ECML PKDD 2009
European Conference on Machine Learning and Principles and
Practice of Knowledge Discovery in Databases

Workshop on Explorative Analytics of Information Networks

at ECML PKDD 2009

EIN2009

September 11, 2009
Bled, Slovenia

Editors:

Andreas Nüernberger
Otto-von-Guericke-University-Magdeburg, Germany

Michael R. Berthold
Tobias Kötter
Kilian Thiel
University of Konstanz, Germany

Program Committee

Christian Borgelt
Ulrik Brandes
Werner Dubitzky
Andreas Hotho
Nada Lavrac
Trevor Martin
Igor Mozetic
Luc de Raedt
Hannu Toivonen

Foreword

This volume contains the papers presented at the 1st workshop on Explorative Analytics of Information Networks held on Friday, September 11th, 2009 as part of the workshop program of PKDD ECML 2009 in Bled, Slovenia. The goals of the workshop are to intensify the exchange of ideas between different research communities to enable the design of tools for creation, analysis and visualization of complex information networks. The workshop focuses especially on researchers that are working on methods for representation of complex knowledge resources, (dynamic) data analysis methods, semantic networks, and visualization methods as well as user interface design.

The final program of eleven presentations covers a nice variety of in-depth papers on different aspects of information network creation, abstraction, and analysis complemented by two review papers covering the state of the art of information networks and network abstraction methods. The program shows nicely how this research area receives increasing attention in the Machine Learning and Data Mining community.

Magdeburg and Konstanz, August 2009
Andreas Nürnberger, Michael R. Berthold, Tobias Kötter and Kilian Thiel

Table of Contents

Creation

Analysis

Exploration

Information Networks: State of the Art

Tobias Kötter and Michael R. Berthold

Nycomed-Chair for Bioinformatics and Information Mining, University of Konstanz,
Fach M712, 78484 Konstanz, Germany
Tobias.Koetter@uni-Konstanz.de

Abstract. This paper provides an overview of different types of information networks and categorizes them by identifying several key properties of information units and relations. These properties reflect the expressiveness and thus ability of an information network to model data of a diverse nature.

Keywords: Information Networks, Data Integration.

1 Introduction

During the past years information networks have gained more and more attention in various application areas ranging from the formal modeling of conceptual hierarchies to tools for semantic-free data integration. Especially in the biomedical domain a number of different types of information networks have been proposed in the last years [1]. This area of research is known for its diverse information sources that need to be considered, for instance in the drug discovery process [2]. The integrated sources range from experimental data, such as gene expression results, up to highly curated ontologies, such as the ontology of Medical Subject Headings.

Information networks are composed of information units representing physical items, more generally named entities or simply concepts and relations representing semantic or solely correlational connections between information units. They are almost always based on a graph structure with vertices and edges, where vertices represent units of information e.g. genes, proteins or diseases and the relations between these units of information are usually represented by edges. In some information networks relations are represented by vertices as well, and therefore apply a bi-partite graph representation. This type of representation has the added advantage that relations between more than two information units can be easily supported. An edge can be directed or undirected depending on the relationship it represents. Most networks also allow additional attributes or properties to be attached to vertices and edges, such as a vertex type describing the nature of the information unit or an edge weight representing the strength of the relation.

Once the data is represented in such an information network the now well-defined structure can be used to discover patterns of interest, extract network summarizations or abstractions and develop tools for the visual exploration of the underlying relations. A general analysis on the structure of complex networks stemming from real-world applications has been conducted by Albert et al. [3]. Such real networks often share a number of common properties such as the small-world property, clustering coefficient or degree distribution. A survey on link mining has been conducted by Getoor and Diehl [4]. They classified the link mining task into three categories: object-related tasks, link-related tasks and graph-related tasks.

Network summarizations representing different levels of detail can be visualized to gain insight into the structure of the integrated data. A review of graph visualization tools for biological networks can be found in [5]. The paper compares the functionality, limitation and specific strength of these tools.

2 Different Categories of Information Network

In order to differentiate information networks distinctions can be made between different properties of information units and relations. These properties are, of course, not exclusive. The properties of an information network define its expressiveness and thus its ability to model data of a diverse nature e.g. ontologies or experimental data.

2.1 Properties of Information Units

The basic information unit does not posses any additional semantical information. However, they will at least include a label attached to them in order to identify the object or concept they represent. Additional properties are the following:

Attributed Units of information that can have additional attributes attached to them. An attribute might be a link to the original data it stems from, or a translation of the original label. These attributes might be considered while reasoning or analyzing the network but do not carry general semantic information, such as the following properties,

Typed Typed information units carry an additional label that is used to distinguish between different semantics of information units e.g. gene or protein. These types can additionally be organized in a hierarchy or ontology.

Hierarchical Hierarchical information units represent a subgraph composed of any number of information units and relations that can be used to condense parts of the network or to represent more complex concepts such as cellular processes.

2.2 Properties of Relations

The basic connection between information units represents a relationship between the corresponding members. They are not required to carry a label.

- **Attributed** Similar to attributed information units relations that have attributes attached to them also fall into this category. Like attributed information units these attributes can be considered during the reasoning process but do not carry a general semantic information.
- **Typed** Equivalent to typed information units, relations can carry a label identifying their type. This attribute is used to distinguish between different semantics of relations such as activates or encodes. These types, as well as typed information units, can be organized in a hierarchy or ontology.
- **Weighted** The weight of a relation is a special type of label that represents the strength of a relation e.g. a number reflecting the probability or strength of a correlation or some other measure of reliability that allows the integration of facts and pieces of evidence.
- **Directed** Directed relations can be used to explicitly model relationships that are only valid in one direction, such as parent child dependency in a hierarchy.
- **Multi relation** In general, relations are represented as edges supporting only two members. topic maps (see Section 3.3) in contrast represent relations as multi edges supporting any number of members. This allows a more flexible modeling of relationships with any number of members e.g. co-expressed genes of an experiment or authors of a paper. Furthermore connections among relations themselves can be represented. Note that it is complicated to combine this property with the directed property mentioned above. Additional information would need to be provided, such as an embedding graph to identify sources and targets in a relation with more than two members.

3 Prominent Types of Information Networks

3.1 Ontologies

Ontologies are based on typed and directed relations using a controlled vocabulary for information units and relations dedicated to a certain domain. The creation of the curated vocabulary leads in general to a manual or semi-automatic creation of an ontology, requiring a comprehensive knowledge of the area to be described.

Figure 1 depicts a simple ontology where information units are represented as nodes and relations are represented as labeled arrows.

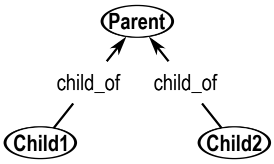


Fig. 1. Example of an ontology

In the area of life science particularly, many ontologies have been developed to share data from diverse research areas such as chemistry, biology or pharmacokinetic. One of the probably best known and most integrated ontologies in the biological field ist the Gene Ontology (GO)[6]. The GO consists of three main ontologies describing the molecular function, biological process and cellular component of genes.

An attempt to integrate diverse ontologies has been made by the Open Biomedical Ontologies (OBO) consortium [7]. They have created a file exchange format and over 60 ontologies for different domains defining a general vocabulary that can be used by other systems.

A classification of biomedical ontologies has been accomplished by Bodenreider [8]. He classified these ontologies into three major categories: knowledge management; data integration, exchange and semantic interoperability; decision support and reasoning.

An ontology-based data integration platform is described in [9]. The authors describe a system that extends the existing text-mining framework ONDEX. ONDEX uses a core set of ontologies, which are aligned by several automated methods to integrate biological databases. The existing system is extended to support not only the alignment and integration of texts but heterogeneous data sources. The data is represented as a graph with attributed edges.

Tzitzikas et al. [10] describe a system that is based on the hierarchical integration of ontologies from different data sources. The system uses a mediator ontology, which bridges the heterogeneity of the different data source ontologies.

3.2 Semantic Networks

Semantic Networks use typed relations to model the semantic of the integrated information units and their relations. Information units in Semantic Networks in contrast to ontologies are not represented by a curated vocabulary but rather described by attaching any number of attributes to them whose semantic is defined by the type of the relation.

Most of the Semantic Networks rely on Semantic Web [11] technologies such as the Resource Description Framework (RDF), RDF Vocabulary Description Language (RDF Schema) and the Web Ontology Language (OWL) defined by the W3C consortium[1].

RDF is a knowledge representation and storage framework that uses triples. A triple consists of a subject, predicate and object. The subject and object are information units that are connected with a directed relation defined by the predicate.

In figure 2 subjects and objects that are uniquely identifiable are depicted in ellipses, whereas objects containing values are depicted in boxes. Predicates are shown as arrows pointing from the object to the subject with the type of the relation as an annotation.

[1] http://www.w3.org/2001/sw/

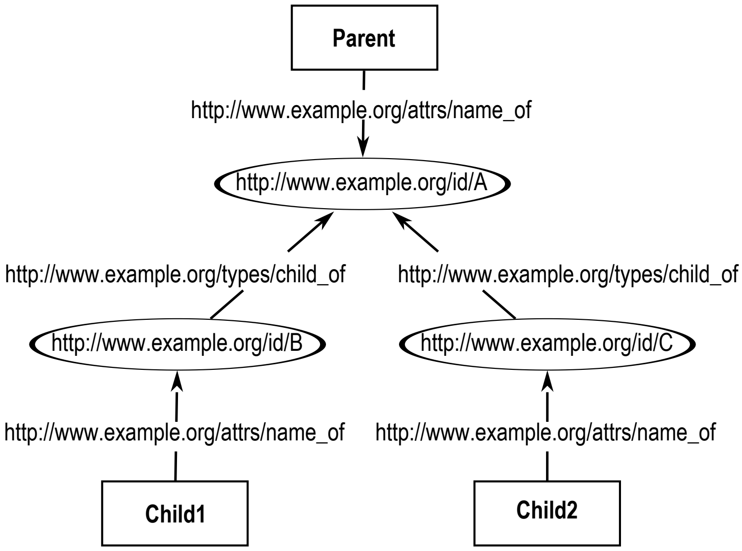

Parent
http://www.example.org/attrs/name_of
http://www.example.org/id/A
http://www.example.org/types/child_of
http://www.example.org/types/child_of
http://www.example.org/id/B
http://www.example.org/id/C
http://www.example.org/attrs/name_of
http://www.example.org/attrs/name_of
Child1
Child2

Fig. 2. Example of a Semantic Web

RDF Schema defines a core vocabulary that can be used to describe properties and classes. These properties and classes can be used to describe the members of a triple. OWL extends RDF Schema by providing a set of additional standard terms to describe properties and classes in more detail such as relations between classes. It also defines the behavior of properties e.g. symmetry or transitivity. OWL as well as RDF Schema extend RDF by providing the means to model the semantics of the integrated data therefore enabling machines to make sense of the data. They both are described using the RDF.

Bales and Johnson [12] analyzed large semantic networks created from 1998-2005 that involve both a graph theoretic perspective and semantic information. The results indicate that networks derived from natural language share common topological properties, such as scale-free and small-world characteristics.

An introduction to semantic networks and semantic graph mining is provided in [13]. In four case studies, they demonstrate the usage of semantic web technologies to analyze disease-causal genes, GO category cross-talks, drug efficacy and herb-drug interactions.

Belleau et al.[14] propose the Bio2RDF project to integrate data from different biological sources. Bio2RDF is used to integrate data from more than twenty different public bioinformatic sources by converting them into the RDF format.

YeastHub [15] another RDF-based data integration approach likewise integrates the data from heterogeneous sources into a RDF-based data warehouse. In addition they propose a standard RDF format for tabular data integration. The format can be used to convert any data table into a standardized RDF format.

A loosely coupled integration of semantic networks is proposed by Smith et al [16] in the form of the LinkHub system. The system consists of smaller networks that can be connected by sharing a common hub. Thus independently maintained networks can be connected to the whole system by connecting them to one of the already integrated sub networks.

Biozon [17] combines the flexible graph structure with an ontology for vertex and edge types similar to the semantic web approach. This combined approach allows a more detailed description of a biological entity by either imposing more constraints on its nature in the hierarchy or on the structure of its relations to other entities in the graph. All vertices within Biozon are direct analogs to physical entities and sets of entities. Proteins as an example are identified by their sequence of amino acids. In contrast to pure semantic networks Biozon allows any number of attributes to be attached to information units as well as to relations.

3.3 Topic Maps

Topic maps [18] use typed information units and relations. Furthermore topic maps support the modeling of multi relations with any number of members. The semantic of a topic is described by attaching any number of attributes to it.

Figure 3 depicts the three major elements of a topic map: topics (ellipses), associations (solid lines) and occurrences (boxes). Association and occurrence types are connected by the dashed lines whereas occurrences are connected by the dotted line.

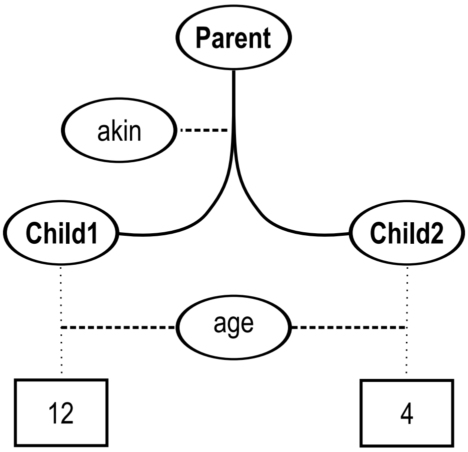


Fig. 3. Example of a topic map

A topic can generally be anything, for example a person, a concept or an idea. Topics can have zero or more topic types assigned which are, in turn, defined as topics describing the semantics of the topic such as gene or protein.

Relations between any number of topics are represented by so-called associations. Associations have a type assigned that describes the association in more detail. Members of associations play a certain role defined by the association role. As with topic and occurrence types association types and association roles are defined as topics themselves. In order to attach attributes to an association it needs to be converted into a topic by the act of reification.

Information resources that represent a topic or describe it in more detail are linked to topics by so-called occurrences. Occurrences are not generally stored in the topic map itself but are referenced using mechanisms supported by the system e.g. Uniform Resource Identifiers. Occurrences can have any number of different types, so-called occurrence types, that describe their semantics. These types are also defined as topics.

Topic maps are self-documenting due to the fact that virtually everything in topic maps is a topic in the map itself, forming the ontology of the used topics and relation types.

An example of a topic-map-like data integration approach is PathSys[19]. In PathSys a relation is also represented as a vertex. This approach models relationships between relations themselves. To distinguish between information units and relations they introduce vertex types. Besides primary vertices representing information units and connector vertices representing relationships, they also introduce graph vertices. By introducing graph vertices, PathSys combines the multi relation property of topic maps with the hierarchical information unit property allowing the representation of subgraphs to describe more complex objects such as protein complexes or cellular processes.

3.4 Weighted Networks

In most weighted networks the edge weight represents the strength of a relation such as reliability or probability. Weighted networks often exhibit additional properties such as types in order to be more expressive by modeling the semantic of the integrated data sources. They support generally only relationships with two members represented by the edges of the graph.

Figure 4 depicts a weighted networks modeling the probability of the cooccurrence of the 3 nodes.

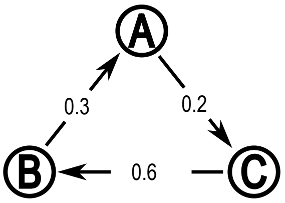


Fig. 4. Example of a weighted network

Heuristic weights Heuristic weights are mostly used to model the reliability or relevance of a given relation. Thus allowing the integration of well-curated sources such as ontologies and pieces of evidence such as noisy experimental data in a single network.

