Causation Generalization Through the Identification of Equivalent Nodes in Causal Sparse Graphs Constructed from Text using Node Similarity Strategies

Brett Drury **Jorge Valverde-Rebaza** **Alneu de Andrade Lopes**
Department of Computer Science
ICMC, University of São Paulo
C.P. 668, CEP 13560-970, São Carlos
{bdrury, jvalverr, alneu}@icmc.usp.br

Abstract

Causal Bayesian Graphs can be constructed from causal information in text. These graphs can be sparse because the cause or effect event can be expressed in various ways to represent the same information. This sparseness can corrupt inferences made on the graph. This paper proposes to reduce sparseness by merging: equivalent nodes and their edges. This paper presents a number of experiments that evaluates the applicability of node similarity techniques to detect equivalent nodes. The experiments found that techniques that rely upon combination of node contents and structural information are the most accurate strategies, specifically we have employed: 1. node name similarity and 2. combination of node name similarity and common neighbours (SMCN). In addition, the SMCN returns "better" equivalent nodes than the string matching strategy.

1 Introduction

Graphs can be constructed to represent a specific domain from which inferences can be made about a future event(s) based on observations (Newman, 2010). These graphs tend to be constructed: 1. manually from information elicited from experts in the field or 2. from other information sources (Horny, 2014). A manual construction process can be slow, and represent a partial slice of the domain. An alternative approach is to construct a domain specific graph from information in text. The advantage of this approach is that graphs can be constructed automatically, and therefore the construction process can be quick and the graph domain coverage can be more comprehensive than a graph constructed manually (Hensman, 2004; Jin and Srihari, 2007).

1.1 Node Merge Problem

A major disadvantage of constructing graphs from text is that the same assertions can be stated in various different ways. These variations may be in the words chosen and their order (Jin and Srihari, 2007). The consequence of varying language is that the graph generated from it can have many nodes, that have one edge, consequently accurate inference may be difficult due to the sparse structure of the graph (Tsang and Stevenson, 2010). An approach to minimize this characteristic of text built graphs is to merge similar nodes and their edges. This will improve the graph by: 1. decreasing the number of nodes, 2. increasing the average number of edges per node and 3. inferring new causes or effects for events which are not explicitly stated in the text the graph is constructed from. For example, "*. . . começa a reduzir preço do etanol*" and "*Preço do etanol começa a diminuir*" represent the same concept, but are written in a different order. In a graph constructed from text these two events would be two different nodes, but arguably these nodes should be merged because they represent the same event.

The merged node process is demonstrated in Figures 1 and 2. The figures demonstrate two candidates nodes for merging *B* and *B#*. The two candidates have very similar node names as well as common neighbours *C* and *A*. The merge process joins the two nodes into one node *B[B#]* which combines the neighbours of the previous two graphs. In a causal Bayesian Network where in-links are causes and out-links are effects, the proposed merge process would infer new causes and effects which are not explicitly stated in the construction text (Girju, 2003; Shpitser and Pearl, 2008).

1.2 Node Similarity

The proposal presented by this work is that identical nodes can be identified by Node Similarity measures and these nodes are candidates for merging. It should be noted that the aim of this work is not to present general similarity measures, but to identify strategies which can accurately identify nodes that represent the same event.

This paper will present a series of experiments that evaluate a number of common node similarity measures as well as a number of novel variations of these techniques. This paper will conform to the following format: Related Work, Proposed Techniques, Evaluation and Future Work.

2 Related Work

The related work covers two main areas: causal graphs constructed from text and node similarity measure.

2.1 Causal Directed Graphs from Text

Causal directed graphs, are graphical models that represent the inference process between two variables: X and Y, through the use of two nodes and a directed link from: X to Y, whenever Y responds to changes in X when all other variables are being held constant (Shpitser and Pearl, 2008).

A common problem in this domain is the manner of the construction of the Bayesian Graph. Manual construction can be a labour intensive process that may not provide good coverage for a specific domain. An alternative is to construct graphs from information in text.

