

Affinity Measures based on the Graph Laplacian

Delip Rao

Dept. of Computer Science
Johns Hopkins University
delip@cs.jhu.edu

David Yarowsky

Dept. of Computer Science
Johns Hopkins University
yarowsky@cs.jhu.edu

Chris Callison-Burch

Dept. of Computer Science
Johns Hopkins University
ccb@cs.jhu.edu

Abstract

Several language processing tasks can be inherently represented by a weighted graph where the weights are interpreted as a measure of relatedness between two vertices. Measuring similarity between arbitrary pairs of vertices is essential in solving several language processing problems on these datasets. Random walk based measures perform better than other path based measures like shortest-path. We evaluate several random walk measures and propose a new measure based on commute time. We use the pseudo inverse of the Laplacian to derive estimates for commute times in graphs. Further, we show that this pseudo inverse based measure could be improved by discarding the least significant eigenvectors, corresponding to the noise in the graph construction process, using singular value decomposition.

1 Introduction

Natural language data lend themselves to a graph based representation. Words could be linked by explicit relations as in WordNet (Fellbaum, 1989) or documents could be linked to one another via hyperlinks. Even in the absence of such a straightforward representation it is possible to derive meaningful graphs such as the nearest neighbor graphs as done in certain manifold learning methods (Roweis and Saul, 2000; Belkin and Niyogi,

2001). All of these graphs share the following properties:

- They are edge-weighted.
- The edge weight encodes some notion of relatedness between the vertices.
- The relation represented by edges is at least weakly transitive. Examples of such relations include, “is similar to”, “is more general than”, and so on. It is important that the relations selected are transitive for the random walk to make sense.

Such graphs present several possibilities in solving language problems on the data. One such task is, given two vertices in the graph we would like to know how related the two vertices are. There is an abundance of literature on this topic, some of which will be reviewed here. Finding similarity between vertices in a graph could be an end in itself, as in the lexical similarity task, or could be a stage before solving other problems like clustering and classification.

2 Contributions of this paper

The major contributions of this paper are

- A comprehensive evaluation of various random walk based measures
- Propose a new similarity measure based on commute time.
- An improvement to the above measure by eliminating noisy features via singular value decomposition.

© 2008. Licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported* license (<http://creativecommons.org/licenses/by-nc-sa/3.0/>). Some rights reserved.

3 Problem setting

Consider an undirected graph $G(V, E, \mathbf{W})$ with vertices V , edges E , and $\mathbf{W} = [w_{ij}]$ be the symmetric adjacency weight matrix with w_{ij} as the weight of the edge connecting vertices i and j . The weight, $w_{ij} = 0$ for vertices i and j that are not neighbors and when $w_{ij} > 0$ it is interpreted as an indication of relatedness between i and j . In our case, we consider uniformly weighted graphs, i.e., $w_{ij} = 1$ for neighbors but this need not be the case. Let $n = |V|$ be the order of the graph. We define a relation $\text{sim} : V \times V \rightarrow \mathbb{R}^+$ such that $\text{sim}(i, j)$ is the relatedness between vertices i and j . There are several ways to define sim ; the ones explored in this paper are:

- $\text{sim}_G(i, j)$ is the reciprocal of the shortest path length between vertices i and j . Note that this is not a random walk based measure but a useful baseline for comparison purposes.
- $\text{sim}_B(i, j)$ is the probability of a random walk from vertex i to vertex j using all paths of length less than m .
- $\text{sim}_P(i, j)$ is the probability of a random walk from vertex i to vertex j defined via a pagerank model.
- $\text{sim}_C(i, j)$ is a function of the commute time between vertex i and vertex j .

