web intelligence technologies

Web People Search Disambiguation using Random Walks

José Iria

Intelligent Web Technologies Lab

Natural Language Processing Group

The University of Sheffield

The University Of Sheffield.

Talk Outline

- Web People Search Disambiguation Task
- Related Work
- Random Walks
- Commute Time Distance
- Clustering
- Experiments
- Results
- Conclusions

WEPS Disambiguation

- Person names are highly ambiguous
  - only 90.000 thousand different names are shared by 100 million people according to the U.S. Census Bureau.
- Using a search engine is for WEPS is far from ideal.
  - many-to-many mapping of person names to the actual persons
- Goal: disambiguate the several referents in web pages returned by a query for a given person name.
- The Semeval'07 Web People Search challenge (Artiles et al., 2007) formally evaluated systems on this task.

The University Of Sheffield.

Past Approaches

- Bagga & Baldwin, 1998
  - unsupervised model based on sentence comparison
  - sentences where entity is mentioned parsed into BOW feature vector and compared using the cosine similarity measure
  - entities in sentence pairs whose similarity scores are above a certain threshold are predicted as referring to the same entity
- Mann & Yarowsky, 2003
  - problem treated as a clustering task
  - BOW representation enhanced with biographic information extracted from the text via automatically induced extraction patterns (using the Web as a corpus)
  - use a bottom-up centroid agglomerative clustering algorithm

The University Of Sheffield.

Past Approaches (2)

- Al-Kamha & Embley, 2004
  - BOW + biographic info + structural similarity features through link analysis of the web graph
  - learn probabilistic model over training data
- Graph-based approaches:
  - Bekkerman & Mccallum, 2005: link structure analysis in the context of a social network graph
  - Han et al., 2005: k-way spectral clustering approach over an author citation graph, using cosine TF-IDF distance in undirected graph
  - Bekkerman et al., 2007: heuristic search methods over web graph
  - Minkov et al., 2006: lazy graph walk approach to rank closed set of persons given a name mention in an email corpus

The University Of Sheffield.

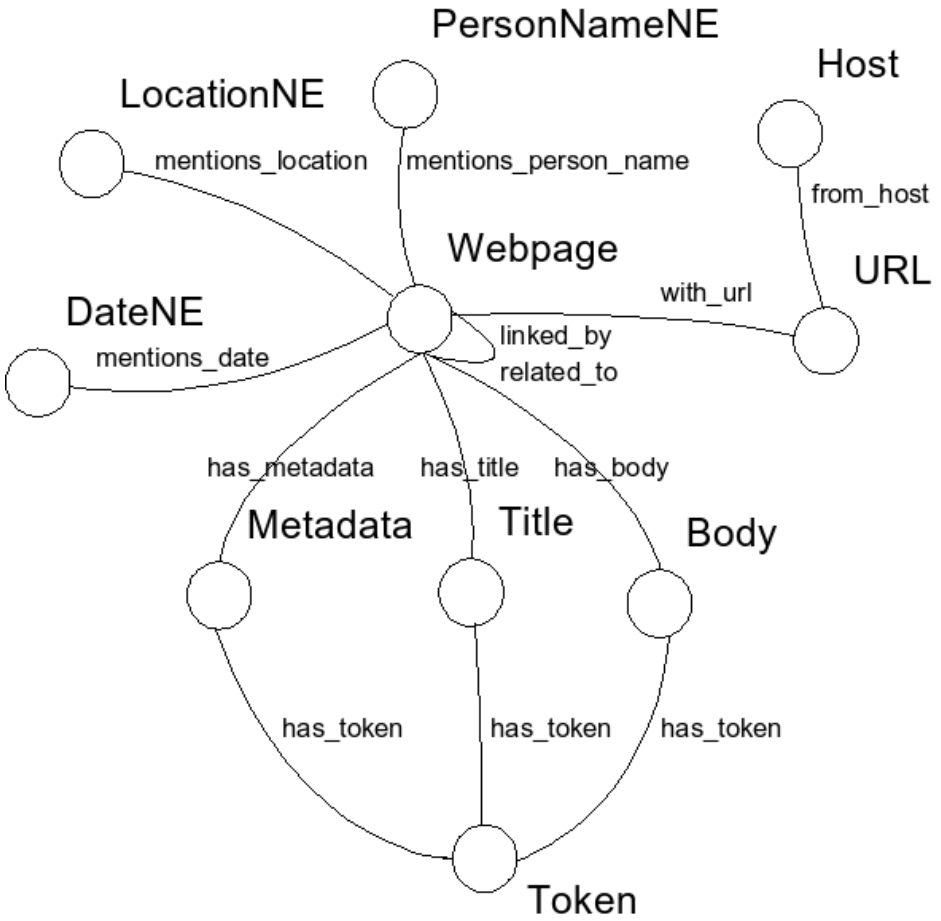
Our Approach

- A random walks-based approach:
  - 1) uses a graph to model the web pages returned by the search engine query,
  - 2) discards irrelevant web pages using a few simple hand-crafted heuristics,
  - 3) computes a similarity matrix for web pages using random walks over the graph, and
  - 4) finally clusters the web pages given the similarity matrix.
- Seamlessly combines different types of features:
  - models page content + attributes + network topological features
  - elegantly arrive at one single measure of similarity between any two webpages in the graph