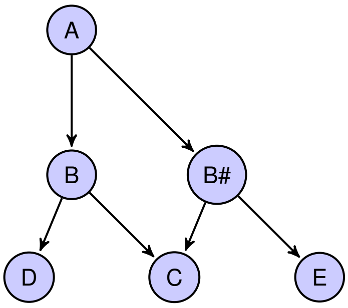

Figure 1: Candidate for Node Merging.

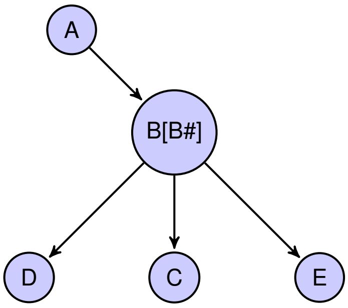

Figure 2: Graph after candidate node merged.

A number of attempts to construct Causal Bayesian Networks from text have been documented. An early attempt at constructing a Bayesian Graph from text was proposed by (Sanchez-Graillet and Poesio, 2004). They constructed a Causal Bayesian Graph from causal relations in text. They generalize about causal relations by identifying synonyms in similar event phrases. The synonyms are identified using external lexical resources that they admit did not provide full coverage. (Bojduj, 2009) used decision rules to extract causal relations to construct a Bayesian graph. It was not clear how causality was generalized and if the constructed graph was sparse. (Raghuram et al., 2011) produced a prototype called Auto-Bayesian that constructed Bayesian Graphss from causal relations in text. Finally, (Miranda Ackerman, 2012) produced a causal Bayesian Networks from causal topics in text. The topic approach provided a partial generalization about causation in the network.

2.2 Node Similarity Measures

The notion of similarity is documented in many domains, consequently similarity can be measured in a variety of ways. The notion of similarity is dependent upon the domain and the appropriate definition of similarity for that domain (Jeh and Widom, 2002).

In graphs, similarity between a pair of nodes indicates that these nodes share a common relation, consequently, similarity measures can be used to: 1. predict new relationships (Valverde-Rebaza and Lopes, 2013; Lü and Zhou, 2011), 2. detect communities (Valejo et al., 2014), 3. node classification (Valverde-Rebaza et al., 2014), and 4. improve the graph construction (Berton et al., 2015).

In graphs, where similarity among nodes is based solely on graph structure, similarity is referred to as structural similarity. Structural similarity measures can be grouped into measures that rely upon: 1. local or 2 .global information.

Global measures can obtain higher accuracy measures than local measures, but they are computational complex, and consequently are unfeasible for large-scale graphs. Local measures are generally faster, but obtain lower accuracy than global measures. Examples of common local measures are: 1. Common Neighbours, 2. Jaccard coefficient, 3. Adamic Adar, 4. Resource Allocation and 5. Preferential Attachment measures (Lü and Zhou, 2011; Valverde-Rebaza and Lopes, 2013). Standard global measures are: 1. SimRank, 3. Katz, and 3. Rooted PageRank (Lü and Zhou, 2011; Valverde-Rebaza and Lopes, 2013).

The following describes two common similarity measures: Common Neighbours and SimRank using a node pair: v_i and v_j that is assigned a score s_{v_i,v_j}. $\Gamma(v_i)$ denotes a set of neighbours of v_i.

The **Common Neighbours (CN)** technique assumes that v_i and v_j are similar if they share neighbours, therefore CN refers to the size of the set of all common neighbours of both v_i and v_j according to Eq. 1.

$$s^{CN}_{v_i,v_j} = |\Gamma(v_i) \cap \Gamma(v_j)| \tag{1}$$

The **SimRank (SR)** technique assumes two nodes are similar if they are joined to similar neighbours. The SimRank measure is defined as Eq. 2.

$$s^{SR}_{v_i,v_j} = \gamma \cdot \frac{\sum_{v_k \in \Gamma(v_i)} \sum_{v_m \in \Gamma(v_j)} s^{SR}_{v_k,v_m}}{|\Gamma(v_i)| \cdot |\Gamma(v_j)|} \tag{2}$$

Where the parameter $\gamma \in [0, 1]$ is the decay factor. Due to SimRank can also be interpreted in terms of a random walk process, that is, the expected value of $s^{SR}_{v_i,v_j}$ measures how soon two random walkers, respectively starting from nodes v_i and v_j, are expected to meet at a certain node.