4 Data and Evaluation

We evaluate each of the similarity measure we consider by using a linguistically motivated task of finding lexical similarity. Deriving lexical relatedness between terms has been a topic of interest with applications in word sense disambiguation (Patwardhan et al., 2005), paraphrasing (Kauchak and Barzilay, 2006), question answering (Prager et al., 2001), and machine translation (Blatz et al., 2004) to name a few. Lexical relatedness between terms could be derived either from a thesaurus like WordNet or from raw monolingual corpora via distributional similarity (Pereira et al., 1993). WordNet is an interesting graph-structured thesaurus where the vertices are the words and the edges represent relations between the words. For the purpose of this work, we only consider relations like hypernymy, hyponymy, and synonymy. The importance of this

problem has generated copious literature in the past – see (Pedersen et al., 2004) or (Budanitsky and Hirst, 2006) for a detailed review of various lexical relatedness measures on WordNet. Our focus in this paper is not to derive the best similarity measure for WordNet but to use WordNet and the lexical relatedness task as a method to evaluate the various random walk based similarity measures. Following the tradition in previous literature we evaluate on the Miller and Charles (1991) dataset. This data consists of 30 word-pairs along with human judgements which is a real value between 1 and 4. For every measure we consider, we derive similarity scores and compare with the human judgements using the Spearman rank correlation coefficient.

5 Graph construction

For the purpose of evaluation of the random walk measures, we construct a graph for every pair of words for which similarity has to be computed. This graph is derived from WordNet as follows:

- For each word w in the pair (w_1, w_2) :
 - Add an edge between w and all of its parts of speech. For example, if the word is `coast`, add edges between `coast` and `coast#noun` and `coast#verb`.
 - For each word#pos combination, add edges to all of its senses (For example, `coast#noun#1` through `coast#noun#4`).
 - For each word sense, add edges to all of its hyponyms
 - For each word sense, add edges to all of its hypernyms recursively.

In this paper we consider uniform weights on all edges as our main aim is to illustrate the different random walk measures rather than fine tune the graph construction process.

6 Shortest path based measure

The most obvious measure of distance in a graph is the shortest path between the vertices which is defined as the minimum number of intervening edges between two vertices. This is also known as the geodesic distance. To convert this distance measure to a similarity measure, we take the reciprocal of the shortest-path length. We refer to this as the geodesic similarity. This is not a random walk

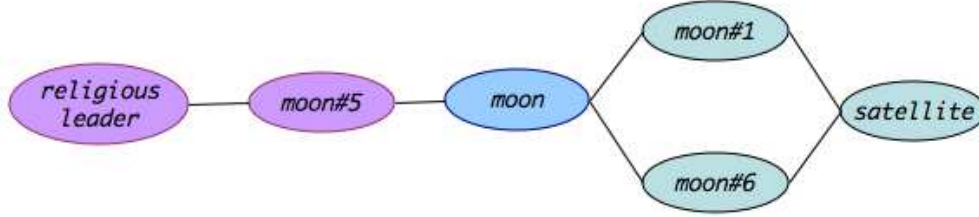


Figure 1: Shortest path distances on graphs

measure but will serve as an important baseline for our work. As can be observed from Table 1, the

Method	Spearman correlation
Geodesic	0.275

Table 1: Similarity using shortest-path measure.

correlation is rather poor for the shortest path measure.

7 Why are shortest path distances bad?

While shortest-path distances are useful in many applications, it fails to capture the following observation. Consider the subgraph of WordNet shown in Figure 1. The term `moon` is connected to the terms `religious_leader` and `satellite`¹. Observe that both `religious_leader` and `satellite` are at the same shortest path distance from `moon`. However, the connectivity structure of the graph would suggest `satellite` to be “more” similar than `religious_leader` as there are multiple senses, and hence multiple paths, connecting `satellite` and `moon`.

Thus it is desirable to have a measure that captures not only path lengths but also the connectivity structure of the graph. This notion is elegantly captured using random walks on graphs.