In order to integrate data from diverse biological sources for protein function prediction, Chua et al. [20] propose Integrated Weighted Averaging (IWA). This combines local prediction methods with a global weighting strategy. Each data source is transformed into an undirected graph with proteins as vertices and relationships between proteins as edges. Each source graph has a score reflecting its reliability. Finally, all source graphs are combined in a single graph using IWA.

Kiemer et al. [21] use a weighted network to integrate yeast protein information from different data sources forming a protein-protein interaction network called WI-PHI. The network consists of 50000 interactions from all data sources. The edge weight of the WI-PHI network is computed using the socio-affinity index [22], quantifying the propensity of proteins to form partnerships, multiplied by a weight constant per integrated data source defining its accuracy.

In Biomine [23] the edge weight is a combination of three different weights: reliability, relevance and rarity. Reliability reflects the reliability of the source the edge stems from. The relevance can be changed by the user to reflect current interests; rarity is computed using the degree of the incident vertices. Edges that connect vertices with a low degree have a higher rarity score than edges that connect vertices with a high degree. Vertices and edges have a type assigned describing their nature. Each edge has its inverse edge with a natural inverse type such as “coded by” and “is referred by”. Thus forming a weighted undirected graph with directed edge types.

Probabilistic weights Probabilistic networks model the probability of the existence of a relationship. They are mostly used in the biological field to model interaction networks e.g. gene-gene or protein-protein interaction networks.

Franke et al. [24] use a three-step data integration process using naive Bayesian networks to fuse the information from the GO with microarray co-expression results and protein-protein interaction data. The resulting network called Genenetwork can be used to detect genes that are related to a disease based on genetic mutation.

In [25] Li et al. use a two-layered approach to integrate gene relations from heterogeneous data sources. The first layer creates a fully connected Bayesian network for each integrated source which represents the gene functional relations. The second layer combines these relations from the different data sources into one integrated network using a naive Bayesian method.

Jansen et al. [26] likewise propose a combination of naive Bayesian networks and fully connected Bayesian networks to create a protein-protein interaction network. They use the fully connected Bayesian networks to integrate experimental interaction data and naive Bayesian networks to incorporate other genomic features such as the the biological process from the GO. To combine all results they use a naive Bayesian network as well.

In [27] Troyanskaya et al. introduce MAGIC (Multisource Association of Genes by Integration of Clusters). For each integrated data source, MAGIC creates a gene-gene relationship matrix to predict the functional relationship of two given genes. The matrices are generated from diverse high-throughput techniques such as gene expression microarrays. These gene-gene relationship matrices are weighted by the confidence in the integrated source and combined into a single matrix. This approach allows genes to be members of more than one group, which subsequently allows fuzzy clustering.

4 Conclusion

In this paper we identified several key properties of information units and relations used in information networks. We provided an overview of different types of information networks and categorized them based on the identified properties. These supported properties reflect the expressiveness and thus ability of an information network to model data of a diverse nature. We believe that future networks need to support most of the identified properties to integrate facts and pieces of evidence from heterogeneous sources in order to support the discovery of connections between concepts from diverse areas ultimately supporting the creative thinking.

5 Acknowledgment

The work presented in this paper was supported by the grant of the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898.

References

1. Kwoh, C.K., Ng, P.Y.: Network analysis approach for biology. Cell Mol Life Sci **64**(14) (Jul 2007) 1739–1751
2. Burgun, A., Bodenreider, O.: Accessing and integrating data and knowledge for biomedical research. Yearb Med Inform (2008) 91–101
3. Albert, R., Albert-Laszlo, B.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74** (June 2002) 47–97
4. Getoor, L., Diehl, C.: Link mining: a survey. ACM SIGKDD Explorations Newsletter **7**(2) (2005) 3–12
5. Pavlopoulos, G., Wegener, A.L., Schneider, R.: A survey of visualization tools for biological network analysis. BioData Min **1**(1) (Nov 2008) 12
6. Consortium, G.O.: Creating the gene ontology resource: design and implementation. Genome Res **11**(8) (Aug 2001) 1425–1433

7. Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L.J., Eilbeck, K., Ireland, A., Mungall, C.J., Consortium, O.B.I., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.A., Scheuermann, R.H., Shah, N., Whetzel, P.L., Lewis, S.: The obo foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol **25**(11) (Nov 2007) 1251–1255
8. Bodenreider, O.: Biomedical ontologies in action: role in knowledge management, data integration and decision support. Yearb Med Inform (2008) 67–79
9. Koehler, J., Rawlings, C., Verrier, P., Mitchell, R., Skusa, A., Ruegg, A., Philippi, S.: Linking experimental results, biological networks and sequence analysis methods using ontologies and generalised data structures. In Silico Biol **5**(1) (2005) 33–44
10. Tzitzikas, Y., Constantopoulos, P., Spyratos, N.: Mediators over ontology-based information sources. In: WISE (1). (2001) 31–40
11. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. (May 2001)
12. Bales, M.E., Johnson, S.B.: Graph theoretic modeling of large-scale semantic networks. J Biomed Inform **39**(4) (Aug 2006) 451–464
13. Chen, H., Ding, L., Wu, Z., Yu, T., Dhanapalan, L., Chen, J.Y.: Semantic web for integrated network analysis in biomedicine. Brief Bioinform **10**(2) (Mar 2009) 177–192
14. Belleau, F., Nolin, M.A., Tourigny, N., Rigault, P., Morissette, J.: Bio2rdf: towards a mashup to build bioinformatics knowledge systems. J Biomed Inform **41**(5) (Oct 2008) 706–716
15. Cheung, K.H., Yip, K.Y., Smith, A., Deknikker, R., Masiar, A., Gerstein, M.: Yeasthub: a semantic web use case for integrating data in the life sciences domain. Bioinformatics **21 Suppl 1** (Jun 2005) i85–i96
16. Smith, A.K., Cheung, K.H., Yip, K.Y., Schultz, M., Gerstein, M.K.: Linkhub: a semantic web system that facilitates cross-database queries and information retrieval in proteomics. BMC Bioinformatics **8 Suppl 3** (2007) S5
17. Birkland, A., Yona, G.: Biozon: a system for unification, management and analysis of heterogeneous biological data. BMC Bioinformatics **7** (2006) 70
18. Pepper, S.: The tao of topic maps: finding the way in the age of infoglut. In: Proceedings of XML Europe. (2000)
19. Baitaluk, M., Qian, X., Godbole, S., Raval, A., Ray, A., Gupta, A.: Pathsys: integrating molecular interaction graphs for systems biology. BMC Bioinformatics **7** (2006) 55
20. Chua, H.N., Sung, W.K., Wong, L.: An efficient strategy for extensive integration of diverse biological data for protein function prediction. Bioinformatics (Nov 2007)
21. Kiemer, L., Costa, S., Ueffing, M., Cesareni, G.: Wi-phi: A weighted yeast interactome enriched for direct physical interactions. Proteomics (Feb 2007)
22. Gavin, A.C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., Rau, C., Jensen, L.J., Bastuck, S., Dmpelfeld, B., Edelmann, A., Heurtier, M.A., Hoffman, V., Hoefert, C., Klein, K., Hudak, M., Michon, A.M., Schelder, M., Schirle, M., Remor, M., Rudi, T., Hooper, S., Bauer, A., Bouwmeester, T., Casari, G., Drewes, G., Neubauer, G., Rick, J.M., Kuster, B., Bork, P., Russell, R.B., Superti-Furga, G.: Proteome survey reveals modularity of the yeast cell machinery. Nature **440**(7084) (Mar 2006) 631–636
23. Sevon, P., Eronen, L., Hintsanen, P., Kulovesi, K., Toivonen, H.: Link discovery in graphs derived from biological databases. Data Integration in the Life Sciences **4075** (2006) 35–49

24. Franke, L., van Bakel, H., Fokkens, L., de Jong, E.D., Egmont-Petersen, M., Wijmenga, C.: Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. Am J Hum Genet **78**(6) (Jun 2006) 1011–1025
25. Li, J., Li, X., Su, H., Chen, H., Galbraith, D.W.: A framework of integrating gene relations from heterogeneous data sources: an experiment on arabidopsis thaliana. Bioinformatics **22**(16) (Jul 2006) 2037–2043
26. Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N.J., Chung, S., Emili, A., Snyder, M., Greenblatt, J.F., Gerstein, M.: A bayesian networks approach for predicting protein-protein interactions from genomic data. Science **302**(5644) (Oct 2003) 449–453
27. Troyanskaya, O.G., Dolinski, K., Owen, A.B., Altman, R.B., Botstein, D.: A bayesian framework for combining heterogeneous data sources for gene function prediction (in saccharomyces cerevisiae). Proc Natl Acad Sci U S A **100**(14) (Jul 2003) 8348–8353

"BisoNet" Generation using Textual Data

Marc Segond and Christian Borgelt

European Center for Soft Computing
Calle Gonzalo Gutiérrez Quirós s/n, E-33600 Mieres (Asturias), Spain
{marc.segond,christian.borgelt}@softcomputing.es

Abstract. According to Koestler, the notion of a *bisociation* denotes a connection between pieces of information from habitually separated domains or categories. In this paper, we consider a methodology to find such bisociations using a network representation of knowledge, which is called a *BisoNet*, because it promises to contain bisociations. In a first step, we consider how to create BisoNets from several textual databases taken from different domains using simple text-mining techniques. To achieve this, we introduce a procedure to link nodes of a BisoNet and to endow such links with weights, which is based on a new measure for comparing text frequency vectors. In a second step, we try to rediscover known bisociations, which were originally found by a human domain expert, namely indirect relations between migraine and magnesium as they are hidden in medical research articles published before 1987. We observe that these bisociations are easily rediscovered by simply following the strongest links. Future work includes extending our methods to non-textual data, improving the similarity measure, and applying more sophisticated graph mining methods.

1 Introduction

The concept of association is at the heart of many of today's powerful ICT technologies such as information retrieval and data mining. These technologies typically employ "association by similarity or co-occurrence" in order to discover new information that is relevant to the evidence already known to a user.

However, domains that are characterized by the need to develop innovative solutions require a form of creative information discovery from increasingly complex, heterogeneous and geographically distributed information sources. These domains, including design and engineering (drugs, materials, processes, devices), areas involving art (fashion and entertainment), and scientific discovery disciplines, require a different ICT paradigm that can help users to uncover, select, re-shuffle, and combine diverse contents to synthesize new features and properties leading to creative solutions. People working in these areas employ creative thinking to connect seemingly unrelated information, for example, by using metaphors or analogical reasoning. These modes of thinking allow the mixing of conceptual categories and contexts, which are normally separated. The functional basis for these modes is a mechanism called *bisociation* (see [1]).

According to Arthur Koestler, who coined this term, *bisociation* means to join unrelated, and often even conflicting, information in a new way. It means being "double minded" or able to think on more than one plane of thought simultaneously. Similarly, Frank Barron [2] says that the ability to tolerate chaos or seemingly opposite information is characteristic of creative individuals.

Several famous scientific discoveries are good examples of bisociations, for instance Isaac Newton's theory of gravitation and James C. Maxwell's theory of electromagnetic waves. Before Newton, a clear distinction was made between *sub-lunar* (below the moon) and *super-lunar physics* (above the moon), since it was commonly believed that these two spheres where governed by entirely different sets of physical laws. Newton's insight that the trajectories of planets and comets can be interpreted in the same way as the course of a falling body joined these habitually separated domains. Maxwell, by realizing that light is an electromagnetic wave, joined the domains of optics and electromagnetism, which, at his time, were also treated as unrelated areas of physical phenomena.

Although the concept of bisociation is frequently discussed in cognitive science, psychology and related areas (see, for example, [1–3]), there does not seem to exist a serious attempt at trying to formalize and computerize this concept. In terms of ICT implementations, much more widely researched areas include association rule learning (for instance, [4]), analogical reasoning (for example, [5, 6]), metaphoric reasoning (for example, [7]), and related areas such as case-based reasoning (for instance, [8]) and hybrid approaches (for example, [9]).

In order to fill this gap in current research efforts, the BISON project[1] was created. This project focuses on a knowledge representation approach with the help of networks of named entities, in which bisociations may be revealed by link discovery and graph mining methods, but also by computer-aided interactive navigation. In this paper we report first results obtained in this project.

The rest of this paper is structured as follows: in Section 2 we provide a definition of the core notion of a *bisociation*, which guides our considerations. Based on this definition, we justify why a network representation—a so-called *BisoNet*—is a proper basis for computer-aided bisociation discovery. Methods for generating BisoNets from heterogeneous data sources are discussed in Section 3, including procedures for selecting the named entities that form its nodes and principles for linking them based on the information extracted from the data sources. In particular, we present a new measure for the strength of a link between concepts that are derived from textual data. Such link weights are important in order to assess the strength of indirect connections like bisociations.

Afterwards, in Section 4 we report results on a benchmark data set (consisting of titles and abstracts of medical research articles), in which a human domain expert already discovered hidden bisociations. By showing that with our system we can create a plausible BisoNet from this data source, in which we can rediscover these bisociations, we provide evidence that the computer-aided search for bisociations is a highly promising technology.

Finally, in Section 5 we draw conclusions from our discussion.

[1] See `http://www.bisonet.eu/` for more information on this EU FP7 funded project.

2 Bisociation and BisoNets

Since the core notion of our efforts is *bisociation*, we start by trying to provide a sufficiently clear definition, which can guide us in our attempts to create a system able to support a user in finding bisociations. A first definition within the BISON project[2] characterizes *bisociation* as follows:

> A *bisociation* is a link L that connects two domains D_1 and D_2 that are unconnected given a specific context or view V by which the domains are defined. The link L is defined by a connection between two concepts c_1 and c_2 of the respective domains.

Although the focus on a connection between two habitually (that is, in the context a user is working in) separated domains is understandable, this definition seems somewhat too narrow. Linking two concepts from the same domain, which are unconnected within the domain, but become connected by employing indirect relations that pass through another domain, may just as well be seen as bisociations. The principle should rather be that the connection is not fully contained in one domain (which would merely be an association), but needs access to a separate domain. Taking this into account, we generalize the definition:

> A *bisociation* is a link L between two concepts c_1 and c_2, which are unconnected given a specific context or view V. The concepts c_1 and c_2 may be unconnected, because they reside in different domains D_1 and D_2 (which are seen as unrelated in the view V), or because they reside in the same domain D_1, in which they are unconnected, and their relation is revealed only through a *bridging concept* c_3 residing in some other domain D_2 (which is not considered in the view V).

In both of these characterizations we define domains formally as sets of concepts. Note that a *bridging concept* c_3 is usually also required if the two concepts c_1 and c_2 reside in different domains, since direct connections between them, even if they cross the border between two domains, can be expected to be known and thus will not be interesting or relevant for a user.

Starting from the above characterization of *bisociation*, a network representation, called a *BisoNet*, of the available knowledge suggests itself: each concept (or, more generally, any named entity) gives rise to a node. Concepts that are associated (according to the classical paradigm of similarity or co-occurrence) are connected by an edge. Bisociations are then indirect connections (technically paths) between concepts, which cross the border between two domains.

Note that this fits both forms of bisociations outlined above. If the concepts c_1 and c_2 reside in different domains, the boundary between these two domains necessarily has to be crossed. If they reside in the same domain, one first has to leave this domain and then come back in order to find a bisociation.

[2] See `http://www.inf.uni-konstanz.de/bisonwiki/index.php5`, which, however, is not publicly accessible at this time.