3 Proposed Techniques

The aim of the proposed techniques is to generalize causal relationships in a causal graph constructed from text without recourse to lexical resources as per (Sanchez-Graillet and Poesio, 2004) by identifying equivalent nodes and merging them. We call this the Node Merge Problem [1].

This paper evaluates a number of node similarity techniques for their ability to identify merge candidates (nodes which have different names, but represent the same event). The base techniques are three common strategies: SimRank, Common Neighbours and Node Name Similarity (String matching) (Robles-Kelly and Hancock, 2004). These techniques are commonly used to identify similar nodes in pre-processing step in link prediction strategies (Lü and Zhou, 2011; Valverde-Rebaza and Lopes, 2013). The strategies which were developed for this paper were: Fuzzy SimRank and String matching with common neighbours.

[1] The node merge problem was explained on page 1

3.1 Fuzzy SimRank

Fuzzy SimRank is an adaptation of SimRank. SimRank is a recursive algorithm which relies upon the structural similarity of nodes. In sparsely connected graphs the structure is poor because very few nodes are connected, and consequently SimRank can not make accurate comparisons between nodes (Jeh and Widom, 2002). Fuzzy SimRank assumes an implied structure through partial edge similarity. The SimRank algorithm computes similarity by making a direct comparison of neighbours of given nodes. A match is only recorded when the nodes are exactly the same. Graphs created from text may have many similar nodes which when compared will be scored the same as nodes that are not related. Fuzzy SimRank applied a value between 0 to 1 based upon the similarity of the node names, i.e. a score of 1 indicates that the node names are equal, and a score of 0 indicates that the nodes names have no common text. The values computed for the similarity between nodes are computed with common string matching algorithms. The string matching algorithms used in the Fuzzy SimRank algorithms for this paper were: Longest Common Sub-sequence, Levenstein Distance and Sorensen Distance (Rahm and Bernstein, 2001).

3.2 String matching with common neighbours

String matching with common neighbours (SMCN) is a technique that computes a similarity between two nodes using: node name similarity and common neighbours. The common neighbours measure was altered to compute two similarity measures: an in-link and out-link similarity because in-links and out-links represent cause and effect respectively, consequently in-links and out-links for equivalent nodes can not be the same. The SMCN is represented by:

$$< Sim(N_1O, N_2O) + Sim(N_1I, N_2I) + \\ Sim(N_1N, N_2N) > \tag{3}$$

where, N_1O is the out-links of Node 1 in a two Node comparison pair, N_2O is the out-links of Node 2 in a two Node comparison pair, N_1I is the in-links of Node 1 in a two Node comparison pair, N_2I is the in-links of Node 2 in a two Node comparison pair, N_1N is the Node name of Node 1 in a two Node comparison pair and N_2N is the Node name of Node 2 in a two Node comparison pair.

There were three versions of SMCN which varied the similarity measure (Sim) used for the nodes comparison. The first similarity measure was a Jacard distance, that relied upon exact matching of nodes to compute a similarity between neighbours of two nodes. The remaining variations computed similarity between neighbours by using a Longest Common Subsequence similarity measure (LCSM). The LCSM measure is approximated by comparing the node names of all the neighbours of one of the candidate node against all of the all the neighbours of other of the candidate node. An average is taken of all of the similarity scores. This measure is demonstrated in Algorithm 1. The algorithm iterates through all of the nodes and compares each node with all of the nodes in the graph. The node pairs that have a similarity above a pre-determined threshold are marked as candidates for merging. It should be noted that a node can be marked as a merge candidate for more than one node, consequently a merged node will represent at least 2 nodes and a maximum of $n - 1$ nodes where n represents the number of nodes in the graph.

```
Input: N1,N2, threshold
Output: Sim
/* N1 = Neighbours of Node 1, N2 =
   Neighbours of Node 2          */
/* Sim = similarity, threshold =
   lower bound similarity score  */
localSim = ()
for node1 in N1 do
    for node2 in N2 do
        sim = simularity(node1,node2)
        if sim > threshold then
            localSim.push(sim)
        end
    end
end
return (mean(localSim))
```

