.686	0.433	0.531	0.719	0.441	0.547
	800/25	0.545	0.426	0.478	0.610	0.430	0.504	0.651	0.434	0.521	0.640	0.442	0.523
	800/50	0.541	0.423	0.475	0.611	0.432	0.506	0.652	0.436	0.523	0.643	0.444	0.525
Set II	200/25	0.476	0.414	0.443	0.628	0.437	0.515	0.713	0.429	0.536	0.718	0.432	0.539
	200/50	0.474	0.414	0.442	0.628	0.437	0.515	0.713	0.429	0.536	0.718	0.432	0.539
	400/25	0.499	0.417	0.454	0.563	0.430	0.488	0.725	0.437	0.545	0.729	0.442	0.550
	400/50	0.497	0.417	0.453	0.565	0.436	0.492	0.674	0.440	0.532	0.729	0.443	0.551
	600/25	0.580	0.425	0.491	0.640	0.442	0.523	0.669	0.439	0.530	0.728	0.435	0.545
	600/50	0.553	0.425	0.481	0.641	0.442	0.523	0.724	0.439	0.547	0.728	0.441	0.549
	800/25	0.549	0.424	0.479	0.615	0.433	0.508	0.720	0.443	0.549	**0.736**	0.441	**0.551**
	800/50	0.549	0.424	0.479	0.615	0.433	0.508	0.712	**0.447**	0.549	0.731	0.438	0.548
Set III	200/25	0.477	0.403	0.437	0.628	0.431	0.511	0.720	0.432	0.540	0.723	0.438	0.546
	200/50	0.478	0.404	0.438	0.628	0.431	0.511	0.666	0.432	0.524	0.670	0.438	0.530
	400/25	0.522	0.431	0.472	0.574	0.432	0.493	0.732	0.446	0.554	0.731	0.442	0.551
	400/50	0.522	0.431	0.472	0.578	0.441	0.500	0.679	0.446	0.538	0.732	0.445	0.554
	600/25	0.581	0.427	0.492	0.630	0.432	0.513	0.673	0.446	0.536	0.677	0.441	0.534
	600/50	0.554	0.427	0.482	0.631	0.432	0.513	0.726	0.439	0.547	0.731	0.442	0.551
	800/25	0.548	0.426	0.479	0.616	0.435	0.510	0.721	**0.449**	0.553	**0.733**	0.447	**0.555**
	800/50	0.545	0.423	0.476	0.620	0.446	0.519	0.721	0.445	0.550	0.732	0.446	0.554
Set IV	200/25	0.472	0.404	0.435	0.629	0.436	0.515	0.724	0.436	0.544	0.723	0.440	0.547
	200/50	0.474	0.404	0.436	0.575	0.436	0.496	0.671	0.436	0.529	0.670	0.440	0.531
	400/25	0.521	0.429	0.471	0.572	0.429	0.490	0.730	0.443	0.551	0.731	0.441	0.550
	400/50	0.521	0.429	0.471	0.573	0.436	0.495	0.680	0.447	0.539	**0.732**	0.444	0.553
	600/25	0.579	0.423	0.489	0.628	0.429	0.510	0.673	0.446	0.536	0.678	0.437	0.531
	600/50	0.552	0.423	0.479	0.629	0.428	0.509	0.728	0.446	0.553	0.731	0.438	0.548
	800/25	0.547	0.423	0.477	0.616	0.433	0.509	0.715	0.445	0.549	0.723	0.444	0.550
	800/50	0.544	0.420	0.474	0.618	0.439	0.513	0.716	0.444	0.548	0.731	**0.449**	**0.556**

- Set I: Quadgrams
- Set II: Quadgrams + Verbs
- Set III: Quadgrams + Verbs + Prepositions
- Set III: Quadgrams + Verbs + Prepositions + ReVerb Patterns

