Measuring the Similarity between Implicit Semantic Relations from the Web

Danushka Bollegala, Yutaka Matsuo, Mitsuru Ishizuka

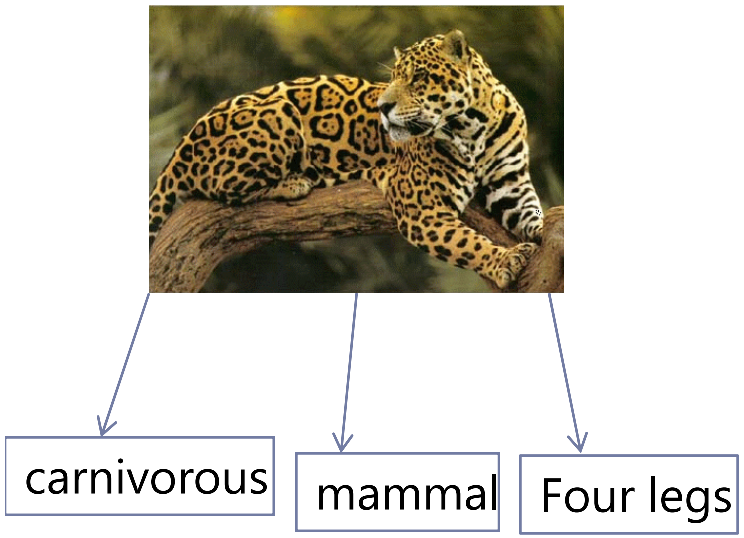
18th International World Wide Web Conference, 2009.
Madrid, Spain

THE UNIVERSITY OF TOKYO

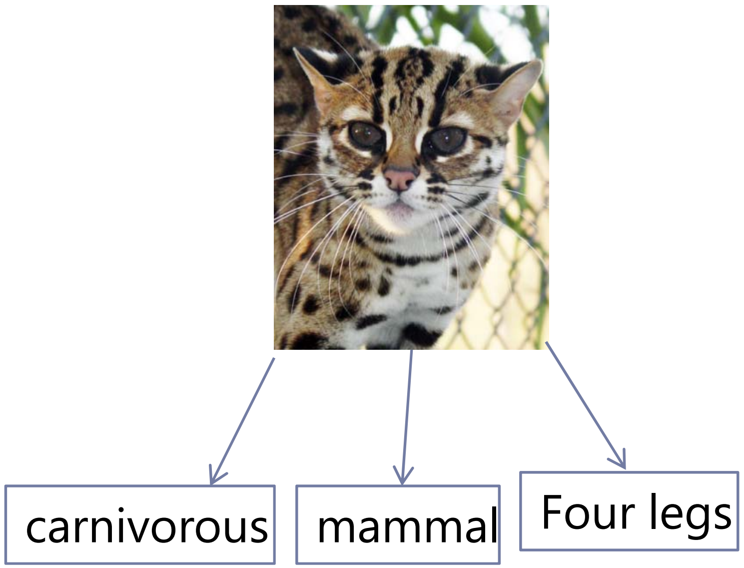
Attributional vs. Relational Similarity

Attributional Similarity is the correspondence between the attributes of two objects

Jaguar



cat



A high degree of attributional similarity exists between Jaguar and cat : sim(X,Y)

Attributional vs. Relational Similarity

Relational similarity is the correspondence between the relations that exist between two pairs of objects

(ostrich, bird)

Ostrich is a large bird

(lion, cat)

Lion is a large cat

A high degree of relational similarity exists between the two object pairs

sim(A,B,X,Y)

Applications of Relational Similarity

- Recognizing Analogies (Turney ACL 2006)
  - (traffic, road) vs. (water, pipe)

X *flows in* Y

- Semantic Relation Classification
  - (Natase & Szpakowicz 2003)
  - laser printer (*instrument*), concert hall (*purpose*), student discount (*benificiary*)
- *Implicit* Relation extraction
  - Given a word pair (A,B) for which relation R holds, and a word C, find a word D s.t. (A,B) and (C,D) are analogous.
    - (A,B)=(Christianity, Bible), C=Muslim => D=Qur'an