Graph Representation of WEPS

- Undirected weighted typed graph derived from the corpus.
- The graph is a 5-tuple G = (V, E, t, I, w):
  - V is the set of nodes
  - E : V ×V is the set of edges
  - t : V → T is the node type function (T is a set of types)
  - I : E → L is the edge label function (L is a set of labels)
  - w : L → R is the label weight function

Graph Representation of WEPS (2)



Using Random Walks for WEPS

- Goal: determine the similarity matrix between any two nodes of type *Webpage*
  - input to clustering step
- Assumption – two pages talk about the same person if:
  - similar discourse
  - overlap between entities mentioned
  - coming from the same web host or having similar URLs
  - closer to each other in the network of linked pages
- Intuitively:
  - the “harder” it is for a drunkard to arrive from webpage Y starting from webpage X, the less similar the two pages are.

Using Random Walks for WEPS (2)

- We build an adjacency matrix:

$$W_{ij} = \begin{cases} \sum_{l_k \in L} \frac{w(l_k)}{|(i,\cdot) \in E : l(i,\cdot) = l_k|}, & (i,j) \in E \\ 0, & otherwise \end{cases}$$

- i.e., distributes uniformly weights of edges of same type leaving node
- Weights are trained by maximizing score on the gold standard
  - via simulated annealing
- Build row stochastic matrix $\mathbf{D^{-1}W}$, where $D_{ii} = \sum_k W_{ik}$
- Similarity given by *t*-step transition probability

$$P^{(t)}(j|i) = [(D^{-1}W)^t]_{ij}$$

- Not interested in stationary distribution, so use small *t*

The University Of Sheffield.

Commute Time Distance

- Euclidean Commute Time (ECT) distance (Saerens et al., 2004)
  - also based on a random walk model
  - interesting properties:
    - decreases when the number of paths connecting two nodes increases
    - decreases when the length of any path decreases
    - non-parametric
- Average first-passage time:
  - average number of steps a random walker, starting in state *i*, will take to enter state *j* for the first time.
- Average commute time:
  - average number of steps from *i* to *j* and back to *i*.

The University Of Sheffield.

Intuitive Idea

- *t*-step of the random walk:
  - generalizing the typical TF-IDF measure to a measure that takes in all features represented in the graph simultaneously
  - if graph is reduced to contain only tokens and wepages and t=2, original TF-IDF is recovered
- ECT distance
  - connections to PCA and spectral theory (Saerens et al., 2004)
  - “booster” to the simple clustering techniques
  - shown to be competitive with state-of-the-art algorithms such as spectral clustering (Luh Yen et al., 2007) when coupled with a simple clustering algorithm

Simple Aglomerative Clustering

- Idea: at any given iteration, greedly merge the two most similar clusters
- Start with clusters of 1 element
- Also tried with spectral clustering and k-means

Input: symmetric similarity matrix S, threshold θ
Output: a set of clusters C
1. (i, j) ← find min score in S
2. if Sij > θ then exit
3. place i and j in the same cluster in C (merging existing clusters of i and j if needed)
4. (average pairs of edges connecting to nodes i,j from any node k)
4a. Sik ← (Sik + Sjk)/2, k = i, j
4b. Ski ← (Ski + Skj)/2, k = i, j
5. remove j-th column and j-th line from S (effectively merging nodes i,j into a single node)
6. goto 1
7. return clusters C

Clustering Threshold Problem

- Difficult to determine when to stop clustering
  - equivalent to number of clusters problem in k-means
- Solutions tried:
  - Calinski&Harabasz stopping rule
  - Eigengap measure (for spectral clustering)

The University Of Sheffield.

Experiments

- Semeval'07 Web People Search competition:
  - provided several sets of 100 web pages each
    - built by querying a search engine with combinations of common given and family names
  - goal: cluster pages that refer to the same person
  - multiple assignments allowed
- Training and test data provided by organizers, consisting of:
  - contents of the pages in HTML, unfiltered
  - search engine data for each page, such as page rank and related pages (dropped later on)
  - gold standard clusters (for the training data)

Corpus Pre-Processing

- Text → Graph:
  - process *metadata, title* and *body* seperately
  - window of *n* characters around mentions of entity in question
  - *Token* nodes from tokenization
    - “clean up” HTML
    - remove stop words and infrequent words (<3)
    - apply Porter’s stemming algorithm
  - NE nodes from named entity recognizer, associate to webpages
  - *linked_by* edges by analysis of HTML <a> tag
  - *related_to* edges from Google
  - *URL* and *Host* nodes from the URL provided for each page

Corpus Pre-Processing (2)

- Some pages not relevant
  - would introduce noise into clusters
- "Safe" heuristics to discard pages:
  - empty pages
  - page contains no mention of entity in question or its simple variants, e.g.:
    - John Smith -> J. Smith
    - John Smith -> Smith, John
    - John Smith -> Smith, J.
    - John Smith -> J. XXX John
    - etc.
  - tested "safeness" on the training data

The University Of Sheffield.