Algorithm 1: Fuzzy Node Matching

The variations of the fuzzy node matching used differing threshold values, these values were a range of $>= 0.0 <= 1.0$, where 1.0 is a perfect match. The threshold values used for the variations were: 0.9 and 0.0. These values were chosen to see if that 'neighbour near misses' (0.9) produced better results than measuring the similarity of all neighbours.

4 Evaluation

The evaluation was intended to demonstrate the ability of the proposed candidate techniques to identify equivalent nodes in a graph. These nodes would be candidates for merging. The candidate techniques were evaluated on a graph created from Brazilian - Portuguese news stories [2] . The graph was created from causal relations extracted from the Brazilian - Portuguese news corpus. The relations were extracted using Levin's causative pattern: $NP\ V\ NP$, where NP is a noun phrase and V is a causal verb (Levin, 1993). The verb used in these experiments was the verb " causar ". This verb was chosen because: 1. it is a simple causative verb and consequently it will not form part of the cause or effect and 2. it is unambiguous. Levin's pattern assumes that: the first NP is the cause and the second is the effect. The position of cause and effect NP can be reversed. The reversing of the cause and effect NPs in these experiments was based upon lexical indicators such as "por" or "de". An example of this phenomenon is demonstrated in the phrase: " falta de chuva por causar de seca ", the NP 'falta de chuva' is the effect rather than the cause because of the preposition "de" .

The graph was created by transforming the NPs into nodes. The nodes were connected using the causal verbs. For example, the phrase " falta de chuva por causar de seca " would be transformed in into the structure shown in Figure 3.

The final graph contained 4045 nodes and 2180 edges. It was expected that this graph would contain duplicate nodes because the corpus it was constructed from contained repeating themes over a long period of time.

The typical node similarity evaluation strategies such as *Top K holdout* were not appropriate for this problem because edges in this graph do not indicate similarity, but cause or effect. This was

[2] Graph available from `https://goo.gl/IPe8qB` in pickled NetworkX Digraph Format

confirmed in a brief experiment where all candidate similarity strategies failed to identify the missing neighbours. We therefore used a manual evaluation strategy. The evaluation were conducted by a single annotator. The three evaluations were: precision for top 'n' similarities for 'n' randomly selected nodes, precision for most statistically significant similarities and precision by similarity score.

4.1 Precision for 'n' similarities for 'n' randomly evaluation

This evaluation is adapted from the information retrieval literature (Manning et al., 2008). Thus, randomly are selected 10 nodes from the graph and ranked the most similar nodes by descending accuracy score from 1 to n. The evaluation verified whether two nodes represented equivalent events. Thus, was evaluated: a. 5 most similar nodes, b. 10 most similar nodes and c. 20 most similar nodes. An average of the results for all nodes was then calculated. The results are in Figure 4. Strategies which returned no documents or a score of 0 for intervals are excluded from the diagram for clarity. The results demonstrate that rank is not a good indicator for node equivalence as all strategies performed poorly. The SimRank variations scored 0 accuracy or did not return any results for all of the selected nodes. The local similarity measures fared little better. Although the evaluation was limited it is an indication that rank provides little information when identifying equivalent nodes.

4.2 Precision for most statistically significant similarities evaluation

In this subsection, we evaluate if statistical significance was an indicator of node equivalence. Statistical significance in this case was the number of standard deviations between an accuracy for a node pair and average accuracy for all node pairs.

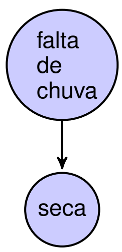

Figure 3: Sample Structure.