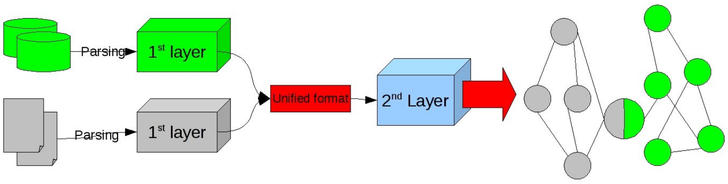


Fig. 1. Illustration of the structure of the BisoNet generator.

3 BisoNet Generation

A system for generating BisoNets requires three ingredients: (1) A component to access the original, usually heterogeneous data sources. In order to cope with different data formats, we suggest, in Section 3.1, a two-layer architecture. (2) A method for choosing the named entities that are to form the nodes of the BisoNet. Here we rely on standard keyword extraction techniques, as discussed in Section 3.2. (3) A procedure for linking the nodes of a BisoNet and for endowing them with weights that indicate the association strength. For this we suggest, in Section 3.3, a new association measure for keywords.

3.1 Data Access and Pre-Processing

As explained above, a BisoNet is a network that promises to contain bisociations. In order to generate such networks, we first have to consider two things: we must be able to read different and heterogeneous data sources, and we have to be able to merge the information derived from them in one BisoNet. Data sources can be databases (relational or of any other type), text collections, raw text, or any data that provide information about a domain. Due to the wide variety of formats a data source can have, the choice we made here is not to provide an interface of maximal flexibility that can be made to read any data source type, but to structure our creation framework into two separate layers.

The first layer directly accesses the data source and therefore has to be newly developed for or at least adapted to the specific format of the data source. The second layer is the actual BisoNet generation part. It takes its information from the first layer, always in the same format, and therefore can generate a BisoNet from any data source, as far as it is parsed and exported in the form provided by the first layer (see Figure 1 for a sketch).

The way data should be provided to the second layer is fairly simple, because in this paper we confine our considerations to textual data. As a consequence, the second layer creates nodes from data that are passed as records containing textual fields. These textual fields can contain, for now, either words or authors names. This procedure and data format is well adapted to textual databases or text collections, but is meant to evolve in future development in order to be able to take other types of data sources into account. However, since most of the data sources that we have used so far were textual data sources, this protocol seems simple and efficient. Future extensions could consist in including raw data fields (for example, to handle images), and will then require an adaptation of the second layer to be able to create nodes from other objects than textual data.

The second layer builds a BisoNet by extracting keywords using standard text mining techniques such as stop word removal and stemming (see [10]). The extracted keywords are weighted by their TFIDF (Text Frequency - Inverse Document Frequency) value (see [11]), thus allowing us to apply a (user-defined) threshold in order to filter the most important keywords, as will be detailed in Section 3.2. Links between nodes are created according to the presence of co-occurrences of the corresponding keywords in the same documents, and are weighted using a similarity measure adapted to the specific requirements of our case, which will be presented in Section 3.3. In the case that author lists are provided with each text string, extracted keywords are also linked to the related authors. These links are weighted according to the number of times a keyword occurs in a given author's work.

3.2 Creating nodes

In our BisoNets nodes represent concepts. As we only talk about textual databases, we made the choice to characterize concepts by keywords that are extracted from the textual records taken from the data sources. In the second layer of our framework, each textual record j is processed with stop word removal algorithm. Then the text frequency values are computed for each remaining term i as shown in Equation 1, where $n_{i,j}$ is the number of occurrences of the considered term in textual record j and $\sum_k n_{k,j}$ is the sum of number of occurrences of all terms in textual record j.

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

Naturally, this procedure of keyword extraction is limited in its power to capture the contents of the text fields. The reason is that we are ignoring synonyms (which should be handled by one node rather than two or more), hyper- and hyponyms, pronouns (which may refer to a relevant keyword and thus may have to be counted for the occurrence of this keyword) etc. However, such linguistic properties are very difficult to take into account and need sophisticated tools (like thesauri etc.). Since such advanced text mining is not the main goal of our work (which rather focuses on BisoNet creation), keeping the processing simple seemed a feasible option. Nevertheless, advanced implementations may require such advanced processing, because ignoring, for example, synonyms and pronouns can distorts the statistics underlying, for instance, the term frequency value: ignoring pronouns that refer to a keyword, or not merging two synonyms makes the TF lower than it should actually be.

After all records have been processed, the inverse document frequency of each keyword i is computed according to Equation 2, where $|D|$ is the total number of records in the database and $|\{d \in D \mid t_i \in d\}|$ is the number of records in which the term t_i appears.

$$\mathrm{idf}_i = \log \frac{|D|}{|\{d \in D \mid t_i \in d\}|} \tag{2}$$

Each node is then weighted with its corresponding average TFIDF value (summing and normalizing all the TF values for a node and then multiplying by the IDF value). This TFIDF approach is a very well known approach in text mining that is easy to implement and makes one able to easily apply a threshold, thus selecting only the most important nodes (keywords). A node then contains, as an attribute, a list of all the TF values for each document of the BisoNet in which its associated keyword appears. This allows us to compute the similarity measure presented in Section 3.3 in order to create links.

According to the definition of a bisociation presented in Section 2, two concepts have to be linked by other concepts that are not in their proper domain (so-called *bridging concepts*). This leads us to introduce the notion of domains, into which the nodes are grouped, so that we can determine when borders between domains are crossed. In order to be able to classify nodes according to their membership in different domains, it is important that they keep, also as an attribute, the domains the data sources belong to, from which they have been extracted. Since the same keyword can occur in several data sources, taken from different domains, one has to be able (for example, for graph mining and link discovery purposes) to know whether a certain keyword has to be considered from a certain domain's point of view. The nodes therefore keep this information as vector of domains their associated keyword belongs to.

This can be interesting, for example, to mine or navigate the BisoNet, keeping in mind that a user may be looking for ideas related to a certain keyword belonging to a domain A. The results of a search for bisociations might also belong to domain A, because it is the domain of interest of the user. However, these results should be reached following paths using keywords from other domains, that is to say bisociations. This procedure provides related keywords of interest for the user, as they belong to its research domain, but they might be also original and new connections as they are the result of a bisociation process.

3.3 Linking nodes

As explained in Section 3.2, nodes are associated with a keyword and a set of documents in which this keyword occurs with a certain term frequency (TF). Practically, this is represented using a vector of real values containing, for each document, the term frequency of the node's keyword. In order to determine whether a link should be created between two nodes or not, and if there is to be a link, to assign it a weight, we have to use a similarity measure to compare two nodes (that is to say: the two vectors of TF values).

One basic metric that directly suggests itself is an adaptation of the Jaccard index (see [12]), shown in Equation 3, to this case. Then $|A \cap B|$ represents the number of elements at the same index that both have a positive value in the two vectors and $|A \cup B|$ the total number of elements in the two vectors.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

This index makes one able to compare two nodes according to the number of similar elements it contains, but does not take into account the importance of the text frequency values. It can also be interpreted as a probability, namely the probability that both elements are positive, given that at least one is positive.

In the Jaccard measure, as applied above, we would consider only whether a vector element is zero or positive and thus neglect the actual value (if it is positive). However, considering two elements at the same index i in two vectors, one way of taking their values into account would be to use their absolute difference (that is, in our case, the absolute difference of the TF values for two terms, but the same document). With this approach, it is easy to compare two vectors (of TF values) by simply summing these values and dividing by the total number of values (or the total number of elements that are positive in at least one vector).

However, this procedure does not properly take into account that both values have to be strictly positive, because a vanishing TF value means that the two keywords do not co-occur in the corresponding document. In addition, we have to keep in mind that having two elements, both of which have a TF value of 0.2, should be less important than having two elements with a TF value of 0.9. In the first case, the keywords associated with the two nodes we are comparing appear only rarely in the document with index i. On the other hand, in the latter case these keywords appear very frequently in this document, which means that they are strongly linked according to this document.

A possibility of taking the TF values itself (and not only their difference) into account is to use the product of the two TF values as a coefficient to the (absolute) difference between the TF values. This takes care of the fact that the two TF values have to be positive, and that the similarity value should be the greater, the larger the TF values are (and, of course, the smaller their absolute difference is). However, in our case, we also want to take into account that it is better to have two similar TF values of 0.35 (which means that the two keywords both appear rather infrequently in the document) than to have TF values of 0.3 and 0.7 (which means the first keywords appears quite rarely, while the other quite frequently).

In order to adapt the product to this consideration, we use the expression in Equation 4, in which k can be adjusted according to the importance one is willing to give to low TF values.

$$\sqrt[k]{\mathrm{tf}_i^A \cdot \mathrm{tf}_i^B} \cdot (1 - |\,\mathrm{tf}_i^A - \mathrm{tf}_i^B\,|), \quad \mathrm{tf}_i^A, \mathrm{tf}_i^B \in [0, 1] \tag{4}$$

Still another thing that we have to take into account in our case is that the same difference between tf_i^A and tf_i^B can have a different impact depending on whether tf_i^A and tf_i^B are large or small. To tackle this issue, we combine Equation 4 with the use of the arctan function, which allows us to end up with a similarity measure shown in Equation 5. This form has the advantage that it takes into account that two TF values for the same index have to be positive, that the similarity should be the greater, the larger the TF values are, and that the same difference between tf_i^A and tf_i^B should have a different impact according to the values of tf_i^A and tf_i^B.

$$\sqrt[k]{\mathrm{tf}_i^A \cdot \mathrm{tf}_i^B} \cdot \left(1 - \frac{|\arctan(\mathrm{tf}_i^A) - \arctan(\mathrm{tf}_i^B)|}{\arctan(1)}\right), \quad \mathrm{tf}_i^A, \mathrm{tf}_i^B \in [0,1] \qquad (5)$$

Links in our BisoNets are weighted using the similarity measure shown in Equation 5. This approach allows us to use several different kinds of graph mining algorithms, such as simply thresholding the values to select a subset of the edges, or more complex ones, like calculating, for example, shortest paths.

4 The Swanson Benchmark

Having shown how BisoNets can be built from textual data sources, we present a benchmark application in this section. The idea is to provide a proof of principle, that this approach of creating a BisoNet can help a user to discover bisociations.

Swanson's approach [13] to literature-based discovery of hidden relations between concepts A and C via intermediate B-terms is the following: if there is no known direct relation A-C, but there are published relations A-B and B-C one can hypothesize that there is a plausible, novel, yet unpublished indirect relation A-C. In this case the B-terms take the role of *bridging concepts*. In his paper [13], Swanson investigated plausible connections between migraine (A) and magnesium (C), based on the titles of papers published before 1987. He found eleven indirect relations (via bridging concepts B) suggesting that magnesium deficiency may be causing migraine.

We tried our approach on the Swansons data source which consists of 8000 paper titles, taken from the PubMed database, published before 1987 and talking about either migraine or magnesium, to see if it was possible to find again these relations between migraine and magnesium. In order to generate a BisoNet, we implemented a parser for text files containing the data from PubMed able to export them in the format understandable by the second layer of our framework. Then, this second layer performed the keywords extraction, using these keywords as nodes and linking these nodes the way described in Section 3.

For testing purpose, we gave the name "domain A" to the data coming from the file concerning migraine, and "domain B" to the data coming from the one concerning magnesium as the purpose here is to discover links crossing domains. Whereas the data of the benchmark was make selecting papers talking about magnesium and paper talking about migraine, we could think about another Swanson-like benchmark selecting papers from the "diseases" domain and others from the "molecules" domain in order to have a larger investigation field.

Given this BisoNet, using a simple threshold filtering the less important nodes and links makes us able to discover indirect relations between magnesium and

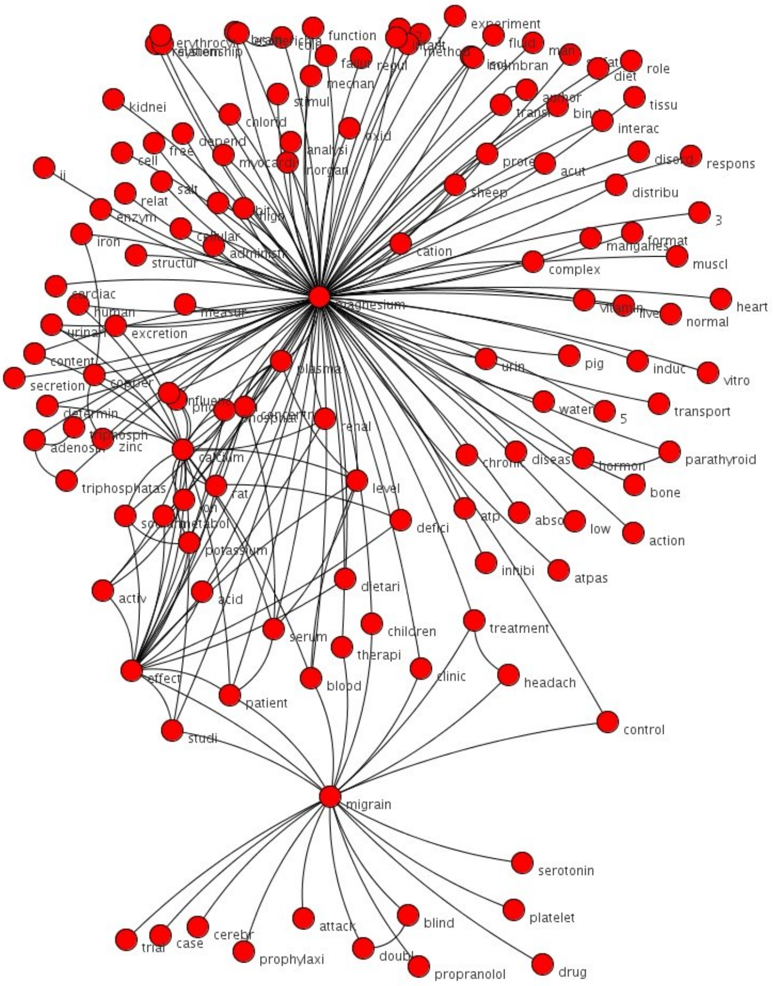


Fig. 2. An example of a BisoNet generated from the Swanson benchmark data sources.

migraine, relations that use keywords, belonging to both domains A and B, such as “deficit”, “headache”, “therapy” or “treatment” to link the two concepts we are talking about. This can be easily seen just looking at Figure 2.

5 Conclusion and further work

In this article, we provided a definition of the notion of a bisociation, as understood by Koestler, which is the key notion of the BISON project. Building on this defintion, we then defined the concept of a BisoNet, which is a network bringing together data sources from different domains, and therefore may help a user to discover bisociations. We presented a way we create nodes using simple text-mining techniques, and a procedure to generate links between nodes, which is based on comparing text frequency vectors using a new similarity measure we introduced.

We then perform a benchmark in order to rediscover bisociations between magnesium and migraine that have been discovered by Swanson using articles published before 1987. We see that bisociations between these two terms are easily discovered using the generated BisoNet, thus indicating that BisoNets are a promising technology for such search.

In summary, we venture to say that this work can be easily applied to any kind of textual data source in order to mine data looking for bisociations, thanks to the two layers architecture implementation. In addition, we are working on generalizing these techniques to non-textual data sources, introducing different types of attributes for the nodes, and therefore, other types of similarity measures in order to link the heterogeneous set of nodes. Further work also consists in performing other benchmarks and applying graph mining algorithms in order to confirm the quality of the so generated BisoNets.

The work presented here was supported by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898.