Analogy making in AI

- Structure Mapping Theory (SMT) (Gentner, *Cognitive Science* '83)
  - Analogy is a mapping of knowledge from one domain (the base) into another (the target) which conveys that a system of relations known to hold in the base also holds in the target.
- Mapping rules: $M: b_i \rightarrow t_i$
  - Attributes of objects are dropped
    - $RED(b_i) \not\rightarrow RED(t_i)$
  - Certain relations between objects in the base are mapped to the target
    - REVOLVES(EARTH,SUN) → REVOLVES(ELECTRON,NEUCLEUS)
  - ***systematicity principle***: base predicate that belongs to a mappable system of mutually constraining interconnected relations is more likely to be mapped to the target domain.
    - $CAUSE[PUSH(b_i,b_j), COLLIDE(b_j,b_k)] \rightarrow CAUSE[PUSH(t_i,t_j), COLLIDE(t_j,t_k)]$

Challenges in Measuring Relational Similarity

- **How to explicitly state the relation between two entities?**
- **How to extract the multiple relations between two entities?**
  - Extract lexical patterns from contexts where the two entities co-occur
- **A single semantic relation can be expressed by multiple patterns.**
  - E.g. "ACQUISITION": X *acquires* Y, Y *is bought by X*
  - Cluster the semantically related lexical patterns into separate clusters.
- **Semantic Relations might not be independent.**
  - E.g. IS-A and HAS-A. Ostrich is a bird, Ostrich has feathers
  - Measure the correlation between various semantic relations
    - Mahalanobis Distance vs. Euclidian Distance
- **The contribution of different semantic relations towards relational similarity is unknown**
  - Learn the contribution of different semantic relations using training data
    - Information Theoretic Metric Learning (ITML) (Davis 2008)

How to explicitly state the relations between the two words in a word pair?

Pattern Extraction

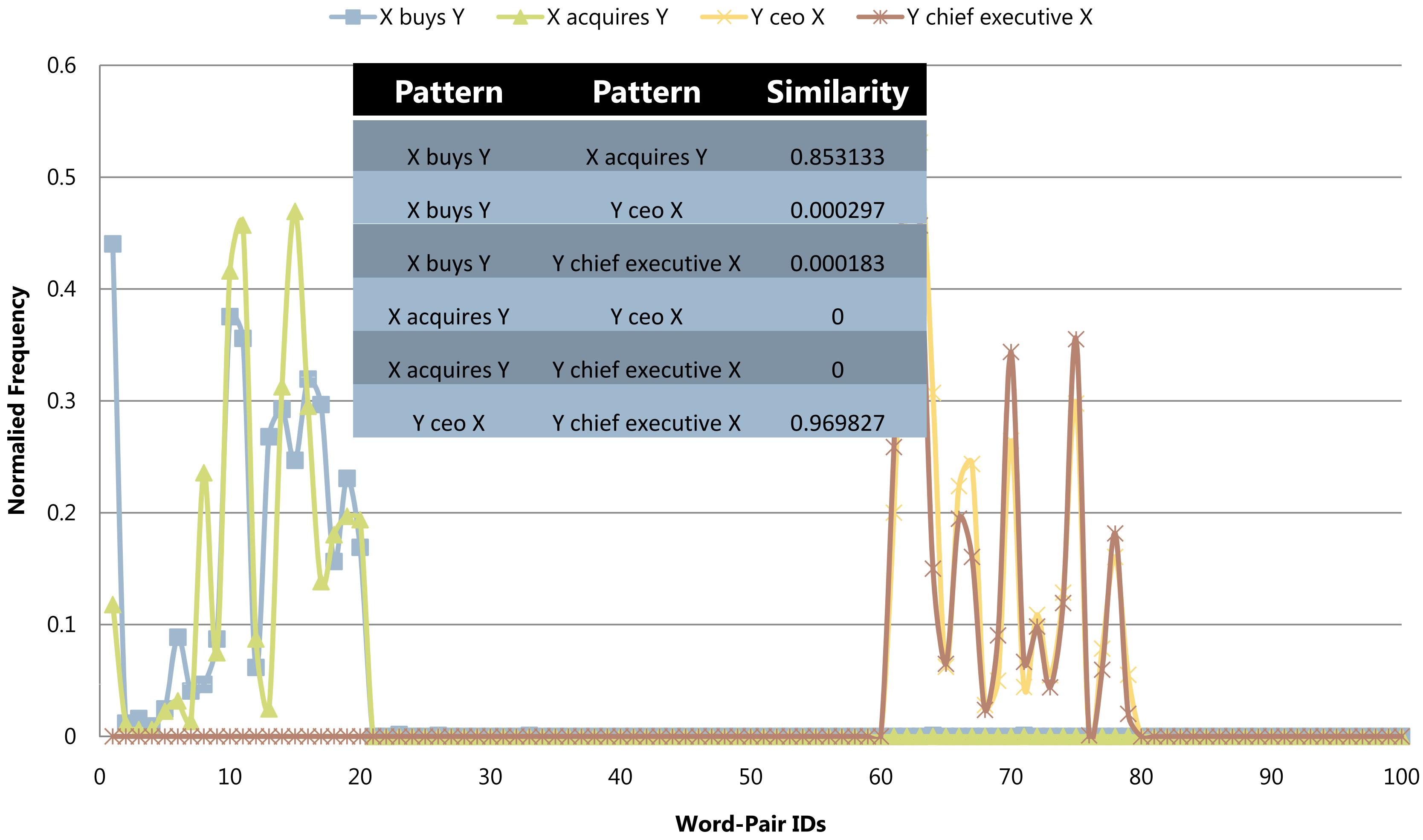
- We use prefix-span, a sequential pattern mining algorithm, to extract patterns that describe various relations, from text snippets returned by a web search engine.
- query = lion * * * * * * * cat
- snippet = .. lion, a large heavy-built social cat of open rocky areas in Africa ..
- patterns = X, *a large* Y / X a large Y / X a Y / X a large Y of
- Prefix span algorithm is used to extract patterns because:
  - It is efficient
  - It can considers gaps
- Extracted patterns can be noisy:
  - misspellings, ungrammatical sentences, fragmented snippets