7.1 Similarity via Random walks

A random walk is a stochastic process that consists of a sequence of discrete steps taken at random defined by a distribution. Random walks have interesting connections to Brownian motion, heat diffusion and have been used in semi-supervised learning – for example, see (Zhu et al., 2003). Certain properties of random walks are defined for ergodic processes only². In our work, we assume these

¹The `religious_leader` sense of `moon` is due to Sun Myung Moon, a US religious leader.

²A stochastic process is ergodic if the underlying Markov chain is irreducible and aperiodic. A Markov chain is irre-

ducible if there exists a path between any two states and it is aperiodic if the GCD of all cycle lengths is one.

7.1.1 Bounded length walks

As our first random walk measure, we consider the bounded length walk – i.e., all random walks of length less than or equal to a bound m . We derive a probability transition matrix \mathbf{P} from the weight matrix \mathbf{W} as follows:

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$$

where, \mathbf{D} is a diagonal matrix with $d_{ii} = \sum_{j=1}^n w_{ij}$. Observe that:

- $p_{ij} = \mathbf{P}[i, j] \geq 0$, and
- $\sum_{j=1}^n p_{ij} = 1$

Hence p_{ij} can be interpreted as the probability of transition from vertex i to vertex j in one step. It is easy to observe that \mathbf{P}^k gives the transition probability from vertex i to vertex j in k steps. This leads to the following similarity measure:

$$\mathbf{S} = \mathbf{P} + \mathbf{P}^2 + \mathbf{P}^3 + \dots + \mathbf{P}^m$$

Observe that $\mathbf{S}[i, j]$ derives the total probability of transition from vertex i to vertex j in at most m steps³. Given \mathbf{S} , we can derive several measures of similarity:

1. Bounded Walk: $\mathbf{S}[i, j]$
2. Bounded Walk Cosine: dot product of rowvectors \mathbf{S}_i and \mathbf{S}_j .

When we evaluate these measures on the Miller-Charles data the results shown in Table 2. are observed. For this experiment, we consider all walks that are at most 20 steps long, i.e., $m = 20$. Observe that these results are significantly better than the Geodesic similarity based on shortest-paths.

³The matrix \mathbf{S} is row normalized to ensure that the entries can be interpreted as probabilities.

Method	Spearman correlation
Bounded Walk	0.346
Bounded Walk Cosine	0.365

Table 2: Similarity using bounded random walks ($m = 20$).

7.1.2 How many paths are sufficient?

In the previous experiment, we arbitrarily fixed $m = 20$. However, as observed in Figure 2., beyond a certain value the choice of m does not affect the result as the random walk converges to its stationary distribution. The choice of m depends on

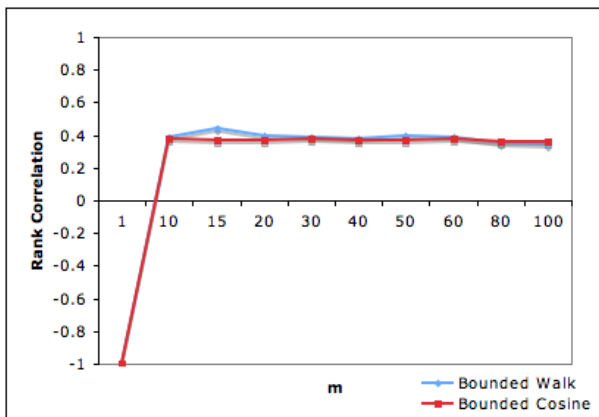


Figure 2: Effect of m in Bounded walk

the amount of computation available. A reasonably large value of m ($m > 10$) should be sufficient for most purposes and one could use lower values of m to derive an approximation for this measure. One could derive an upper bound on the value of m using the mixing time of the underlying Markov chain (Aldous and Fill, 2001).