References

1. Koestler, A.: The act of creation. London Hutchinson (1964)
2. F.Barron: Putting creativity to work. In: The nature of creativity. Cambridge Univ. Press (1988)
3. Cormac, E.M.: A cognitive theory of metaphor. MIT Pess (1985)
4. R. Agrawal, T. Imielinski, A.S.: Mining association rules between sets of items in large databases. In: Proceedings of the ACM SIGMOD internation conference on management of data. (1993) 207–216
5. D.J. Chalmers, R.M. French, D.H.: High-level perception, representation and analogy: a critique of artificial intelligence methodology. Journal of Experimental and Theoretical Artificial Intelligence **4** (1992) 185–211
6. B. Falkenhainer, K.D. Forbus, D.G.: The structure mapping engine: algorithm and examples. Artificial Intelligence **41** (1989) 1–63

7. Barnden, J.: An implemented system for metaphor-based reasoning - with special application to reasoning about agents. In: Lecture Notes in Computer Science. Volume 1562. (1999) 143–153
8. A. Aamodt, E.P.: Case-based reasoning: foundational issues, methodological variations and system approaches. Artificial Intelligence Communications **7**(1) (1994) 39–59
9. A. Cardoso, E. Costa, P.M.F.P.P.G.: An architecture for hybrid creative reasoning. In: Soft Computing in Case Based Reasoning. Springer Berlin/Heidelberg (2000)
10. C.J. van Rijsbergen, S.E. Robertson, M.P.: New models in probabilistic information retrieval. In: British Library Research and Development Report. Number 5587. London British Library (1980)
11. Gerard Salton, M.M.G.: Introduction to modern information retrieval. McGraw-Hill (1983)
12. Jaccard, P.: Étude comparative de la distribution florale dans une portion des alpes et du jura. Bulletin de la Société Vaudoise des Sciences Naturelles **37** (1901) 547–579
13. Don R. Swanson, Neil R. Smalheiser, V.I.T.: Ranking indirect connections in literature-based discovery: The role of medical subject headings. Journal of the American Society for Information Science and Technology (JASIST) **57**(11) (September 2006)

Constructing Information Networks from Text Documents

Matjaž Juršič[1], Nada Lavrač[1,2], Igor Mozetič[1], Vid Podpečan[1], Hannu Toivonen[3]

[1] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia
[3] Dept. of Compute Science, FI-00014 University of Helsinki, Finland
{matjaz.jursic, nada.lavrac, igor.mozetic, vid.podpecan}@ijs.si,
hannu.toivonen@cs.helsinki.fi

Abstract. A major challenge for next generation data mining systems is creative knowledge discovery from diverse and distributed data/knowledge sources. In this task, an important challenge is information fusion of diverse representations into a unique data/knowledge format. This paper focuses on the graph representation of data/knowledge generated from text documents available on the web. The problem addressed is how to efficiently and effectively create an information network, named a BisoNet, from large text corpora. Several options concerning node and arc representation are discussed, and a case study information network is created from articles concerning autism, downloaded from the PubMed repository of medical publications. Open issues and lessons learned concerning representation choices are discussed

1 Introduction

Information fusion can be defined as the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human and automated decision making [Bos07]. Creative knowledge discovery can only be performed on the basis of a sufficiently large and sufficiently diverse underlying corpus of information. The larger the corpus, the more likely it is to contain interesting, still unexplored relationships.

The diversity of data/knowledge sources demands a solution that is able to represent and process highly heterogeneous information in a uniform way. This means that unstructured, semi-structured and highly structured content needs to be integrated. Information fusion approaches are diverse, and domain dependent. For instance, recent investigations in using information fusion to support scientific decision making within bioinformatics include [Dur06, Rac05]. [Smi06] exploit the idea of formulating an ontology-based model of the problem to be solved by the user and interpreting it as a constraint satisfaction problem taking into account information from a dynamic environment.

In this paper, we explore a graph-theoretic approach [Alb02, Bal06] which appears to provide the best framework to accommodate the two dimensions of information source complexity – type diversity as well as volume size. Efficient management and processing of very large graph structures can be realized in suitable distributed computing environments, such as grids, peer-to-peer networks or service-oriented architectures on the basis of modern database management systems, such as XML, object-oriented or graph-oriented database management systems. The still unresolved challenge of graph-theoretic approaches is the creation, maintenance, and update of the graph elements in the case of very large and diverse data/knowledge sources.

This paper focuses on the creation of large graph representations of data/knowledge from text document resources available on the web. The problem addressed is how to efficiently and effectively create an information network, named a BisoNet, from large text corpora. A BisoNet representation, as investigated in the BISON[1] project and discussed in [Ber08] is a graph representation, consisting of labelled nodes and edges. The original idea underlying the BISON project was to have a node for every relevant concept of an application domain, captured by terms denoting these concepts, that is, by "named entities". For example, if the application domain is drug discovery, the relevant (named) entities are diseases, genes, proteins, hormones, chemical compounds etc. The nodes representing these entities are connected if there is evidence that they are related in some way. Reasons for connecting two terms/concepts can be linguistic, logical, causal, empirical, a conjecture by a human expert, a co-occurrence observed in documents dealing with the considered domain. E.g., an edge between two nodes may refer to a document (for example, a research paper) that connects the represented entities.

Open issues in BisoNet creation are how to identify entities and relationships in data, especially from unstructured data like text documents: i.e., which nodes should be created from text documents, what edges should be created, what are the attributes with which they are endowed and how should edge weights be computed. This paper discusses several possible choices that can be made concerning the entities that constitute nodes and edges in a graph when the target knowledge representation is a BisoNet.

Another core question is the granularity chosen for describing the network elements, as well as the diversity of resources. To illustrate a great variety of text sources we use two extreme examples. Firstly, there is a concept of a generic document. We usually do not know much about texts from these sources, sometimes we do not even know which topics they describe. A general document can also contain a lot of noise. Examples of general documents are: a random text from the internet, blogs, newsgroup posts, mobile messages (sms) or mail archives. On the other extreme there are documents from well defined sources. These documents share a predefined

[1] Bisociation Networks for Creative Information Discovery: http://www.BisoNet.eu/.

vocabulary, we precisely know the subject they describe, and usually they are annotated with keywords. Text of this kind is often written by experts in some area who use a similar language to describe similar concepts. Sometimes we can even get access to an ontology or a hierarchy of concepts used in the documents. Examples of these documents are scientific articles from various domains and other documents from well structured and controlled sources (e.g.: encyclopaedia articles).

In this paper we use the example from the second of the two extremes. As a representative of a set of scientific documents we used subsets of medical articles from the PubMed[2] database, in combination with MeSH[3], a controlled vocabulary hierarchical thesaurus. A case study information network is presented, created from articles concerning autism, downloaded from the PubMed repository of medical publications. The open issues concerning representation choices are discussed in substantial detail.

The paper is structured as follows: The second section provides the problem description and outlines the structure of the solution proposed in this paper. The next section sets the standard terminology used in the area of text mining and describes some basic procedures for preprocessing a collection of documents. Definition and representation of network entities is presented in the fourth section. The fifth section explains what types of distance measures can be used with network entities or documents. The next section suggests some tips and practices to be followed when deciding which relations are appropriate for the generated BisoNet. Use case about autism is presented in the seventh section. The last section sketches our plans for future work in the Bison project. Acknowledgements and references are listed at the end of this paper.

2 Problem Description: Creation of BisoNets from Text

When creating large bisociation networks (BisoNets) from texts, we have to address the same two issues as in network creation from any other source: define a method for identifying entities, and define a method for discovering relations between these entities. Since text documents can be acquired from very diverse sources we can apply very diverse techniques to generate BisoNets.

In practice, a workflow for converting a set of documents into a BisoNet is more complex than just identifying entities and relations. We have to be able to preprocess text and filter out noise, to generate a large number of in-memory entities and calculate various distance measures between them effectively. As these tasks are not just conceptually difficult, but also computationally very intensive, a great care is needed when designing and implementing algorithms for BisoNet construction.

[2] PubMed database: http://www.ncbi.nlm.nih.gov/pubmed.
[3] Medical Subject Headings: http://www.ncbi.nlm.nih.gov/sites/entrez?db=mesh.

The currently proposed "text to BisoNet" system, called Texas (Text Assistant), consists of the following modules:

- connect to a data source and collect a set of documents,
- preprocess the documents,
- define network entities (considering background knowledge),
- search / count entities in the text and create the in-memory entity representation,
- define and calculate various measures of similarities/distances between entities,
- establish relations between entities using the calculated measures, and
- output the created BisoNet.

A sample workflow, as implemented in the Orange4WS extension [Pod09] of the Orange data mining toolbox [Dem04], is illustrated in Figure 1 (BOW="bag of words" representation of documents).

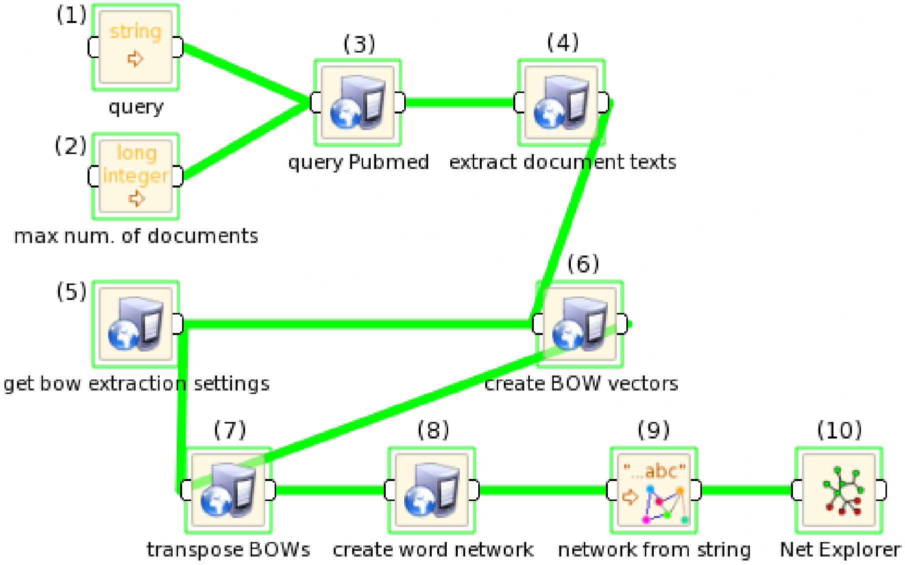


Figure 1: A workflow of text mining algorithms and services.

This paper describes the specific issues that arise when dealing with texts and which can usually not be applied directly to other kinds of databases. The described Texas implementation is built on top of the LATINO[4] library of link analysis and text mining software. This library contains a majority of elementary text mining procedures, but, as the creation of BisoNet is a very specific task (in the text mining world), a lot of modules had to be implemented from scratch or at least optimized considerably.

[4] LATINO library: http://sourceforge.net/projects/latino/.

3 Acquiring a Text Corpus and Creating a BisoNet

This section briefly describes the first and the last step in the workflow of BisoNet creation, i.e., connecting to a data source to collect the documents and the output of the created BisoNet. Since these two issues are mainly technical - they are neither difficult nor computationally expensive - we here only list what our implementation supports and what are the options to be considered.

As there are no standards about the text interchange format in the BISON project and for the sake of simplicity we currently accept just textual and XML files as an input to the procedure. In the future, we can simply add also the following alternatives:

- acquiring documents using soap web services (e.g.: PubMed uses soap web service interface to access their database),
- selecting documents from various SQL bases,
- crawling the internet and gathering documents from web pages. (e.g.: Wikipedia articles), and
- collecting documents from snippets returned from search engines (e.g.: Google snippets).

We have provided the output of the created BisoNets in two different formats:

- the Biomine[5] network file format, used in the Biomine Knowledge discovery in biological databases project [Sev06],
- the Pajek[6] network file format, used in the Pajek program for large network analysis [Bat03],

enabling BisoNet visualization and analysis with Biomine and Pajek, respectively.

In addition to explaining various aspects of preprocessing, this section also briefly describes basic text mining concepts and terminology, some of which are taken from [Fel07].

Preprocessing is the most important part of network extraction from text documents. Its main task is the transformation of unstructured data from text documents into a predefined well-structured document data representation. As shown below, preprocessing is inevitability very tightly connected to the extraction of network entities. In our case, actual network entities are totally defined after preprocessing is finished. The only thing we can later do is to remove some of the useless entities from the set.

In general, the task of preprocessing consists of the extraction of documents' features from documents. The set of all features from document collection is called a representational model. Each document can be presented as a subset of features that it contains. If we write these features of every document in the form of a vector we get the most standard document representation called feature vectors. Given that one of

[5] Biomine project: http://www.cs.helsinki.fi/group/biomine/.

[6] Pajek program: http://pajek.imfm.si/doku.php.

the characteristics of documents’ feature vectors is their sparseness, they are often referred also as sparse vectors. In short, the goal of preprocessing is to extract a sparse feature vector for each document from the given document collection.

Commonly used document features are characters, words, terms and concepts [Fel07]. Characters and words carry little semantic information and are therefore not interesting to consider. On the other hand, terms and concepts carry much more semantic information. Terms are usually considered as single or multiword phrases selected from the corpus by means of term-extraction mechanisms (e.g. because of their high frequency) or are present in an external lexicon of a controlled vocabulary. Concepts or keywords are features generated for documents employing the categorization or annotation of documents. Common concepts are derived from manually annotating a document with some predefined keywords or by inserting a document into some predefined hierarchy. When we refer to document features, we mean terms and concepts that we were able to extract from the documents.

Since high-quality features are hard to acquire, all possible methods that could improve this process should be used at this point. The general approach that usually helps the most is achieved by incorporating background knowledge about the documents and their domain. The most elegant technique to incorporate background knowledge is the use of a controlled vocabulary. Controlled vocabulary is a lexicon of all relevant terms that exist in a given domain. Here we can see a major difference when processing general documents as compared to scientific documents. For many scientific domains there exists not only a controlled vocabulary but also a lot of documents inside scientific article collections are pre-annotated. In this case we can quite easily create feature vectors since we have terms as well as concepts already pre-defined. We just have to find them in the documents. Other interesting approaches to identifying concepts include methods such as KeyGraph [Ohs98], which extract keywords/concepts with minimal assumptions or background knowledge, even from individual documents.

A standard collection of preprocessing techniques [Fel07] is listed below, together with a set of functionalities implemented in our system contains.

- Tokenization: continuous character stream must be broken up into meaningful sub-tokens, usually words or terms in case where a controlled vocabulary is present. Our system uses a standard unicode tokenizer: it partly follows the Unicode Standard Annex #29[7] for Unicode Text Segmentation. The alternative is a more advanced tokenizer which tokenizes strings according to a predefined controlled vocabulary and discards all the other words/terms. Such a tokenizer was used in the test scenario of BisoNet creation from PubMed documents described in Section 8.
- Stopword removal: stopwords are some predefined words from a language that usually carry no relevant information (e.g.: and, or, a, an, ... in English); the usual practice is to ignore them when building a feature set. Our implementation uses a predefined list of stopwords - some common lists that

[7] Unicode Standard Annex #29: http://www.unicode.org/reports/tr29/#Word_Boundaries.

are already included in the library are taken from Snowball[8] - a small string processing language designed for creating stemming algorithms.