How to identify the different patterns that talk about the same semantic relation?

Clustering the Lexical Patterns

- We have ca. 150,000 patterns that occur more than twice in the corpus that express various semantic relations
- However, a single semantic relation is expressed by more than one lexical patterns
- How to identify the patterns that express a particular semantic relation?
  - Distributional Hypothesis (Harris 1957)
  - *Patterns that are equally distributed among word-pairs are semantically similar*
- We can cluster the patterns according to their distribution in word-pairs
  - Pair-wise comparison is computationally expensive
  - Propose a sequential pattern clustering algorithm

Distribution of patterns in word-pairs



Pattern	Pattern	Similarity
X buys Y	X acquires Y	0.853133
X buys Y	Y ceo X	0.000297
X buys Y	Y chief executive X	0.000183
X acquires Y	Y ceo X	0
X acquires Y	Y chief executive X	0
Y ceo X	Y chief executive X	0.969827

Greedy Sequential Clustering

1. Sort the patterns according to their total frequency in all word-pairs
2. Select the next pattern:
    1. Measure the similarity between each of the existing clusters and the pattern
    2. If the similarity with the most similar cluster is greater than a threshold θ, then add to that cluster, otherwise form a new cluster with this pattern.
    3. Repeat until all patterns are clustered.
3. We view each cluster as a vector of word-pair frequencies and compute the cosine similarity between the centroid vector and the pattern.

- Properties of the clustering algorithm
    - Scales linearly with the number of patterns O(n)
    - More general clusters are formed ahead of the more specific clusters
    - Only one parameter to be adjusted (clustering threshold θ)
    - No need to specify the number of clusters
    - Does not require pair-wise comparisons, which are computationally costly
    - A greedy clustering algorithm

How to account for the inter-dependence between semantic relations?

How to compute the relational similarity from the pattern clusters?