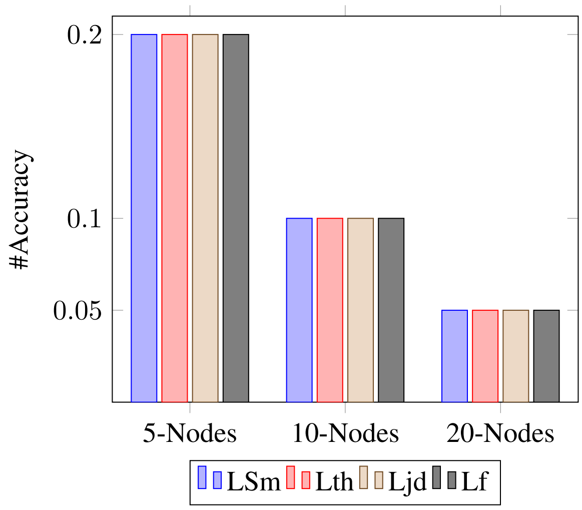

Figure 4: Accuracy for Randomly Selected Nodes, *where Lsm = Local similarity with string matching, Lth = Local similarity with string matching and common nodes with a minimum similarity threshold of 0.9, Ljd = Local similarity with string matching and common nodes with a minimum similarity threshold of 1.0 and Lf = Local similarity with string matching and common nodes with a minimum similarity threshold of >0.0*

This evaluation computed node similarities for every possible combination of nodes in the graph. The candidate node pair similarities were ranked by node (as per previous evaluation). A standard deviation is computed from the non zero node similarities. The number of standard deviations is computed between: 1. the most similar node pair for a given node and 2. the second most similar node pair. All the node pairs are then ranked by the number of standard deviations. Thus, was evaluated the: a. 5 most statistically significant, b. 10 most statistically significant and c. 20 most statistically significant, similar candidate pairs. The results are in Figure 5. Techniques which scored 0 for all of the sample intervals were excluded from the diagram for clarity.

The techniques provided improved candidate pair equivalences. The SimRank variations which used Levenstein or Common Longest Sequence generated better node equivalence pairs than the basic SimRank. However, do not was observed a statistically significant among node pairs. The best results were gained by the string matching (Lsm) approach. The approach returned very similar node pairs where the difference between the node names were minor differences in words. An example is provided in Table 1. For example in the first example the only difference between the pairs is the word **nesta**.

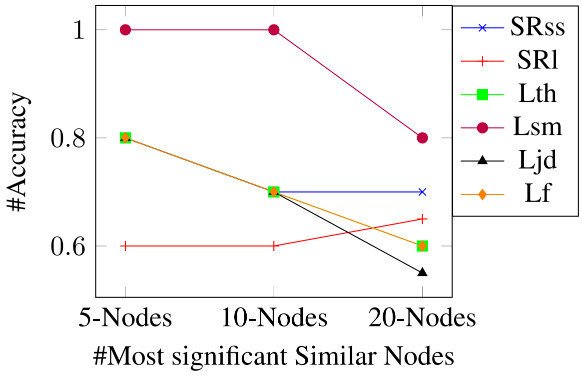

Figure 5: Accuracy for the statistically significant most similar nodes, *where, SRss = SimRank with Common Subsequence, SRl = SimRank with Levenstein Distance, Lsm = Local similarity with string matching, Lth = Local similarity with string matching and common nodes with a minimum similarity threshold of 0.9, Ljd = Local similarity with string matching and common nodes with a minimum similarity threshold of 1.0 and Lf = Local similarity with string matching and common nodes with a minimum similarity threshold of >0.0*

Node 1 Node Name	Node 2 Node Name
reconheceu **nesta** terça-feira pode faltar gasolina alguns postos	reconheceu terça-feira pode faltar gasolina alguns postos
traders importaram cerca toneladas produto desde outubro	**operadores** importaram cerca toneladas produto desde outubro
acusações envolvimento mensalão **esquema financiamento ilegal suposta compra deputados pelo**	acusações envolvimento mensalão

Table 1: Equivalent Node Pairs Examples

4.3 Accuracy by similarity score evaluation

The goal here is evaluate if the node similarity score was an indicator of node equivalence. The evaluation computed a similarity score for each node candidate pair. The evaluation created a range of 0.5 <= 1.0 in steps of 0.1, i.e there were 5 sub-ranges in the overall range. The lower bound of the sub-range acts as minimum similarity and the upper bound acts a maximum similarity. For each of these sub-ranges candidate pairs were randomly chosen and evaluated for node equivalence. The results are demonstrated in Figure 6. Techniques that scored 0 for all intervals are not included. The results show that the SimRank variants perform poorly. The string matching (Lsm) did improve accuracy with very high similarities. At these high similarities the differences between node names was very small.