- Stemming or lemmatization: the process that converts each word/token into the morphologically neutral form. The following alternatives have been made available: Snowball stemmers, the Porter stemmer [Por80], Lemmagen lemmatizer [Jur07].
- Part-of-speech (POS) tagging: the annotation of words with the appropriate POS tags based on the context in which they appear.
- Syntactical parsing: performs a full syntactical analysis of sentences according to a certain grammar. Usually shallow (not full) parsing is used since it can be efficiently applied to large text corpora.
- Entity extraction: methods that indentify which terms should be promoted as entities and which not. Entity extraction through words grouping into terms using n-gram extraction mechanisms (an n-gram is a sub-sequence of n items from a given sequence) has been implemented.

4 Network Entities

The design choice of our approach is that the entities of the BisoNets will be directly the features of documents, i.e., the terms and concepts, described in the previous section. The following steps are independent of how terms and concepts have actually been identified.

After entities definition one also has to provide some representation of entities in a way which enables efficient calculation of distance measures between them. In the same way as documents are represented as sparse vectors of features (entities), also entities can be represented as sparse vectors of documents. This is illustrated in Example 1: if entity ent_1 is present in documents doc_1, doc_3 and doc_4 then its feature vector would consist of all these documents (with appropriate weights). By analogy to the original vector space - feature space, the newly created vector space is called the document space. While documents “live” in the feature (entity) space, the entities “live” in the document space.

Note that if we write document vectors in the form of a matrix, than the conversion between the feature space and the document space is performed by just transposing the matrix (see Example 1). The only question that remains open for now is what to do with the weights? Is weight $w^f_{x:y}$ identical to weight $w^d_{y:x}$? This depends on various aspects, but mostly on how we define weights of the entities (features) in the first place (when defining document vectors.)

There are four most common weighting models for assigning weights to features:

- Binary: feature weight is either one, if the corresponding feature is present in the document, or zero otherwise.

[8] Snowball: http://snowball.tartarus.org.

- Term occurrence: feature weight is equal to the number of occurrences of this feature.
- Term frequency: weight is derived from the term occurrence by dividing the vector by the sum of all the weights (number of all the features) – it can be also viewed as term occurrence normalized by the Manhattan length of the vector.
- TF-IDF: Term Frequency-Inverse Document Frequency is the most common scheme for weighting features. It is defined as: $w_{x:y}^{TFIDF} = \text{TermFreq}(ent_x, doc_y)\log\left(\frac{N}{DocFreq\ (ent_x)}\right)$, where $\text{TermFreq}(ent_x, doc_y)$ is the frequency of feature ent_x inside document doc_y, N is the number of all documents and $DocFreq(ent_x)$ is the number of documents that contain ent_x. The idea behind TF-IDF measure is to lower the weight of features that appear in many documents.

Documents	Extracted entities
doc_1	ent_1, ent_2, ent_3
doc_2	ent_3, ent_4, ent_4
doc_3	ent_1, ent_2, ent_2, ent_5
doc_4	ent_1, ent_1, ent_1, ent_3, ent_4, ent_4

Original documents and extracted entities

Feature space	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	$w^f_{1:1}$	$w^f_{1:2}$	$w^f_{1:3}$		
doc_2			$w^f_{2:3}$	$w^f_{2:4}$	
doc_3	$w^f_{3:1}$	$w^f_{3:2}$			$w^f_{3:5}$
doc_4	$w^f_{4:1}$		$w^f_{4:3}$	$w^f_{4:4}$	

Sparse matrix of documents: $w^f_{x:y}$ denotes the weight (in the feature space) of entity y in the feature vector of document x

Document space	doc_1	doc_2	doc_3	doc_4
ent_1	$w^d_{1:1}$		$w^d_{1:3}$	$w^d_{1:4}$
ent_2	$w^d_{2:1}$		$w^d_{2:3}$	
ent_3	$w^d_{3:1}$	$w^d_{3:2}$		$w^d_{3:4}$
ent_4		$w^d_{4:2}$		$w^d_{4:4}$
ent_5			$w^d_{5:3}$	

Sparse matrix of entities: $w^d_{x:y}$ denotes the weight (in the document space) of document y in the document vector of entity x

Example 1: Conversion between the feature and the document space.

These four methods can be further modified with vector normalization (dividing each vector so that length - usually the Euclidian or Manhattan length - of the vector is 1). If and when this should be done depends on several reasons: one of them is also the decision which distance measure one will use in the next step – the relation identification step. If cosine similarity is used, it actually does not matter if the vectors are pre-normalized, as this is also done during distance calculation. Example 2 shows the four measures in practice – documents are taken from Example 1. Weights are calculated for the feature space and are not normalized.

For testing purposes we have implemented all four weighting models so one can experiment which is the most suitable to some domain. It is also up to workflow designer to decide whether vectors should be normalized or not. Currently we are still researching what to do with weights when we are transforming back and forth between feature space and document space. At this point we leave this decision also to a workflow designer and support three most sensible approaches:

- Leave weights unchanged.
- Leave weights unchanged but normalize the entities vectors after transformation.
- Recalculate all weights according to the new space.

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	1	1	1		
doc_2			1	1	
doc_3	1	1			1
doc_4	1		1	1	

Binary weight

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	1	1	1		
doc_2			1	2	
doc_3	1	2			1
doc_4	3		1	2	

Term occurrence

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	$^1/_3$	$^1/_3$	$^1/_3$		
doc_2			$^1/_3$	$^2/_3$	
doc_3	$^1/_4$	$^2/_4$			$^1/_4$
doc_4	$^3/_6$		$^1/_6$	$^2/_6$	

Term frequency

	ent_1	ent_2	ent_3	ent_4	ent_5
doc_1	$(^1/_3)\cdot\log(^4/_3)$	$(^1/_3)\cdot\log(^4/_2)$	$(^1/_3)\cdot\log(^4/_3)$		
doc_2			$(^1/_3)\cdot\log(^4/_3)$	$(^2/_3)\cdot\log(^4/_2)$	
doc_3	$(^1/_4)\cdot\log(^4/_3)$	$(^2/_4)\cdot\log(^4/_2)$			$(^1/_4)\cdot\log(^4/_1)$
doc_4	$(^3/_6)\cdot\log(^4/_3)$		$(^1/_6)\cdot\log(^4/_3)$	$(^2/_6)\cdot\log(^4/_2)$	

TF-IDF: term frequency – inversed document frequency

Example 2: Weighting models of features in document vectors (from Example 1).

It is worthwhile to notice again the analogy between the feature space and the document space. Although we have developed the methodology for entities network extraction, the developed approach can be used also for document network extraction.

Moreover, both approaches can be used to extract the same network where documents and entities are connected using some special relations.

5 Distance Measures between Vectors

This section describes some distance measures between vectors in either the feature space or the document space. The choice of a preferable distance measure should be tightly connected to the choice of the weighting model. Some of the combinations are very suitable for each other and may even have some understandable interpretation or experimentally evaluated important value, while others may be less appropriate combination pairs. Therefore we also list commonly used pairs of weighting model and distance measure and describe them.

Our implementation is optimized to the calculation of lengths of sparse vectors: $|vec_x|$ and dot products between those vectors: $\mathrm{DotProd}(vec_x,\ vec_y)$. For that reason, we state also how different distance measures are expressed using these two calculations (if applicable for the described measure).

The most common measures in vector spaces, which are also implemented in our system, are the following:

- Dot products: $\mathrm{DotProd}(vec_x,\ vec_y)$.
- Cosine similarity: which is actually dot product normalized by the length of both vectors $\mathrm{CosSim}(vec_x,\ vec_y) = \frac{\mathrm{DotProd}\ (vec_x,\ vec_y)}{|vec_x||vec_y|}$. In the cases where vectors are already normalized, cosine similarity is identical to the dot product.
- Jaccard index: this similarity coefficient measures the similarity between sample sets. It is defined as the size of the intersection divided by the size of the union of the sample sets:
  $\mathrm{JaccInx}(vec_x,\ vec_y) = \frac{|vec_x \cup vec_y| - |vec_x \cap vec_y|}{|vec_x \cup vec_y|} = \frac{\mathrm{DotProd}\ (vec_x,\ vec_y)}{|vec_x|+|vec_y|-\mathrm{DotProd}\ (vec_x,\ vec_y)}$,
  where lengths $|vec_x|$ and $|vec_y|$ are manhattan lengths of these vectors.
- Bisociation index: is the similarity measure defined for the needs of the BISON project. It is explained in more detail in [Bor09]. This measure cannot be expressed by dot product, therefore, the following definition uses the notation derived from Example 1:
  $\mathrm{BisInx}(vec_x,\ vec_y) = \sum_{i=0}^{M}\left(\sqrt[k]{w_{x:i}w_{y:i}}\left(1\ -\ \frac{|\tan^{-1}(w_{x:i}) - \tan^{-1}(w_{y:i})|}{\tan^{-1}(1)}\right)\right)$,
  where *M* is the number of all entities.

Pairs of weighting models for features/entities and distance measures that are usually used together in vector spaces are the following:

- TF-IDF weighting, cosine similarity – this is probably the most commonly used combination for computing similarity in the feature space.
- Binary weighting, dot product – if used in the document space the result is the co-occurrence measure which counts the number of documents where two entities appear together. This is probably the most widely used measure in the document space.
- Term occurrence weighting, dot products – this is another measure of concurrence of entities in same documents. Compared to the previous measure, this one considers also multiple co-occurrence of two entities inside a document and gives them a greater weight in comparison with the case were each appears only once inside the same document.
- Binary weighting, Jaccard index – Jaccard index is defined on the domain of sets, therefore the only reasonable weighting model to use with it is the binary weighting model (since every vector then represents a set of features).
- Term frequency, “Bisociation index” – since Bisociation index was designed with the term frequency weighting in mind, it seems reasonable, to firstly try this combination when determining the weighting model for the Bisociation index.

6 Relations between Entities

At this point a workflow designer has all the required ingredients to create a BisoNet: definition of the entities and the means to calculate distances/similarities between them. This section describes some design techniques to be considered when deciding which of the many possible relations should be included in the network.

Ideas for some of the described approaches were drawn from [Swa06] and its descendant [Pet09]. The main idea of these two articles is to exploit weak relations between entities. This is an innovative and promising attempt to finding interesting – hidden – relations between entities. Hence, we try to simulate this procedure and recreate interesting discoveries made with those algorithms. Consequently, we were encouraged to include also information of weak links into our BisoNet creation procedure.

A common and generally good practice to be followed when creating relations is to annotate them with different types if they are derived using different approaches. In the case one follows this idea, the algorithms of the next step (searching through BisoNets) will have much easier tasks to solve. In such a way one also does not need to worry so much if some relations are unnecessarily defined twice (if the same information comes up using two different techniques), since relations are not merged together but are distinguished by the following algorithms.
We have implemented the following relations/links identifying techniques:

- Strongest links extraction: go through all combinations of pairs of nodes and find the strongest links (usually this means to find relations between most similar entities.) We see at least three options how to accomplish this:
    - The first option is to extract the n strongest links in the whole network.
    - The second option is to extract the m strongest links for every node in the network.
    - The third option is the combination of the first and the second. Retrieve the n strongest links in general and append the m strongest links for each node (if they do not already exist). In this way, the network is connected – every node has minimally m connections, but "stronger" nodes get the opportunity to get better connected than the others.
- Weakest links extraction: find links that have weight more than zero (they exist) but are the weakest among all the links.
    - The three options described in the strongest links extraction can be also applied here.
- Adding links from background knowledge. In the case where we have some background knowledge that already contains links between entities (e.g.: MeSH thesaurus in the case of PubMed articles) we should consider adding them also to the output network.
- Adding inverse vectors. If we are building a network of entities there is also the possibility of adding documents as nodes in the network. Links between entities can be added using numerous described ways, while the relation between entities and documents could be of type "document contains entity". The same conclusion is valid if we are creating a document network – we can add entities. One concern here can be the great number of links added with this approach; however, some filtering techniques may be applied.

Which of these techniques are appropriate and which are not can only be evaluated using advanced BisoNet search/crawler/exploration algorithms and tools. Given that there are many possible combinations of relations to include in the network, also promising subsets should be identified. So far we did not research this issue, as it is conceptually a separate process – compared to generating BisoNets from text documents. In view of the fact that only results from these algorithms will be able to evaluate the entire process of network creation, this is one of the most important items on our future work agenda.

7 The Autism Case Study

The goal of this use case was to construct a BisoNet from PubMed articles on autism. Autistic disorder (also called autism; more recently described as "mindblindedness") is a neurological and developmental disorder that usually appears during the first three years of life. A child with autism appears to live in his/her own world, showing little interest in others, and a lack of social awareness. Autistic children often have problems in communication, avoid eye contact, and show limited attachment to others. However, many persons with autism excel consistently on certain mental tasks (i.e., counting, measuring, art, music, memory).

We applied the above described Texas process to obtain a BisoNet for autism. We retrieved articles about autism from the PubMed database, identified entities in them using the MeSH vocabulary, and derived co-occurrence relations between entities. A part of the resulting BisoNet, as visualized by the Biomine visualization engine [Sev06], is shown in Figure 2.

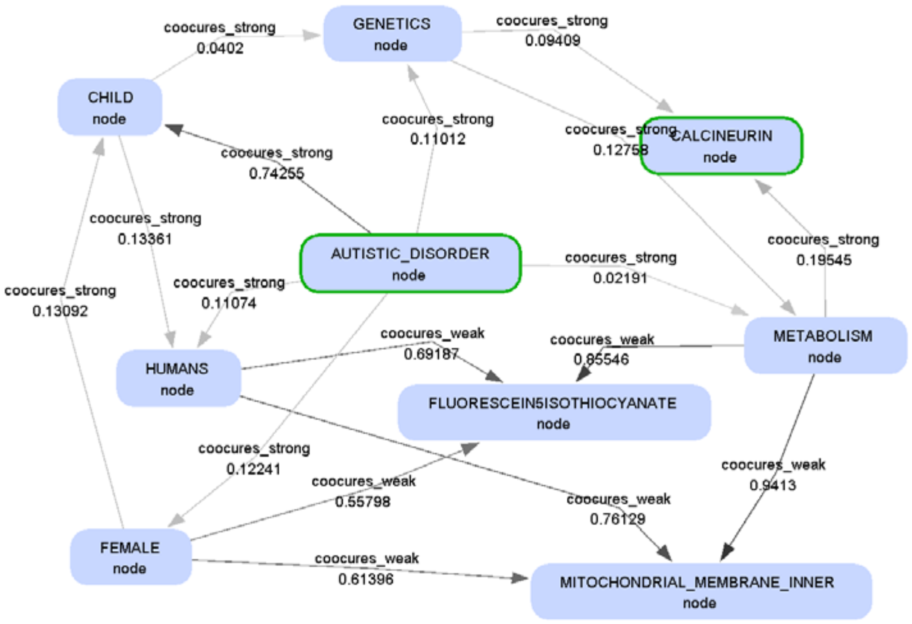


Figure 2: Part of BisoNet, created from PubMed articles on autism.

The cause of autism is not known. Research suggests that autism is a genetic condition, as evidenced by a link between autism and genetics in the BisoNet of Figure 2. It is believed that several genes are involved in the development of autism. Research studies in autism have found a variety of abnormalities in the brain structure and chemicals in the brain; however, there have been no consistent findings. The BisoNet of Figure 2 suggests possible relationships to calcineurin and fluorescensisohticyanate. Ideally, through BisoNet exploration, we hope to discover some still unknown links in this domain.

A part of the BisoNet, created from the PubMed articles on autism, as visualized by the Biomine visualization engine [Sev06], is shown in Figure 2.

8 Future Work

The methodology for creating BisoNets from text, presented in this paper, will be used as a foundation for our forthcoming research on case studies investigated in the BISON project, which include the use of texts in BisoNets. These case studies (benchmarks) will help us not only to validate this methodology, but also to get the overall view of the progress we are doing on bisociation discovery (the core of the BISON project).