7.1.3 Similarity via pagerank

Pagerank (Page et al., 1998) is the celebrated citation ranking algorithm that has been applied to several natural language problems from summarization (Erkan and Radev, 2004) to opinion mining (Esuli and Sebastiani, 2007) to our task of lexical relatedness (Hughes and Ramage, 2007). Pagerank is yet another random walk model with a difference that it allows the random walk to “jump” to its initial state with a nonzero probability (α). Given the probability transition matrix \mathbf{P} as defined above, a stationary distribution vector for any vertex (say i) could be derived as follows:

1. Let \mathbf{e}_i be a vector of all zeros with $\mathbf{e}_i(i) = 1$

2. Let $v_0 = \mathbf{e}_i$

3. Repeat until $\|v_t - v_{t-1}\|_F < \epsilon$

- $v_{t+1} = \alpha v_t \mathbf{P} + (1 - \alpha)v_0$
- $t = t + 1$

4. Assign v_{t+1} as the stationary distribution for vertex i .

Armed with the stationary distribution vectors for vertices i and j , we define pagerank similarity either as the cosine of the stationary distribution vectors or the reciprocal Jensen-Shannon (JS) divergence⁴ between them. Table 3. shows results on the Miller-Charles data. We use $\alpha = 0.1$, the best value on this data. Observe that these results are

Method	Spearman correlation
Pagerank JS-Divergence	0.379
Pagerank Cosine	0.393

Table 3: Similarity via pagerank ($\alpha = 0.1$).

better than the best bounded walk result. We further note that our results are different from that of (Hughes and Ramage, 2007) as they use extensive feature engineering and weight tuning during the graph generation process that we have not been able to reproduce. Hence for simplicity we stuck to a simpler graph generation process. Nevertheless, the result in Table 3. is still useful as we are interested in the performance of the various spectral similarity measures rather than achieving the best performance on the lexical relatedness task. The graphs we use in all methods are identical making comparisons across methods possible.

7.2 Similarity via Hitting Time

Given a graph with the transition probability matrix \mathbf{P} as defined above, the hitting time between vertices i and j , denoted as $h(i, j)$, is defined as the expected number of steps taken by a random walker to first encounter vertex j starting from vertex i . This can be recursively defined as follows:

$$h(i, j) = \begin{cases} 1 + \sum_{k: w_{ik} > 0} p_{ik} h(k, j) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \quad (1)$$

⁴The Jensen-Shannon divergence between two distributions p and q is defined as $D(p \parallel a) + D(q \parallel a)$, where $D(\cdot \parallel \cdot)$ is the Kullback-Liebler divergence and $a = (p + q)/2$. Note that unlike KL-divergence this measure is symmetric. See (Lin, 1991) for additional details.

The lower the hitting times of two vertices, the more similar they are. It can be easily verified that hitting time is not a symmetric relation hence graph theory literature suggests another symmetric measure – the commute time.⁵ The commute time, $c(i, j)$, is the expected number of steps taken to leave vertex i , reach vertex j , and return back to i . Thus,

$$c(i, j) = h(i, j) + h(j, i) \quad (2)$$

Observe that, the commute time is a metric in that it is positive definite, symmetric, and satisfies triangle inequality. Hence, commute time could be used as a distance measure as well. We derive a similarity measure from this distance measure using the following lemma.

Lemma 1. *For every edge (i, j) , $c(i, j) \leq 2l$ where $l = |E|$, the number of edges.*

Proof. This can be easily observed by defining a Markov chain on the edges with probability transition matrix \mathbf{Q} with $2l$ states, such that $Q_{e_1 e_2} = 1/\text{degree}(e_1 \cap e_2)$. Since this matrix is doubly stochastic, the stationary distribution on this chain will be uniform with a probability $1/2l$. Now $c(i, j) = h(i, j) + h(j, i)$, is the expected time for a walk to start at i , visit j , and return back to i . When the stationary probability at each edge is $1/2l$, this expected time evaluates to $2l$. Hence the commute time can be at most $2l$. \square

This lemma allows us to define a similarity measure as follows:

$$\text{sim}_C(i, j) = 1 - \frac{c(i, j)}{2l} \quad (3)$$

Observe that the measure defined in Equation 3 is a metric and further its range is defined in $[0, 1]$. We now only need a way to compute the commute times to use Equation 3. One could compute the hitting times and hence the commute times from the Equations 1 and 2 using dynamic programming, akin to shortest paths in graphs. In this paper, we instead choose to derive commute times via the graph Laplacian. This also allows us to handle “noise” in the graph construction process which cannot be taken care by naive dynamic programming.