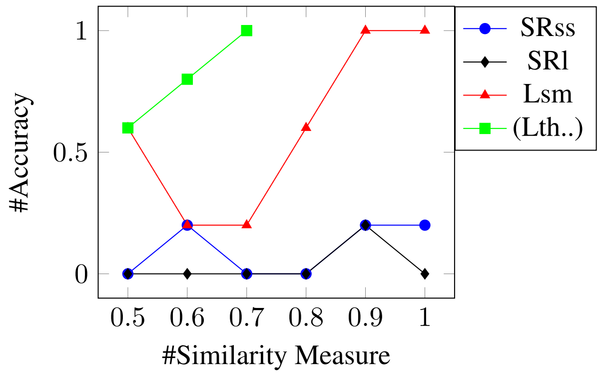

Figure 6: Accuracy by confidence intervals, where *SRss = SimRank with Common Subsequence, SRl = SimRank with Levenstein Distance, Lsm = Local similarity with string matching and Lth.. = L similarity with string matching and common nodes (techniques: Lth,Ljd, and Lf).*

The techniques that combined string matching with common neighbours performed well, gaining the best results at similarity level 0.7 after which no candidates pairs were returned. In contrast with the string matching results in the previous evaluation, the SMCN techniques returned "less" similar node names, but the events were equivalent. The common neighbours reinforced the notion of equivalence identified through string similarity. A comparison of high similarity examples from the string matching (SM) and SMCN techniques is shown in Table 2. It is quite clear from the comparison that the high similarity from the string matching returns node names where the differences are due to extraneous information, i.e the removal of the differences did not alter the meaning of the sentences. The SMCN differences were equivalents where removing the differences would change the meaning of the sentence.

5 Conclusion

The results demonstrate that local measures return the best results when compared to the various global (SimRank) techniques. In particular, the local measures that used: 1. node name similarity and 2. node name similarity with common neighbours (SMCN) produced the best results. It is arguable that the SMCN technique gained " better results " than the node name similarity technique. The node name similarity returned nodes that had similar node names that were differentiated by:

Node 1 Node Name	Node 2 Node Name	Technique
infecção pode destruir rapidamente tecido **causar** danos irreversíveis	infecção pode destruir rapidamente tecido **provocar** danos irreversíveis	SMCN
radiografia pulmões jornalista mostrou inflamações características doença	radiografia pulmões jornalista mostrou **infiltrações** inflamações características doença	SMCN
edmundo volta após sofrer **várias** punições disciplinares	edmundo volta após sofrer **diversas** punições disciplinare	SMCN
caso consigam manter vendas elevadas exterior	caso consigam manter vendas elevadas exterior **por**	SM
depredações piquetes durante greve geral	depredações piquetes durante greve geral **ontem**	SM
reconheceu terça-feira pode faltar gasolina alguns postos	reconheceu **nesta** terça-feira pode faltar gasolina alguns postos	SM

Table 2: Equivalent Node Pairs Examples (Confidence)

additional characters or words whereas the SMCN technique returned nodes that had lower node name similarity, but conveyed the same meaning. In addition the SMCN technique has the potential to be used in an iterative process because increasing the number edges may identify additional equivalent nodes. Furthermore, the SMCN technique avoids a common mistake made by the node name similarity technique, where two node names have a high superficial similarity, but convey the opposite meaning, for example 'momento oportuno' and 'momento inoportuno'. The SMCN similarity score would be low because these two nodes would have different edges. The use of partial node name (fuzzy) matching in the global and local measures did not improve the accuracy of the technique.

In general Node Similarity measures seem to be a viable strategy for identifying equivalent nodes in a "node merge" causation generalization strategy.