The case studies we plan to address using the developed methodology are:

- Migraine treatment and unknown facts detection from the selection of documents out of the PubMed database. The goal of this benchmark is to recreate the Swanson's approach [Swa06] to literature-based discovery of hidden relations between concepts A and C via intermediate B-terms. If there is no known direct relation A-C, but there are published relations A-B and B-C one can hypothesize that there is a plausible, novel, yet unpublished indirect relation A-C. The result of [Swa06] that we want to rediscover is a bisociative link between migraine and magnesium, which was previously unknown.
- Discovery of interesting (previously unstudied) specifics in the domain of autism from the selection of documents out of the PubMed database. This benchmark is about reconstructing the RaJoLink approach [Pet09] to literature-based open discovery process. The Swanson's approach implements closed discovery, the A-B-C process, where A and C are given and one searches for intermediate B concepts. In open discovery, in contrast, only A is given. The RaJoLink idea is to find C via B terms which are rare (and therefore potentially interesting) in conjunction with A.
- Cross contexts (domain) bisociation link discovery in the 20 newsgroups data set[9]. In this setting we want initially to find some mappings between the entities from one domain and equivalent entities from another domain. After identification of such connections, we will try to find bisociations between whole concepts among domains. These bisociations can indicate how to apply solutions of problems from one domain to the open problems of another domain.

We expect that the most time-consuming task during the creation of BisoNets for the above presented case studies will be the definition of the numerous setting at each step of the network creation workflow. Although this paper leaves many such topics unanswered, decisions will have to be made and supported by reasonable arguments.

We will also investigate alternative methods for identifying concepts and discovering relationships between them. In particular, we would like to be able to identify rare but important relationships and separate them from common relationships, even when they are strong. This would give further support to discovery of novel and non-trivial links.

[9] The 20 newsgroups data set: http://people.csail.mit.edu/jrennie/20Newsgroups/.

Acknowledgement

The work presented in this paper was supported by the Slovenian Research Agency grant Knowledge Technologies, and by the grant of the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898. We thank Kimmo Kulovesi for his help and collaboration with using Biomine.

References

Albert, R., Barabasi, A.L.: Statistical mechanics of complex networks. In: Rev Mod Phys, vol. 74(1), pp. 47--97 (2002)

Bales, M.E., Johnson., S.B.: Graph theoretic modeling of large-scale semantic networks. In: Journal of Biomedical Informatics, vol. 39(4), pp. 451--464 (2006)

Batagelj, V., Mrvar, A.: Pajek - Analysis and Visualization of Large Networks. In: Graph Drawing Software, pp. 77--103, (2003)

Berthold, M.R., Dill, F., Kötter, T., Thiel, K.: Supporting Creativity: Towards Associative Discovery of New Insights. In Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD-2008, LNAI 5012, pp. 14--25, (2008)

Borgelt, C., et al.: BISON project Deliverable D2.1: Network Elements. (to appear 2009)

Bostrom, H., et al.: On the definition of information fusion as a field of research. In: Technical report, University of Skovde, School of Humanities and Informatics, Skovde, Sweden (2007)

Demšar, J., Zupan, B., Leban, G.: Orange: From experimental machine learning to interactive data mining. White Paper (2004)

Dura, E., Gawronska, B., Olsson, B., Erlendsson, B.: Towards Information Fusion in Pathway Evaluation: Encoding Relations in Biomedical Texts. In: Proceedings of the 9th International Conference on Information Fusion (2006)

Feldman, R., Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press (2007)

Juršič, M., Mozetič, I., Lavrač., N.: Learning Ripple Down Rules for Efficient Lemmatization. In: Proceedings of the 10th International Multiconference Information Society 2007, vol. A, pp. 206--209 (2007)

Ohsawa, Y., Benson, N.E., Yachida, M.: KeyGraph: Automatic Indexing by Co-occurrence Graph based on Building Construction Metaphor. In: Proceedings of the Advances in Digital Libraries Conference (ADL), pp. 12--18 (1998)

Petrič, I., et. al.: Literature mining method RaJoLink for uncovering relations between biomedical concepts. In: Journal of Biomedical Informatics, vol. 42(2), pp. 219--227 (2009)

Podpečan, V., Žakova, M., Lavrač, N.: Towards a Service-Oriented Knowledge Discovery Platform. In: Proceedings of the Second Service-oriented Knowledge Discovery Workshop at ECML/PKDD - in review (2009)

Porter, M.F.: An algorithm for suffix stripping. In: Program, vol. 14(3), pp. 130--137 (1980)

Racunas, S., Griffin, C.: Logical data fusion for biological hypothesis evaluation. In: Proceedings of the 8th International Conference on Information Fusion (2005)

Sevon, P., Eronen, L., Hintsanen, P., Kulovesi, K., Toivonen, H.: Link discovery in graphs derived from biological databases. In: Proceedings of 3rd International Workshop on Data Integration in the Life Sciences (2006)

Smirnov, A., Pashkin, M., Shilov, N., Levashova, T., Krizhanovsky, A.: Intelligent Support for Distributed Operational Decision Making. In: Proceedings of the 9th International Conference on Information Fusion (2006)

Swanson, D.R., Smalheiser, N.R., Torvik, V.I.: Ranking Indirect Connections in Literature-Based Discovery: The Role of Medical Subject Headings (MeSH). In Journal of the American Society for Information Science and Technology, vol. 57, pp. 1427--1439 (2006)

Gene Analytics: Discovery and Contextualization of Enriched Gene Sets

Nada Lavrač[1,2], Igor Mozetič[1], Vid Podpečan[1], Petra Kralj Novak[1], Helena Motaln[3], Marko Petek[3] and Kristina Gruden[3]

[1] Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia
{nada.lavrac, igor.mozetic, vid.podpecan, petra.kralj}@ijs.si
[2] University of Nova Gorica, Vipavska 13, Nova Gorica, Slovenia
[3] National Institute of Biology, Večna pot 111, Ljubljana, Slovenia
{helena.motaln, marko.petek, kristina.gruden}@nib.si

Abstract. The paper present a preliminary study of creative knowledge discovery through bisociative data analysis. Bisociative reasoning is at the heart of creative, accidental discovery (serendipity), and is focused on finding unexpected links by crossing different contexts. Contextualization and linking between highly diverse and distributed data and knowledge sources is therefore crucial for implementation of bisociative reasoning. In the paper we explore these ideas on the problem of analysis of microarray data. We show how enriched gene sets are found by using ontology information as background knowledge in semantic subgroup discovery. These genes are then contextualized by the computation of probabilistic links to diverse bioinformatics resources. Results of two case studies are used to illustrate the approach.

1 Introduction

Biologists collect large quantities of data from wet lab experiments and high-throughput platforms. Public biological databases, like Gene Ontology, Kyoto Encyclopedia of Genes and Genomes and ENTREZ, are sources of biological knowledge. Since the growing amounts of available knowledge and data exceed human analytical capabilities, technologies that help analyzing and extracting useful information from such large amounts of data need to be developed and used.

The concept of association is at the heart of many of today's ICT technologies such as information retrieval and data mining. However, scientific discovery requires creative thinking to connect seemingly unrelated information, for example, by using metaphors or analogical reasoning. These modes of thinking allow the mixing of conceptual categories and contexts, which are normally separated. The functional basis for these modes is a mechanism called *bisociation* [8]:

> "The pattern underlying ... is the perceiving of a situation or idea, L, in two self-consistent but habitually incompatible frames of reference, M_1 and M_2. The event L, in which the two intersect, is made to vibrate

> simultaneously on two different wavelengths, as it were. While this unusual situation lasts, L is not merely linked to one associative context but bisociated with two."

From the computational point of view, we say that two concepts are bisociated [14] if:

- there is no direct, obvious evidence linking them,
- one has to cross contexts to find the link, and
- this new link provides some novel insight into the problem domain.

We have to emphasize that context crossing is subjective, since the user has to move from his 'normal' context (frame of reference) to an habitually incompatible context to find the bisociative link [2]. Thus, contextualization is one of the fundamental mechanisms in bisociative reasoning. In this paper we present an approach to discovery and contextualization of genes which should help in analysis of microarray data. The approach is based on information fusion, semantic subgroup discovery (by using ontologies as background knowledge in microarray data analysis), and the linking of various publicly available bioinformatics databases. We first explain the basic notions: information fusion, subgroup discovery and semantic subgroup discovery.

1.1 Information fusion

Information fusion can be defined as the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human and automated decision making [1]. Recent investigations in using information fusion to support scientific decision making within bioinformatics include [3, 9]. Smirnov et al. [12] exploit the idea of formulating an ontology-based model of the problem to be solved by the user and interpreting it as a constraint satisfaction problem taking into account information from a dynamic environment.

An approach to the integration of biological databases GO, KEGG and ENTREZ is implemented in the SEGS information fusion engine (Searching for Enriched Gene Sets, [16]). Another, much larger, integrated annotated bioinformatics information source is Biomine [11].

1.2 Subgroup discovery

Subgroup discovery techniques are used to generate explicit knowledge in the form of rules that allow the user to recognize important relationships in a set of class labeled training instances, describing the target property of interest. Consider two applications. In the first one, the induced subgroup describing rules suggest the general practitioner how to select individuals for population screening, concerning high risk for coronary heart disease (CHD) [4]. The rule below describes a group of overweight female patients older than 63 years:

$$\begin{aligned}\text{High\_CHD\_Risk} \leftarrow\ & \text{sex} = \text{female}\ \&\ \text{age} > 63\ \text{years}\ \&\\ & \text{body\_mass\_index} > 25\ kgm^{-2}\end{aligned}$$

In the second application [5], subgroup describing rules suggest genes that are characteristic for a given cancer type (i.e., leukemia cancer) in an application of distinguishing among 14 different cancer types: leukemia, CNS, lung cancer, etc.:

$$\begin{aligned}\text{Leukemia} \leftarrow\ & \text{KIAA0128 is diff\_expressed}\ \&\\ & \text{prostaglandin\_d2\_synthase is not diff\_expressed}\end{aligned}$$

1.3 Semantic subgroup discovery

Semantic subgroup discovery refers to subgroup discovery, where semantically annotated knowledge sources (ontologies) are used as background knowledge in the data mining process. Using the technology of relational subgroup discovery [17], we have developed an approach to information fusion and semantic data mining, enabling background knowledge in the form of ontologies to be used in relational machine learning. The relational subgroup discovery approach, which was successfully adapted and applied to mining of bioinformatics data [15], and further refined in the SEGS algorithm (Searching for Enriched Gene Sets, [16]), is used in the information fusion and semantic subgroup discovery technology described in this paper. Example rules below are induced by a semantic knowledge discovery engine for two cancer types (ALL and AML) and ranked according to the enrichment score. The rules are a conjunction of ontology terms from the GO, KEGG and ENTREZ ontologies:

$$\begin{aligned}\text{ALL} \leftarrow\ & \text{Func('zinc ion binding' \& Comp('chromosomal part')}\\ \text{AML} \leftarrow\ & \text{Func('metal ion binding') \& Comp('cell surface') \&}\\ & \text{Proc('response to pest,pathogen,parasite')}\end{aligned}$$

1.4 Overview of the paper

This paper describes first steps in creative data and knowledge exploration through *semantic subgroup discovery* and contextualization through *link discovery* between diverse bioinformatics databases. The described approach to semantic subgroup discovery employs semantically annotated knowledge sources as background knowledge for subgroup discovery. In this paper we investigate a special subgroup discovery task: the *gene set enrichment* analysis task. A gene set is *enriched* if the genes that are members of the set are statistically significantly differentially expressed compared to the rest of the genes.

The SEGS method [16] uses as background knowledge data from three publicly available, semantically annotated biological data repositories GO, KEGG and ENTREZ. Based on the background knowledge, it automatically formulates biological hypotheses: rules which define groups of differentially expressed genes. Finally, it estimates the relevance (or significance) of the automatically formulated hypotheses on experimental microarray data. The Biomine service [11] provides links to a large number of biomedical resources, complementing our semantic subgroup discovery technology, due to the explanatory potential of additional link discovery and Biomine graph visualization.

The paper is structured as follows. Section 2 gives an overview of five steps in exploratory analysis of gene expression data. Section 3 describes an approach to the analysis of microarray data, using semantic subgroup discovery in the context of gene set enrichment. A novel methodology, a first attempt at bisociative discovery through contextualization, composed of using SEGS and Biomine (SEGS+Biomine, for short) is in Section 4. Two preliminary case studies are presented in Section 5.

2 Exploratory gene analytics

This section describes the methodological ingredients of the semantic subgroup discovery technology, targeted at the analysis of differentially expressed gene sets: gene ranking, the SEGS method for enriched gene set construction, linking of the discovered gene set to related biomedical databases, and finally visualization in Biomine. The shematic overview is in Figure 1.

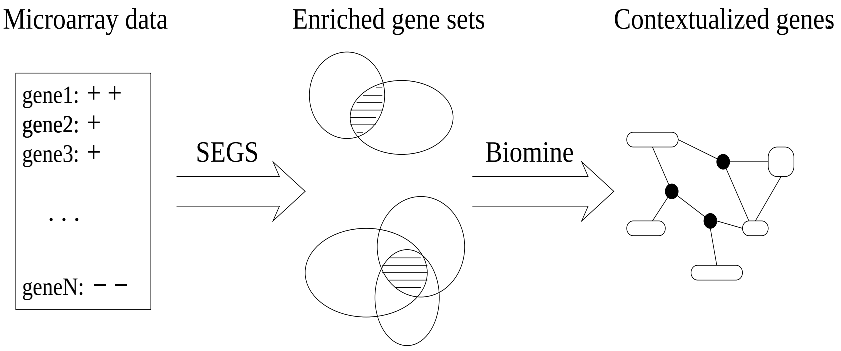


Fig. 1. Microarray gene analytics proceeds by first finding candidate enriched gene sets, expressed as intersections of GO, KEGG and gene-gene interaction sets. Selected enriched genes are then put in context of different bioinformatic resources, as computed by Biomine link discovery engine.

The proposed method consists of the following five steps:

1. **Ranking of genes.** In the first step, class-labeled microarray data is processed and analysed, resulting in a list of genes, ranked according to differential expression.
2. **Ontology information fusion.** A unified database, consisting of GO (processes, functions and components), KEGG (biological pathways) and ENTREZ (gene-gene interactions) terms and relationships is constructed. To this end, a set of scripts was written, enabling easy updating of the integrated database.

3. **Discovering groups of differentially expressed genes.** The ranked list of genes is used as input to the SEGS algorithm [16], an upgrade of the RSD relational subgroup discovery algorithm [15], specially adapted to microarray data analysis. The result is a list of most relevant gene groups that semantically explain differential gene expression in terms of gene functions, components and processes as annotated in biological ontologies.
4. **Finding links between gene group elements**. The elements of the discovered gene groups (GO and KEGG terms or individual genes) are entered as queries to the Biomine crawler. Biomine computes most probable links between these elements and a number of public biological databases. These links help the experts to uncover unexpected relations and biological mechanisms potentially characteristic for the underlying biological processes.
5. **Gene group visualization.** Finally, in order to help in explaining the discovered ontological relationships, the discovered gene relations are visualized using Biomine visualization toolbox.