Computing Relational Similarity

- We represent each word pair by an *N* dimensional feature vector
  - *N*: Total number of clusters
  - *feature value*: total frequency of patterns that belong to a cluster
  - feature vectors are normalized to unit length
- Using a labeled dataset of positive and negative instances, we learn a Mahalanobis distance metric.
  - Mahalanobis distance between two vectors **x** and **y** is defined by,

$$(\mathbf{x}-\mathbf{y})^t A(\mathbf{x}-\mathbf{y})$$

  where A is the Mahalanobis matrix.
- We use the Information Theoretic Metric Learning algorithm (Davis et al. 2007).
  - No eigenvalue or eigenvector computations are required
  - Scalable to large datasets via lower rank approximations
  - Can incorporate slack variables

EXPERIMENTS

Datasets

- ENT dataset
  - We created a dataset that has 100 entity-pairs covering five relation types. (20X5 = 100)
  - **ACQUIRER-ACQUIREE** (e.g. [*Google, YouTube*])
  - **PERSON-BIRTHPLACE** (e.g. [*Charlie Chaplin, London*])
  - **CEO-COMPANY** (e.g. [*Eric Schmidt, Google*])
  - **COMPANY-HEADQUARTERS** (e.g. [*Microsoft, Redmond*])
  - **PERSON-FIELD** (e.g. [*Einstein, Physics*])
- ca. 100,000 snippets are downloaded for each relation type
- SAT word analogy dataset (Turney 2003)
  - 374 SAT word analogy questions (2178 word pairs)
  - Each question has five choices out of one is correct

Relation Classification on ENT Dataset

- We use the proposed relational similarity measure to classify entity pairs according to the semantic relations between them.
- We use k-nearest neighbor classification (*k*=10)
  - For each entity pair in the ENT dataset, assign the relation type of the most relationally similar *k* entity pairs.
  - Repeat the above process for all entity pairs in the dataset
- Evaluation measure:

$$\text{Average Precision} = \frac{\sum_{r=1}^{k} \text{Precision}(r) \times \text{Relevant}(r)}{\text{No. of relevant pairs}}$$

Results – Relation Classification Task

Relation	VSM	LRA	EUC	PROPOSED
ACQUIRER-ACQUIREE	92.7	92.24	91.47	94.15
COMPANY-HEADQARTERS	84.55	82.54	79.86	86.53
PERSON-FIELD	44.70	43.96	51.95	57.15
CEO-COMPANY	95.82	96.12	90.58	95.78
PERSON-BIRTHPLACE	27.47	27.95	33.43	36.48
OVERALL	68.96	68.56	69.46	74.03

Comparison with baselines and previous work

VSM: Vector Space Model (cosine similarity between pattern frequency vectors)
LRA: Latent Relational Analysis (Turney '06 ACL, Based on LSA)
EUC: Euclidean distance between cluster vectors
PROPOSED: Proposed method (Learned Mahalanobis distance between entity-pairs)

Pattern Clusters

Cluster 1 (2868)	**X** acquires **Y**	**X** has acquired **Y**	**X**'s **Y** acquisition	**X**, acquisition, **Y**	**Y** goes **X**
Cluster 2 (2711)	**Y** legend **X** was	**X**'s championship **Y**	**Y** star **X** was	**X** autographed **Y** ball	**Y** start **X** robbed
Cluster 3 (2615)	**Y** champion **X**	world **Y** champion **X**	**X** teaches **Y**	**X**'s greatest **Y**	**Y** players like **X**
Cluster 4 (2008)	**X** to buy **Y**	**X** and **Y** confirmed	**X** buy **Y** is	**Y** purchase to boost **X**	**X** is buying **Y**
Cluster 5 (2002)	**Y** founder **X**	**Y** founder and CEO **X**	**X**, founder of **Y**	**X** says **Y**	**X** talks up **Y**
Cluster 6 (1364)	**X** revolutionized **Y**	**X** professor of **Y**	in **Y** since **X**	ago, **X** revolutionized **Y**	**X**'s contribution to **Y**
Cluster 7 (845)	**X** and modern **Y**	genius: **X** and modern **Y**	**Y** in DDDD, **X** was	on **Y** by **X**	**X**'s lectures on **Y**
Cluster 8 (280)	**X** headquarters in **Y**	**X** offices in **Y**	past **X** offices in **Y**	the **X** conference in **Y**	**X** headquarters in **Y** on
Cluster 9 (144)	**X**'s childhood in **Y**	**X**'s birth in **Y**	**Y** born **X**	**Y** born **X** introduced the	sobbing **X** left **Y** to
Cluster 10 (49)	**X** headquarters in **Y** **.**	**X**'s **Y** headquarters	**Y** – based **X**	**X** works with the **Y**	**Y** office of **X**