⁵Note that distance measures, in general, need not be symmetric but we interpret distance as proximity which mandates symmetry.

Chandra et. al. (1989) show that the commute time between two vertices is equal to the resistance distance between them. Resistance distance, as proposed by Klein and Randic (1993), is the effective resistance between two vertices in the electrical network represented by the graph, where the edges have resistance $1/w_{ij}$. Xiao and Gutman (2003), show the relation between resistance distances in graphs to the Laplacian spectrum, thus enabling a way to derive commute times from the graph Laplacian in closed form.

We now introduce graph Laplacians, which are interesting in their own right besides being related to commute time. The Laplacian of a graph could be viewed as a discrete version of the Laplace-Beltrami operator on Riemannian manifolds. It is defined as

$$\mathcal{L} = \mathbf{D} - \mathbf{W}$$

The graph Laplacian has interesting properties and a wide range of applications, in semi-supervised learning (Zhu et al., 2003), non-linear dimensionality reduction (Roweis and Saul, 2000; Belkin and Niyogi, 2001), and so on. See (Chung, 1997) for a thorough introduction on Laplacians and their properties. We depend on the fact that \mathcal{L} is:

1. symmetric (since D and W are for undirected graphs)
2. positive-semidefinite : since it is symmetric, all of the eigenvalues are real and by the Greshgorin circle theorem, the eigenvalues must also be non-negative and hence \mathcal{L} is positive-semidefinite.

Throughout this paper we use normalized Laplacians as defined below:

$$\mathbf{L} = \mathbf{D}^{-1/2} \mathcal{L} \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$$

The normalized Laplacians preserve all properties of the Laplacian by construction.

As noted in Xiao and Gutman (2003), the resistance distances can be derived from the generalized Moore-Penrose pseudo-inverse of the graph Laplacian (\mathbf{L}^\dagger) – also called the inverse Laplacian. Like Laplacians, their pseudo inverse counterparts are also symmetric, and positive semi-definite.

Lemma 2. \mathbf{L}^\dagger is symmetric

Proof. The Moore-Penrose pseudo-inverse is defined as $\mathbf{L}^\dagger = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T$. From this definition, it is clear that $(\mathbf{L}^\dagger)^T = (\mathbf{L}^T)^\dagger$. By the symmetry

property of graph Laplacians, $\mathbf{L}^T = \mathbf{L}$. Hence, $(\mathbf{L}^\dagger)^T = \mathbf{L}^\dagger$. \square

Lemma 3. \mathbf{L}^\dagger is positive semi-definite

Proof. We make use of the following properties from (Chung, 1997):

- The Laplacian, \mathbf{L} , is positive semi-definite (also shown above).
- If the Eigen-decomposition of \mathbf{L} is $\mathbf{Q}\Lambda\mathbf{Q}^T$, then the Eigen-decomposition of the pseudo-inverse \mathbf{L}^\dagger is $\mathbf{Q}\Lambda^{-1}\mathbf{Q}^T$. If any of the eigenvalues of \mathbf{L} is zero then the corresponding eigenvalue for \mathbf{L}^\dagger is also zero.

Since \mathbf{L} is positive semi-definite, and the eigenvalues of \mathbf{L}^\dagger have the same sign as \mathbf{L} , the pseudo inverse \mathbf{L}^\dagger has to be positive semi-definite. \square

Lemma 4. The inverse Laplacian is a gram matrix

Proof. To prove this, we use the fact that the Laplacian Matrix is symmetric and positive semi-definite. Hence by Cholesky decomposition we can write $\mathbf{L} = \mathbf{U}\mathbf{U}^T$.