3 SEGS: Search for Enriched Gene Sets

The goal of gene set enrichment analysis is to find groups of genes—the so-called gene sets—that are enriched. A gene set is *enriched* if the genes that are members of that gene set are statistically significantly differentially expressed compared to the rest of the genes. Two methods for testing the enrichment of gene sets were developed: Gene Set Enrichment Analysis (GSEA) [13] and Parametric Analysis of Gene Set Enrichment (PAGE) [7]. Originally, these methods take terms (gene sets) from the Gene Ontology (GO), the Kyoto Encyclopedia of Genes and Genomes (KEGG) and ENTREZ interactions, and test whether the genes that are annotated by a specific term are statistically significantly differentially expressed in the given dataset.

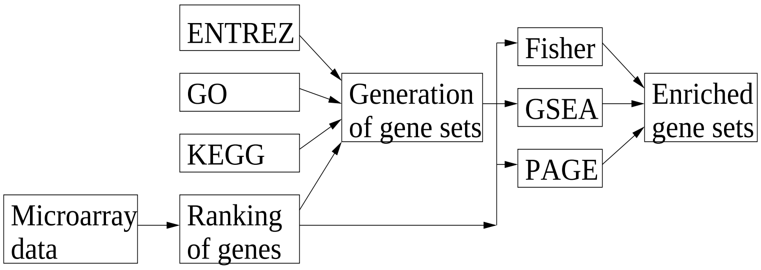


Fig. 2. Schematic representation of the SEGS method.

The novelty of our SEGS method, developed by Trajkovski et al. [16] and used in this study, is that the method does not only test existing gene sets for differential expression but it also generates new gene sets that represent novel biological hypotheses. In short, in addition to testing the enrichment of individual GO and KEGG terms, this method tests the enrichment of newly defined gene sets constructed by the intersection of GO terms, KEGG terms and gene sets defined by taking into account also the gene-gene interaction data from ENTREZ.

The SEGS method has four main components: the background knowledge, the hypothesis language, the hypothesis generation procedure and the hypothesis evaluation procedure. The schematic workflow of the SEGS method is shown in Figure 2.

4 SEGS+Biomine: Contextualization of genes

We made an attempt at exploiting bisociative discoveries within the biomedical domain by explicit contextualization of enriched gene sets. We applied two methods that use publicly available background knowledge for supporting the work of biologists: the SEGS method for searching for enriched gene sets [16] and the Biomine method for contextualization by finding links between genes and other biomedical databases [11]. We combined the two methods in a novel way: we used SEGS for hypothesis generation and evaluation from microarray experimental data, and then input the SEGS results into Biomine for inter-context link discovery and visualization (see Figure 3). We believe that by forming hypotheses with SEGS, constructed as conjunctions of terms from different ontologies (different contexts), discovering links between them by Biomine, and visualizing the SEGS hypotheses and the discovered links by the Biomine graph visualization engine, the interpretation of the biological mechanisms underlying differential gene expression is easier.

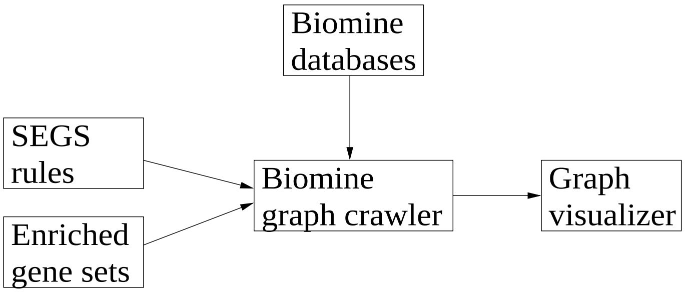


Fig. 3. SEGS+Biomine workflow.

In the Biomine project [11], data from several publicly available databases were merged into a large graph and a method for link discovery between entities in queries was developed. In the Biomine framework vertices correspond to entities and concepts, and edges represent known, annotated relationships between vertices. A link (a relation between two entities) is manifested as a path or a subgraph connecting the corresponding vertices.

Vertex Type	Source Database	Vertices	Degree
Article	PubMed	330,970	6.92
Biological process	GO	10,744	6.76
Cellular component	GO	1,807	16.21
Molecular function	GO	7,922	7.28
Conserved domain	ENTREZ Domains	15,727	99.82
Structural property	ENTREZ Structure	26,425	3.33
Gene Entrez	Gene	395,611	6.09
Gene cluster	UniGene	362,155	2.36
Homology group	HomoloGene	35,478	14.68
OMIM entry	OMIM	15,253	34.35
Protein Entrez	Protein	741,856	5.36

Table 1. Databases included in Biomine.

The Biomine graph data model consists of various biological entities and annotated relations between them. Large, annotated biological data sets can be readily acquired from several public databases and imported into the graph model in a straightforward manner. Some of the databases used in Biomine are summarized in Table 1. Currently, Biomine consists of a total of 1,968,951 vertices and 7,008,607 edges. This particular collection of data sets is not meant to be complete, but it certainly is sufficiently large and versatile for real link discovery.

5 Two case studies

In the first case study, SEGS was applied to find enriched gene sets for distinguishing between two cancer types. In the second case, SEGS and Biomine were combined in order to find an underlying mechanism which might explain why some specific cells are growing faster then the others, in terms of genetic markers.

5.1 Functional genomics

In functional genomics, gene expression monitoring by DNA microarrays (gene chips) provides an important source of information that can help in understanding many biological processes. The database we analyzed consists of a set of gene expression measurements (examples), each corresponding to a large number of measured expression values of a predefined family of genes (attributes). Each measurement in the database was extracted from a tissue of a patient with a specific disease; this disease is the class for the given example. The domain, described in [5, 10] and used in our experiments, is a typical scientific discovery domain characterised by a large number of attributes compared to the number of available examples. As such, this domain is especially prone to overfitting, as it has two different cancer classes and a few training examples, where the examples are described by thousands of attributes presenting gene expression values. While the standard goal of machine learning is to start from the labeled examples and construct models/classifiers that can successfully classify new, previously unseen examples, our main goal is to uncover interesting patterns/rules that can help to better understand the dependencies between classes (diseases) and attributes (gene expressions values).

Gene Set	ES
Enriched in ALL	
1. ALL ← GO_Func('zinc ion binding') & GO_Comp('chromosomal part') & GO_Proc('interphase of mitotic cell cycle')	0.60
2. ALL ← GO_Proc('DNA metabolism')	0.59
3. ALL ← GO_Func('ATP binding') & GO_Comp('chromosomal part') & GO_Proc('DNA replication')	0.55
Enriched in AML	
1. AML ← GO_Func('metal ion binding') & GO_Comp('cell surface') & GO_Proc('response to pest,pathogen,parasite')	0.54
2. AML ← GO_Comp('lysosome')	0.53
3. AML ← GO_Proc('inflammatory response') & GO_Comp('cell surface')	0.51

Table 2. The top most enriched gene sets found in the leukemia dataset with the p-value ≤ 0.001.

Sample top-ranked rules, induced by a semantic knowledge discovery engine for two cancer types (ALL and AML), ranked according to enrichment score (ES), are listed in Table 2. Note that in Table 2 a term *enrichment* is used, meaning the enrichment of differential expression of a set of genes, annotated by the given conjunction of GO, KEGG and/or ENTREZ terms.

5.2 Systems biology

In the systems biology domain, our goal is to help the expert to find a biological interpretation of wet lab experiment results. In the particular experiment, the task is to analyse microarray data in order to distinguish between fast and slowly growing cell lines. The aim of this study was to explain the differences between the cases of fast and slowly growing cell lines through differential expression of gene sets, responsible for cell growth.

Gene Set
1. SLOW-vs-FAST ← GO_Proc('DNA metabolic process') & INTERACT(GO_Comp('cyclin-dependent protein kinase holoenzyme complex'))
2. SLOW-vs-FAST ← GO_Proc('DNA replication') & GO_Comp('nucleus') & INTERACT(KEGG_Path('Cell cycle'))
3. SLOW-vs-FAST ← . . .

Table 3. Top SEGS rules found in the cell growth experiment. The second rule states that one possible distinction between the slow and fast growing cells is in genes participating in the process of DNA replication which are located in the cell nucleus and which interact with genes that participate in the cell cycle pathway.

Table 3 gives the top rules resulting from the SEGS search for enriched gene sets. For each rule, there is a corresponding set of over expressed genes from the experimental data. Figure 4 shows a part of the Biomine graph which links a selected subset of enriched gene set to the rest of the nodes in the Biomine graph.

We believe that SEGS in combination with Biomine may give a wet lab scientist additional hints on what to focus on when comparing the expression data of cells. Additionally, such an in-silico analysis can considerably lower the costs of in-vitro experiments with which the researchers in the wet lab are trying to get a hint of a novel process or phenomena observed. This may be especially true for situations when just knowing the final outcome one cannot explain the drug effect, organ function, or disesase satisfactory, since the gross, yet important characteristics of the cells (organ function) are hidden (do not affect visual morphology) or could not be recognized soon enough. An initial predisposition for this approach is wide accessibility and low costs of high throughput microarray analyses which generate appropriate data for in-silico analyses.

6 Conclusions

A prototype version of the gene analytics software, which enchances SEGS and creates links to Biomine queries and graphs is available as a web application at `http://zulu.ijs.si/web/segs_ga/`.

In the future work we plan to enchance the contextualization of genes with biomedical literature as available in PubMed. To this end, we already have a preliminary implementation of software, called Texas [6], which createas a probabilistic network (BisoNet, compatible to Biomine) from textual sources. By

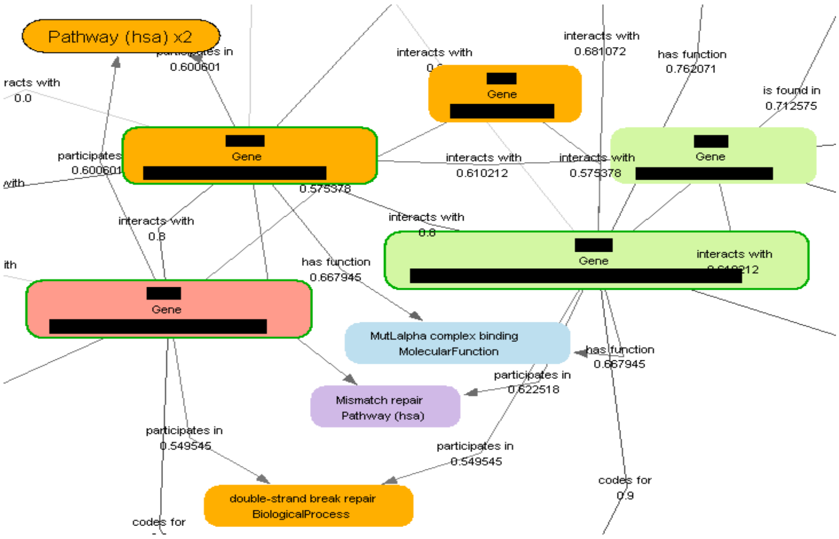


Fig. 4. Biomine subgraph related to three genes from the enriched gene set produced by SEGS. Note that some information is hidden, due to preliminary nature of these results.

focusing on different types of links between terms (e.g., frequent and rare coocurances) we expect to get hints at some unexpected relations between concepts.

Our long term goal is to help biologists at better understanding of intercontextual links between genes and their role in explaining (at least qualitatively) underlying mechanisms which regulate gene expressions.

7 Acknowledgment

The work presented in this paper was supported by the Slovenian Research Agency grant Knowledge Technologies, and by the grant of the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898. We thank Igor Trajkovski for his work on SEGS, and Hannu Toivonen and Kimmo Kulovesi for their help with using Biomine.

References

1. H. Bostrom et al. On the definition of information fusion as a field of research. Technical report, University of Skövde, School of Humanities and Informatics, 2007.
2. W. Dubitzky. Personal communication, FP7 BISON project review, Leuven, June 2009.
3. E. Dura, B. Gawronska, B. Olsson and B. Erlendsson, Towards Information Fusion in Pathway Evaluation: Encoding Relations in Biomedical Texts. Proc. of the 9th International Conference on Information Fusion, 2006.

4. D. Gamberger and N. Lavrač. Expert-Guided Subgroup Discovery: Methodology and Application. *Journal of Artificial Intelligence Research* 17:501–527, 2002.
5. D. Gamberger, N. Lavrač, F. Železný, and J. Tolar. Induction of comprehensible models for gene expression datasets by the subgroup discovery methodology. *Journal of Biomedical Informatics* 37:269–284, 2004.
6. M. Juršič, N. Lavrač, I. Mozetič, V. Podpečan, H. Toivonnen. Constructing information networks from text documents. ECML/PKDD 2009 Workshop "Explorative Analytics of Information Networks", Bled, 2009.
7. S.Y. Kim and D.J. Volsky. PAGE: Parametric Analysis of Gene Set Enrichment. *BMC Bioinformatics* 6:144, 2005.
8. A. Koestler. *The Act of Creation*, The Macmillan Co, New York, 1964.
9. S. Racunas and C. Griffin, Logical data fusion for biological hypothesis evaluation. Proc. of the 8th International Conference on Information Fusion, 2005.
10. S. Ramaswamy et al. Multiclass cancer diagnosis using tumor gene expression signitures. In *Proceedings of the National Academy of Science, USA*, 98(26): 15149–15154, 2001.
11. P. Sevon, L. Eronen, P. Hintsanen, K. Kulovesi, and H. Toivonen. Link discovery in graphs derived from biological databases. In *Proceedings of 3rd International Workshop on Data Integration in the Life Sciences, (DILS'06)*, July 2006. Springer.
12. Smirnov, M. Pashkin, N. Shilov, T. Levashova and A. Krizhanovsky, Intelligent Support for Distributed Operational Decision Making. In: Proceedings of the 9th International Conference on Information Fusion, 2006.
13. P. Subramanian, P. Tamayo, V.K. Mootha, S. Mukherjee, B.L. Ebert, M.A. Gillette, et al. Gene set enrichment analysis: A knowledge based approach for interpreting genome-wide expression profiles. In *Proceedings of the National Academy of Science, USA*, 102(43):15545–15550, 2005.
14. H. Toivonen. Personal communication, FP7 BISON project meeting, Ulster, Sep. 2008.
15. I. Trajkovski, F. Železny, N. Lavrač, and J. Tolar. Learning relational descriptions of differentially expressed gene groups. *IEEE Transactions of Systems, Man and Cybernetics C*, special issue on *Intelligent Computation for Bioinformatics*, 38(1): 16–25, 2008a.
16. I. Trajkovski, N. Lavrač, and J. Tolar. SEGS: Search for enriched gene sets in microarray data. *Journal of Biomedical Informatics*, 41(4):588–601, 2008b.
17. F. Železny and N. Lavrač. Propositionalization-based relational subgroup discovery with RSD. *Machine Learning*, 62(1–2): 33–63, 2007.

Review of Network Abstraction Techniques

Fang Zhou, Sébastien Mahler, Hannu Toivonen

Department of Computer Science and
Helsinki Institute for Information Technology HIIT,
University of Helsinki, Finland
fang.zhou,sebastien.mahler,hannu.toivonen@cs.helsinki.fi

Abstract. Networks are a common way of representing linked information. The goal of network abstraction is to transform a large network into a smaller one, so that the smaller is a useful summary of the original graph.

In this paper we review different approaches and techniques proposed to abstract a large network. We classify the approaches along two axes. The first one consists of elementary simplification techniques used: pruning of (irrelevant) nodes and edges, partitioning to several smaller networks, and generalization by replacement of subnetworks by more general structures. The other axis is objective *vs.* subjective methods; the latter ones aim to maintain more information about those parts of a network that the user has indicated as interesting.

We conclude the review by a brief analysis of which intersections of the two axes are least researched and could therefore have future potential.