Results

- Possible configurations:
  - Window size around entities and head/tail
  - Number of steps of the random walk
  - Whether to use Commute Time Distance
  - Whether to use Simulated annealing weights
  - Clustering choices:
    - K-means with associated number of clusters problem
    - Simple Aglomerative Clustering with associated threshold problem
    - Spectral Clustering using eigengap heuristic
- Final system: 3000/3000/6000 window size head/tail/entities, t=2, apply ECT, no simulated annealing, simple aglomerative clustering using manually tuned threshold

Results (2)

- Measures used:
  - Purity: analogous to precision; penalizes wrong assignments; hurt by overclustering
  - Inverse purity: analogous to recall; penalizes missing assignments; hurt by underclustering
  - 0.5 F-measure: harmonic mean of the above
  - 0.2 F-measure: same as above, but favoring purity
- Our system:
  - Below Average using 0.5 F-measure: 0.49 < 0.60 (14^{th} place)
  - Below Average using 0.2 F-measure: 0.66 < 0.69 (8^{th} place)
  - 2^{nd} place in terms of inverse purity - tends to overcluster

Results (3)

rank	team-id	aver_f05	aver_f02	aver_pur	aver_inv_pur
1	xxxx	0,78	0,83	0,72	0,88
2	xxxx	0,75	0,77	0,75	0,80
3	xxxx	0,75	0,78	0,73	0,82
4	xxxx	0,67	0,62	0,81	0,60
5	xxxx	0,66	0,73	0,60	0,82
6	xxxx	0,64	0,76	0,53	0,90
7	xxxx	0,62	0,67	0,60	0,73
8	xxxx	0,60	0,73	0,50	0,88
9	xxxx	0,58	0,64	0,55	0,71
10	xxxx	0,58	0,60	0,58	0,64
11	xxxx	0,57	0,71	0,45	0,89
12	xxxx	0,53	0,65	0,45	0,82
13	xxxx	0,50	0,63	0,39	0,83
14	**WIT**	**0,49**	**0,66**	**0,36**	**0,93**
15	xxxx	0,48	0,66	0,35	0,95
16	xxxx	0,40	0,55	0,30	0,91

Best system at Semeval

- CU_COMSEM (University of Colorado)
  - Features:
    - Token level features
      - Local tokens, Full tokens, URL tokens, title tokens in root page
      - TF-IDF weighted token vectors to compute token-level similarity matrix
    - Phrase level features:
      - Local base noun phrases, Named entities
      - Uses SoftTFIDF to compute phrase-level similarity matrix
    - Combines the two similarity matrices at the end
  - Simple single-linkage aglomerative clustering with fixed stop-threshold manually tuned from the training data

The University Of Sheffield.

Summary

- Novel adaptation of clustering using random walks over a graph to the web people search domain
- Design of a particular graph which encodes our assumptions for this kind of problems
- Results on the Semeval'07 WEPS competition
- Future work:
  - Experiments on other datasets
    - Spock challenge

The University Of Sheffield.

References

- Artiles, J., Gonzalo, J., & Sekine, S. (2007). The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. In Proceedings of Semeval 2007, Association for Computational Linguistics.
- Calinski and Harabasz (1974). A Dendrite Method for Cluster Analysis Communications in Statistics, 3(1), 1974, 1-27.
- Erkan, G. (2006). Language model-based document clustering using random walks. Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (pp. 479–486). Association for Computational Linguistics.
- Fleischman, M. B., & Hovy, E. (2004). Multi-document person name resolution. Proceedings of the ACL 2004. Association for Computational Linguistics.
- Guha, R. V., & Garg, A. (2003). Disambiguating People in Search. TAP: Building the Sem Web. ACM Press.
- Luh Yen, Francois Fouss, C. D., Francq, P., & Saerens, M. (2007). Graph nodes clustering based on the commute-time kernel. To appear in the proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2007). Lecture Notes in Computer Science (LNCS).
- Minkov, E., Cohen, W. W., & Ng, A. Y. (2006). Contextual search and name disambiguation in email using graphs. SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 27–34). ACM Press.
- Saerens, M., Fouss, F., Yen, L., & Dupont, P. (2004). The principal components analysis of a graph, and its re lationships to spectral clustering. Proceedings of the 15th European Conference on Machine Learning.

The University Of Sheffield.

Thank You!

- Any questions?

Contact:

j.iria@dcs.shef.ac.uk