Therefore $\mathbf{L}^\dagger = (\mathbf{U}^T)^\dagger\mathbf{U}^\dagger = (\mathbf{U}^\dagger)^T(\mathbf{U}^\dagger)$.

Hence \mathbf{L}^\dagger is a matrix of dot-products or a gram-matrix. \square

Thus, from Lemmas 2, 3 and 4, the inverse Laplacian \mathbf{L}^\dagger is a valid Kernel.

7.2.1 Similarity measures from the Laplacian

The pseudo inverse of the Laplacian allows us to compute the following similarity measures.

1. Since \mathbf{L}^\dagger is a kernel, \mathbf{L}_{ij}^\dagger can be interpreted a similarity value of vertices i and j .
2. **Commute time:** This is due to (Aldous and Fill, 2001). The commute time, $c(i, j) \propto (\mathbf{L}_{ii}^\dagger + \mathbf{L}_{jj}^\dagger - 2\mathbf{L}_{ij}^\dagger)$. This allows us to derive similarities using Equation 3.

Evaluating the above measures with the Miller-Charles data yields results shown in Table 4. Again, these results are better than the other random walk methods compared in the paper.

Method	Spearman correlation
\mathbf{L}_{ij}^\dagger	0.469
Commute Time (sim_C)	0.520

Table 4: Similarity via inverse Laplacian.

7.2.2 Noise in the graph construction process

The graph construction process outlined in Section 5 is not necessarily the best one. In fact, any method that constructs graphs from existing data incorporates “noise” or extraneous features. These could be spurious edges between vertices, missing edges, or even improper edge weights. It is however impossible to know any of this a priori and some noise is inevitable. The derivation of commute times via the pseudo inverse of a noisy Laplacian matrix makes it even worse because the pseudo inverse amplifies the noise in the original matrix. This is because the largest singular value of the pseudo inverse of a matrix is equal to the inverse of the *smallest* singular value of the original matrix. A standard technique in signal processing and information retrieval to eliminate noise or handle missing values is to use singular value decomposition (Deerwester et al., 1990). We apply SVD to handle noise in the graph construction process.

For a given matrix A , SVD decomposes A into three matrices U , S , and V such that $A = USV^T$, where S is a diagonal matrix of eigenvalues of A , and U and V are orthonormal matrices containing the left and the right eigenvectors respectively. The top- k eigenvectors and eigenvalues are computed using the iterative method by Lanczos-Arnoldi (using LAPACK) and the product of these matrices represents a “smoothed” version of the original Laplacian. The pseudo inverse is then computed on this smooth Laplacian. Table 5., shows the improvements obtained by discarding bottom 20% of the eigenvalues.

Method	Original	After SVD
\mathbf{L}_{ij}^\dagger	0.469	0.472
Commute Time (sim_C)	0.520	0.542

Table 5: Denoising graph Laplacian via SVD

Figure 3. shows the dependence on the number of eigenvalues selected. As can be observed in both curves there is a reduction in performance by adding the last few eigenvectors and hence may be safely discarded. This observation is true in other text processing tasks like document clustering or classification using Latent Semantic Indexing.

8 Related Work

Apart from the related work cited throughout this paper, we would also like to note the paper by Yen

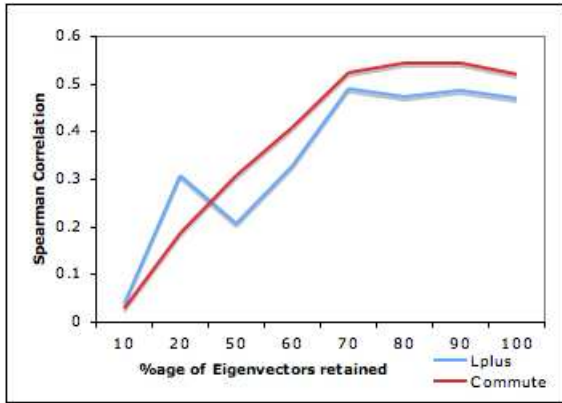


Figure 3: Noise reduction via SVD.

et al (2007) on using sigmoid commute time kernel on a graph for document clustering but our work differs in that our goal was to study various random walk measures rather than a specific task and we provide a new similarity measure (ref. Eqn 3) based on an upper bound on the commute time (Lemma 1). Our work also suggests a way to handle noise in the graph construction process.