5.1 Future Work

The limitations of manual evaluations is that the amount of data that can be evaluated is restricted and the interpretation of results can be subjective, and open to errors. Consequently, the next step is to construct a larger graph and adapt one of traditional neighbour prediction evaluations, although at this stage it is not clear which one. In addition at the most accurate setting the *SMCN* strategy reduced the node count by *1%*, therefore we will be required to find settings that increase the number of nodes merged without sacrificing accuracy.

This work, we believe, has great potential in the generalization of causal statements in text and graph construction because it allows the inference of new causes and effects that are not stated explicitly in the construction text.

Acknowledgments

This work was partially supported by the São Paulo Research Foundation (FAPESP) grants: 2013/12191-5, 2011/22749-8 and 2011/20451-1.

References

L. Berton, J. Valverde-Rebaza, and A. Lopes. 2015. Link prediction in graph construction for supervised and semi-supervised learning. In *Proceedings of The 2015 International Joint Conference on Neural Networks*, IJCNN 2015, pages 1818–1825. IEEE.

B. N. Bojduj. 2009. Extraction of causal-association networks from unstructured text data. Master's thesis, California Polytechnic State University.

R. Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering - Volume 12*, MultiSumQA '03, pages 76–83. Association for Computational Linguistics.

S. Hensman. 2004. Construction of conceptual graph representation of texts. In *Proceedings of the Student Research Workshop at HLT-NAACL 2004*, HLT-SRWS '04, pages 49–54. Association for Computational Linguistics.

M. Horny. 2014. Bayesian statistics. Technical report, Boston UNiversity.

G. Jeh and J. Widom. 2002. Simrank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 538–543. ACM.

W. Jin and R. K. Srihari. 2007. Graph-based text representation and knowledge discovery. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, pages 807–811. ACM.

B. Levin. 1993. *English verb classes and alternations : a preliminary investigation*.

L. Lü and T. Zhou. 2011. Link prediction in complex networks: A survey. *Physica A*, 390(6):1150 – 1170.

C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

E. J. Miranda Ackerman. 2012. Extracting a causal network of news topics. In Pilar Herrero, Hervé Panetto, Robert Meersman, and Tharam Dillon, editors, *On the Move to Meaningful Internet Systems: OTM 2012 Workshops*, volume 7567 of *Lecture Notes in Computer Science*, pages 33–42.

M. E. J. Newman. 2010. *Networks: an introduction*. Oxford University Press.

S. Raghuram, Y. Xia, J. Ge, M. Palakal, J. Jones, D. Pecenka, E. Tinsley, J. Bandos, and J. Geesaman. 2011. Autobayesian: developing bayesian networks based on text mining. In *Database Systems for Advanced Applications*, pages 450–453. Springer.

E. Rahm and P. A. Bernstein. 2001. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4):334–350.

A. Robles-Kelly and E. R. Hancock. 2004. String edit distance, random walks and graph matching. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):315–327.

O. Sanchez-Graillet and M. Poesio. 2004. Acquiring bayesian networks from text. In *LREC*. European Language Resources Association.

I. Shpitser and J. Pearl. 2008. Complete identification methods for the causal hierarchy. *J. Mach. Learn. Res.*, 9:1941–1979.

V. Tsang and S. Stevenson. 2010. A graph-theoretic framework for semantic distance. *Comput. Linguist.*, 36(1):31–69.

A. Valejo, J. Valverde-Rebaza, B. Drury, and A. Lopes. 2014. Multilevel refinement based on neighborhood similarity. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, IDEAS'14, pages 67–76. ACM.

J. Valverde-Rebaza and A. Lopes. 2013. Exploiting behaviors of communities of Twitter users for link prediction. *SNAM*, 3(4):1063–1074.

J. Valverde-Rebaza, A. Soriano, L. Berton, M. C. F. de Oliveira, and A. Lopes. 2014. Music genre classification using traditional and relational approaches. In *Proceedings of 2014 Brazillian Conference on Intelligent Systems*, BRACIS 2014, pages 259–264. IEEE.