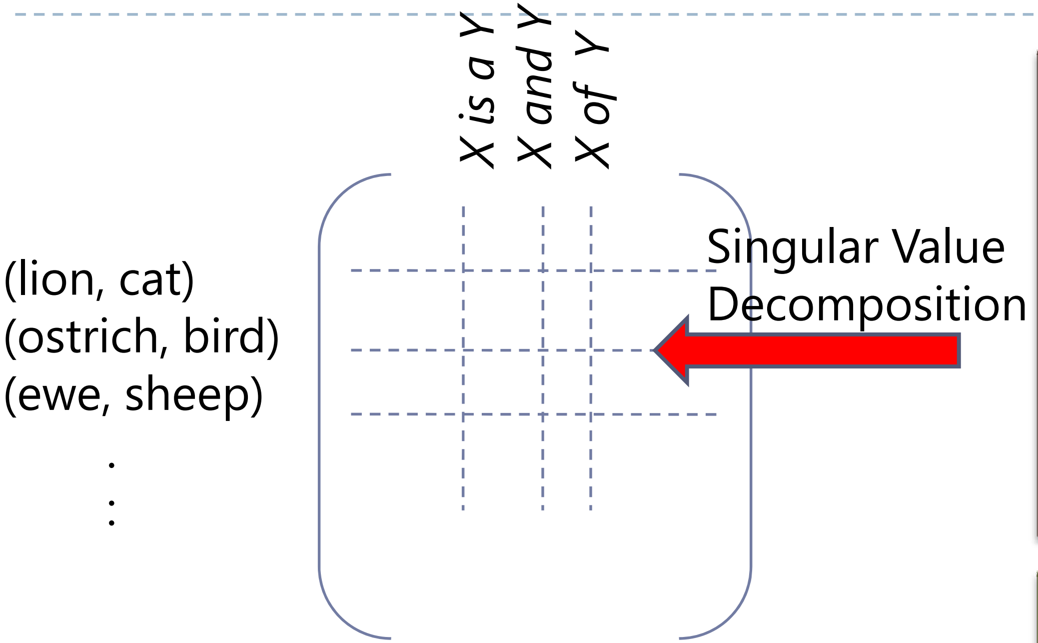
Solving Word Analogies on SAT Dataset

Algorithm	SAT score	Algorithm	SAT score
Random guessing	0.200	LSA+Predictation	0420
Jiang & Conrath	0.273	Veale (WordNet)	0.430
Lin	0.273	Bicici & Yuret	0.440
Leacock & Chodrow	0.313	VSM	0.470
Hirst & St.-Onge	0.321	PROPOSED	0.511
Resnik	0.332	Pertinence	0.535
PMI-IR (Turney 2003)	0.35	LRA (Turney 2006)	0.561
SVM (Bollegala ECAI)	0.401	Human	0.570

less than 6 hours

8 days!!!

Latent Relational Analysis vs. The Proposed Method



- To compute relational similarity between two word-pairs using *N* number of lexical patterns, LRA requires 2*N* web-queries ($N \approx 4000$)
- Proposed method requires only two web-queries and is independent of the number of patterns!

- In LRA, for each new word-pair, we must repeat SVD
- No SVD is required

(A′ , B′)
(A, B)

vs.

(C′, D′)
(C, D)

→ RelSim(A,B,C,D)+RelSim(A′,B′,C′,D′)

Conclusions

- Distributional similarity is useful to identify semantically similar lexical patterns
- Clustering lexical patterns prior to measuring similarity improves performance
- Greedy sequential clustering algorithm efficiently produces pattern clusters for common semantic relations
- Mahalanobis distance outperforms Euclidean distance when measuring similarity between semantic relations
- Future Work
  - Use relational similarity to analogical search

Thank You

Contact: Danushka Bollegala
danushka@mi.ci.i.u-tokyo.ac.jp

http://www.miv.t.u-tokyo.ac.jp/danushka
The University of Tokyo, Japan.