MuSico: results for DBPediaRelations-PT

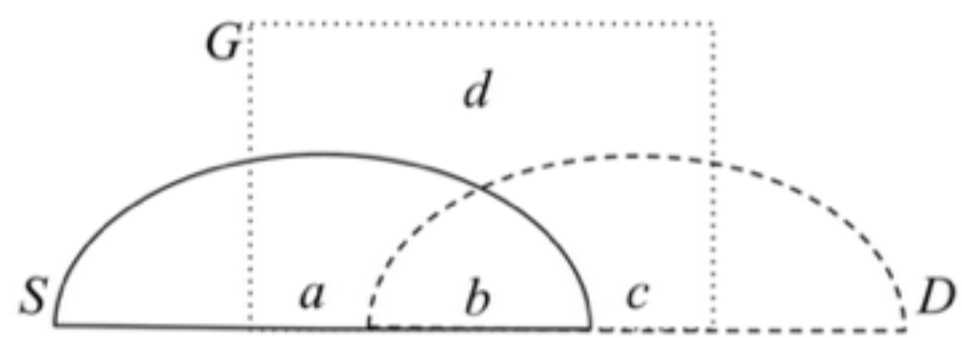
Relationship	Direction	Instances (train/test)	Assymetrical P	Assymetrical A	Assymetrical F_1	Symmetrical P	Symmetrical A	Symmetrical F_1
local-de-enterro-ou-falecimento	(e_1,e_2)	4 788/1 596	0.802	0.595	0.683	0.806	0.574	0.671
	(e_2,e_1)	257/85	0.375	0.035	0.065			
influenciado-por	(e_1,e_2)	84/28	0.000	0.000	0.000	0.000	0.000	0.000
	(e_2,e_1)	26/9	1.000	0.111	0.199			
pessoa-chave-em	(e_1,e_2)	106/35	0.500	0.086	0.146	0.233	0.079	0.117
	(e_2,e_1)	161/53	0.200	0.113	0.145			
localizado-em	(e_1,e_2)	33 639/11 213	0.916	0.929	0.922	0.924	0.922	0.923
	(e_2,e_1)	1 038/346	0.395	0.087	0.142			
origem-de	(e_1,e_2)	16 784/5 594	0.723	0.806	0.807	0.733	0.908	0.811
	(e_2,e_1)	965/321	0.664	0.567	0.612			
antepassado-de	(e_1,e_2)	151/50	0.471	0.800	0.593	0.545	0.727	0.623
	(e_2,e_1)	49/16	0.000	0.000	0.000			
parte-de	(e_1,e_2)	2 590/863	0.541	0.544	0.543	0.680	0.576	0.623
	(e_2,e_1)	1 267/422	0.574	0.275	0.372			
sucessor-de	(e_1,e_2)	117/39	0.400	0.051	0.091	0.541	0.161	0.248
	(e_2,e_1)	255/85	0.359	0.165	0.226			
parceiro	—	96/32	—	—	—	0.600	0.188	0.286
não-relacionado	—	4 831/1 610	—	—	—	0.767	0.543	0.636
Macro-Average	—	—	0.516	0.333	0.405	0.583	0.468	0.494
Accuracy	—	—		0.813			0.834	

BREDS / TREMoSSo NLP Pipeline



- Python NLTK 3.0: Sentence segmentation, tokenisation and PoS-tagging
- Stanford NER 3.5.2 (Finkel et al., 2005)
- Word embeddings were computed with the skip-gram model (Mikolov et al., 2013a) using the *word2vec* implementation
    - Skip-length = 5 tokens
    - Vectors = 200 dimensions

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 $\mathrm{PPMI}(\mathrm{e}_1, \mathrm{rel}, \mathrm{e}_2) = \frac{\mathrm{count}(\mathrm{e}_1 \text{ NEAR:}X \; rel \text{ NEAR:}X \; \mathrm{e}_2)}{\mathrm{count}(\mathrm{e}_1 \text{ AND } \mathrm{e}_2)}$

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)