9 Conclusions and Future Work

This paper presented an evaluation of random walk based similarity measures on weighted undirected graphs. We provided an intuitive explanation of why random walk based measures perform better than shortest-path or geodesic measures, and backed it with empirical evidence. The random walk measures we consider include bounded length walks, pagerank based measures, and a new measure based on the commute times in graphs. We derived the commute times via pseudo inverse of the graph Laplacian. This enables a new method of graph similarity using SVD that is robust to the noise in the graph construction process. Further, the inverse Laplacian is also interesting in that it is a kernel by itself and could be used for other tasks like word clustering, for example.

Acknowledgements

The authors would like to thank David Smith and Petros Drineas for useful discussions and to Fan Chung for the wonderful book on Spectral Graph theory.

References

Aldous and Fill. 2001. *Reversible Markov Chains and Random Walks on Graphs*. In preparation.

Belkin, Mikhail and Partha Niyogi. 2001. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Proceedings of the NIPS*.

Blatz, John, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanichis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceeding of the COLING*.

Budanitsky, Alexander and Graeme Hirst. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

Chandra, Ashok, Prabhakar Raghavan, Walter Ruzzo, Roman Smolensky, and Prason Tiwari. 1989. The electrical resistance of a graph captures its commute and cover times. In *Proceedings of the STOC*.

Chung, Fan. 1997. Spectral graph theory. In *CBMS: Conference Board of the Mathematical Sciences, Regional Conference Series*.

Deerwester, Scott, Susan Dumais, George Furnas, Thomas Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41.

Erkan, Günes and Dragomir Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*, 22:457–479.

Esuli, Andrea and Fabrizio Sebastiani. 2007. Pageranking wordnet synsets: An application to opinion mining. In *Proceedings of the ACL*, pages 424–431.

Fellbaum, Christaine, editor. 1989. *WordNet: An Electronic Lexical Database*. The MIT Press.

Hughes, Thad and Daniel Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of the EMNLP*.

Kauchak, David and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings HLT-NAACL*.

Klein, D. and M. Randic. 1993. Resistance distance. *Journal of Mathematical Chemistry*, 12:81–95.

Lin, Jianhua. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1).

Miller, G. and W. Charles. 1991. Contextual correlates of semantic similarity. In *Language and Cognitive Process*.

Page, Larry, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. The pagerank citation ranking: Bringing order to the web. *Technical report, Stanford University, Stanford, CA*.

- Patwardhan, Siddharth, Satanjeev Banerjee, and Ted Pedersen. 2005. Senserelate:: Targetword-A generalized framework for word sense disambiguation. In *Proceedings of the ACL*.
- Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity - measuring the relatedness of concepts. In *Proceedings of the AAAI*.
- Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of english words. In *Proceedings of the ACL*.
- Prager, John M., Jennifer Chu-Carroll, and Krzysztof Czuba. 2001. Use of wordnet hypernyms for answering what-is questions. In *Proceedings of the Text REtrieval Conference*.
- Roweis, Sam and Lawrence Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326.
- Xiao, W. and I. Gutman. 2003. Resistance distance and laplacian spectrum. *Theoretical Chemistry Association*, 110:284–289.
- Yen, Luh, Francois Fouss, Christine Decaestecker, Pascal Francq, and Marco Saerens. 2007. Graph nodes clustering based on the commute-time kernel. In *Proceedings of the PAKDD*.
- Zhu, Xiaojin, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the ICML*.