1 Introduction

Networks (or graphs) are a common and powerful representation for linked data: nodes represent objects and links represent connections between objects. Example applications are practically infinite; prominent examples include biological networks, social networks, communication networks, and World Wide Web.

Networks are often large. Consider networks of thousands of genes, millions of people, or billions of web pages. While networks are a powerful formalism for handling and analysing such data, they are too large to be viewed or explored by users. One solution is to present to the user an abstract view of the information. We call this *network abstraction.*

The goal of network abstraction is to extract, from a large graph, a graph that is simpler and therefore more useful, even though some information is unevitably lost in the abstraction process – often the explicit aim is to lose (irrelevant) information. An absracted view can help users capture the structure of a huge network, or understand connections between distant nodes, or even discover new knowledge difficult see in a huge graph. This paper is a literature review of some applicable approaches to network abstraction.

Taxonomy of network abstraction methods We classify network abstraction techniques roughly along two orthogonal axes: (1) operations performed, and (2) goals.

Three main types of **operations** to produce abstractions of networks are *prune*, *partition*, and generalize by *replacing*:

1. *Prune peripheral or irrelevant nodes and edges.* This reduces the size of the network, with the aim of keeping only the most interesting or relevant nodes and edges.
2. *Partition the network into smaller ones.* Each smaller subnetwork is now easier to explore individually, while longer connections and larger structures still require looking at several subnetworks.
3. *Replace a part of the network by a more general structure.* Generalization may, for instance, replace a path with a single edge, parallel paths with a single one, or a subgraph by a node, in order to simplify the network.

The **goal** of an abstraction technique can be viewed as either objective or subjective. An *objective* technique disregards user-specific emphasis on any part of the network, while a *subjective* method allows the user to indicate which parts or the network should retain more of their details. For instance, a connection subgraph query returns a network (of a limited size) that maximizes the connectivity between given nodes, and thus is a subjective technique (using pruning).

Bias of the review Although we have aimed at covering representative approaches for network abstraction in general, this review inevitably reflects our own interests. Our motivation is to abstract large information networks such as Biomine[1]. The network model is simply a labeled and weighted graph $G = (V, E)$. Elements of the vertex set V are biological entities, such as genes, proteins, articles, or biological processes, and so on. Edges from the set E have types such as "codes for", "interacts with", or "is homologous to". The interpretation of an edge weight is that it is the probability that the edge exists, i.e., the network is a (Bernoulli) random graph. Biomine currently consists of about 1 million vertices and 10 million edges, making it very hard for experts to analyze without abstraction techniques.

Structure of the review We structure this review first by the objectivity (Section 2) *vs.* subjectivity (Section 3), and then by the operations (in subsections). We conlude with brief notes in Section 4.

2 Objective Methods

In this section, we discuss network abstraction methods where the user has no control over how specific parts of the graph are handled (but there may be numerous other parameters for the user to set).

[1] http://biomine.cs.helsinki.fi/

2.1 Pruning Edges or Nodes

In a complex network, not all nodes or edges are equally important. Removing the most irrelevant or least central nodes or edges can greatly simplify the network structure. In addition to methods directly aimed at network abstraction, ranking nodes from a global viewpoint has been investigated for a long time in the web and social network domains. Such methods may also be used to identify least relevant nodes for pruning. We include such methods in this review.

Relative Neighborhood Graph The Relative Neighborhood Graph (RNG) [1, 2] only contains edges whose two endpoints are relatively close: by definition, nodes a and b are connected by an edge if and only if there is no third node c which is closer to both endpoints a and b than a and b are to eachother. RNG has originally been defined for points, but it can also be used to prune edges between nodes a and b that do have a shared close neighbor c. The relative neighborhood graph then is a superset of the Minimum Spanning Tree (MST) and a subset of Delaunay Triangulation (DT). According to Toussaint [1], RNG can in most cases capture a perceptually more significant subgraph than MST and DT.

Node Centrality The field of social network analysis has produced several methods to measure the importance or centrality of nodes [3–6]. Typical definitions of node importance are the following.

1. Degree centrality simply means that nodes with more edges are more central.
2. Betweenness centrality [7–9] measures how influential a node is in connecting pairs of nodes. A node's betweenness is the number of times the node appears on the paths between all other nodes. It can be computed for shortest paths or for all paths [10]. Computation of a node's betweenness involves all paths between all pairs of nodes of a graph. This leads to high computational costs for large networks.
3. Closeness centrality [11] is defined as the sum of graph-theoretic distances from a given node to all others in the network. The distance can be defined as mean geodesic distance, or as the reciprocal of the sum of geodesic distances. Computation of a node's closeness also involves all paths between all pairs of nodes, leading to a high complexity.
4. Feedback centrality of a vertex is defined recursively by the centrality of its adjacent vertices.
5. Eigenvector centrality has also been proposed [12].

Node centrality measures focus on selecting important nodes, not on selecting a subgraph (of a very small number of separate components). Obviously, centrality measures can be used to identify least important nodes to be pruned. For large input networks and small output networks, however, the result of such straightforward pruning would often consist of individual, unconneted nodes, not an abstract network in the intended sense.

Methods in the following subsections (2.1 and 2.1) are similar in this sense: they help to rank nodes individually based on their importance, but do not as such produce (connected) subgraphs.

PageRank and HITS In Web graph analysis, PageRank algorithm [13, 14] is proposed to find the most important web pages according to the web's link structure. It can be understood as the probability of a random walk on a directed graph; the quality of each page depends on the number and quality of all pages that link to it. It emphasizes highly linked pages and their links. A closely related link analysis method is HITS (Hyperlink-Induced Topic Search) [15, 16]. It also aims to discover web pages of importance. Unlike PageRank, it has two values for each page, and is processed on a small subset of pages, not the whole web. Haveliwala [17] discusses the relative benefits of PageRank and HITS.

In their basic forms, both PageRank and HITS value a node just according to the graph topology. It is relatively easy to add edge weights to them. However, if one already has a (Bernoulli) probabilistic interpretation of edge weights, the extension is less trivial.

Birnbaum's Component Importance Birnbaum importance [18] is directly defined on (Bernoulli) random graphs where edge weights are probabilities of the existence of the edge. The Birnbaum importance of an edge depends directly on the overall effect of the existence of the edge. An edge whose removal has a large effect on the probability of other nodes to be connected, has a high importance. The importance of a node can be defined in terms of the total importance of its edges. This concept has been extended for two edges by Hong and Lei [19].

2.2 Partitioning a Graph

Inside a network, there often are clusters of nodes (called communities in social networks) within which connections are stronger, while connections between clusters are weaker and less frequent. In such a situation, a useful abstraction is to split the network into clusters and present each one of them separately to the user.

Often, the division is a partition of the original network. In this subsection, we discuss two popular approaches, namely graph partitioning and hierarchical clustering, and a method based on edge betweenness. We also touch on the issue of determining the number of components.

Graph Partitioning A prevalent class of approaches to dividing a network to small parts is based on graph partitioning [20, 21]. The basic goal is to divide the nodes into subsets of roughly equal size and minimize the sum of weights of edges crossing different subsets. This problem is NP-complete. However, many algorithms have been proposed to find a reasonably good partition.

Popular graph partitioning techniques include spectral bisection methods [22, 23] and geometric methods [24, 25]. While they are quite elegant, they have some downsides. Spectral bisection in its standard form is computationally expensive for very large networks. The geometric methods in turn require coordinates of vertices of the graph.

The multilevel method [26, 27] first collapses sets of nodes and edges to obtain a smaller graph and partitions the small graph. It then refines the partitioning while projecting the smaller graph back to the original graph. The multilevel method combines a global view with local optimization to reduce cut sizes.

An issue with many of these partitioning methods is that they only bisect networks [28]. Good results are not guaranteed by repeating bisections when more than two subgroups are needed. For example, if the graph essentially has three subgroups, there is no guarantee that these three subgroups can be discovered by finding the best division into two and then dividing one of them again.

Kernighan-Lin (K-L) algorithm [29] is a classical representative for methods that take a rough partitioning as input. It iteratively looks for a subset of vertices, from each part of the given graph, so that swapping them will lead to a partition with smaller edge-cut. It does not create partitions but rather improves them. The first (very!) rough partitioning can be obtained by randomly partitioning the set of nodes. Obviously, a weakness of the The K-L method is that it only has a local view of the problem.

Various modifications of K-L algorithm have been proposed [30, 31], one of them dealing with an arbitrary number of parts [30].

Hierarchical Clustering Another popular technique to divide networks is hierarchical clustering [32]. It computes similarities (or distances) between nodes, for which typical choices include Euclidean distance and Pearson correlation (of neighborhood vectors), as well as the count of edge-independent or vertex-independent paths between nodes.

Hierarchical clustering is well-known for its incremental approach. Algorithms for hierarchical clustering fall into agglomerative or divisive class. In an agglomerative process, each vertex is initially taken as an individual group, then the closest pair of groups is iteratively merged until a single group is constructed or some qualification is met. Newman [33] indicates that agglomerative processes frequently fail to detect correct subgroups, and it has tendency to find only the cores of clusters. The divisive process iteratively removes edges between the least similar vertices, thus it is totally the opposite of an agglomerative method.

Obviously, other clustering methods can be applied on nodes (or edges) as well to partition a graph.

Edge Betweenness One approach to find a partitioning is through removing edges. This is similar to the divisive hierarchical clustering, and is based on the principle that the edges which connect communities usually have high betweenness [34]. Girvan and Newman define edge betweenness as the number of paths that run along that given edge [33]. It can be calculated using shortest-path betweenness, random-walk betweenness and current-flow betweenness. The authors first use edge centrality indices to find community boundaries. They then remove high betweenness edges in a divisive process, which eventually leads to a division of the original network into separate parts. This method has a high computational cost: in order to compute each edge's betweenness, one should consider all paths in which it appears. Many authors have already proposed different approaches to speed up that algorithm [35, 36].

Number of Subgroups When partitioning a large network into subgroups, how many subgroups should there be? Some methods depend on user's input, some others compute an objective measurement called modularity Q [28, 33, 37]: it is the difference between the actual and the expected fractions of edges within the clusters. A large positive modularity indicates that there are more edges within clusters than we would expect on the basis of chance. Another measure of the quality of graph fragmentation [38] considers both size and shape of clusters.

2.3 Replacing Subgraphs

The third operation in our taxonomy is replacement of a subgraph by a more general one, e.g., of a set of closely related nodes by a single representative. This operation allows to focus on the larger structures and connections in a graph.

Clustering In section 2.2, we already discussed techniques used to discover clusters (communities) in a network. Clustering methods, especially those that identify dense subgraphs, can also be used in an opposite way: we can replace a dense cluster by a single node, so the overall structure of the network becomes clearer.

Frequent Subgraphs A frequent subgraph may be considered a general pattern whose instances can be replaced by a label of that pattern (i.e., single a node or an edge representing the pattern). Motivation for this is two-fold. Technically, this operation can simply be seen as compression; on the other hand, frequent patterns possibly reflect some semantic structures of the domain and therefore are useful candidates for replacement. As a simple example, connections of the type "gene A codes for protein B" are frequent, and they reflect the known relationship between genes and proteins. Depending on the use, it could be useful to abstract a biological graph by collapsing all gene-protein pairs into a single node.

In this subsection, we briefly review frequent subgraph mining, where the goal is to identify subgraphs that appear with a frequency higher than a given minimum frequency (also called support).

Two early methods use frequent probabilistic rules [39] and compression of the database [40]. Some early approaches use greedy, incomplete schemes [41, 42]. Many of the frequent subgraph mining methods are based on the Apriori algorithm [43], for instance AGM [44] and FSG [45, 46]. However, such methods usually suffer from complicated and costly candidate generation, and high computation time of subgraph isomorphism [47]. To circumvent these problems, gSpan [47] explores depth-first search in frequent subgraph mining. CloseGraph [48] in turn mines closed frequent graphs, which reduces the size of output without losing any information. The Spin method [49] only looks for maximal connected frequent subgraphs.

Most of the methods mentioned above consider a database of graphs as input, not a single large graph. More recently, several methods have been proposed to find frequent subgraphs also in a single input graph [50–53].

3 Subjective Methods

In this section, we discuss abstraction methods for which the user can explicitly indicate which parts or aspects are more important, according to his interests. Such network abstraction methods are useful when providing more flexible ways to query a graph (database).

3.1 Pruning Edges or Nodes

Relevant Subgraph Extraction Given two or more nodes, the idea here is to extract the most relevant subnetwork (of a limited size) with respect to connecting the given nodes as strongly as possible. This subnetwork is then in some sense maximally relevant to the given nodes. There are several alternatives for defining the objective function, i.e., the quality of the extracted subnetwork.

An early approache by Grötschel *et al* [54] bases the definition on the count of edge-disjoint or vertex-disjoint paths from the source to the sink. A similar principle has later been applied to multi-relational graphs [55], where a pair of entities could be linked by a myriad of relatively short chains of relationships.

The problem in its general form was later formulated as the connection subgraph problem by Faloutsos *et al.* [56]. The authors also proposed a method based on electricity analogies, aiming at maximizing electrical currents in a network of resistors. However, Tong and Faloutsos later point out the weaknesses of using delivered current criterion as a goodness of connection [57]: it only deals with pair of query nodes, and is sensible to the order of the query nodes. As an improved method, they propose the center-piece subgraph problem to extract a subgraph with strong connections to any arbitrary number of nodes.

For random graphs, work from reliability research suggests network reliability as suitable measure [58]. This is defined as the probability that the given original nodes are connected, given that edges fail randomly according to their probabilities. This approach was then formulated more exactly and algorithms were proposed by Hintsanen and Toivonen [59]. Hintsanen and Toivonen restrict the set of terminals to a pair, and propose two incremental algorithms for the problem.

A logical counterpart of this work, in the field of probabilistic logic learning, is based on ProbLog [60]. In a ProbLog program, each Prolog clause is labeled with a probability. The ProbLog program can then be used to compute the success probabilities of queries. In the theory compression setting for ProbLog [61], the goal is to extract a subprogram of limited size that maximizes the success probability of given queries. The authors use subgraph extraction as the application example.

Detecting Interesting Nodes or Paths Some techniques aim to detect interesting paths and nodes, with respect to given nodes. Lin and Chapulsky [62] focus on determining novel, previously unknown paths and nodes from a labeled graph. Based on computing frequencies of similar paths in the data, they use rarity as a measure to find interesting paths or nodes with respect to the given nodes.

An alternative would be to use node centrality to measure the relative importance; White and Smyth [63] define and compute the importance of nodes in a graph relative to one or more given root nodes. They have also pointed out advantages and disadvantages of such measurement based on shortest paths, k-short paths and k-short node-disjoint paths.

Personalized PageRank On the basis of PageRank, Personlized PageRank (PPR) is proposed to personalize ranking of web pages. It assigns importances according to the query or user preferences. Early work in this area includes Jeh and Widon [64] and Haveliwala [17]. Later, Fogaras *et al* [65] have proposed improved methods for the problem.

An issue for network abstraction with these approaches is that they can idenfity relevant individual nodes, but not a relevant subgraph.

3.2 Partitioning a Graph

We are not aware of subjective partitioning or clustering methods for graphs. Generic clustering methods that allow user input, such as constrained clustering [66] or supervised clustering [67], could be applicable on graphs as well.

3.3 Replacing User Input Subgraph

Some substructures may represent obvious or general knowledge, which may moreover occur frequently. Complementary to the approach of Subsection 2.3 where such patterns are identified automatically, here we consider user-input patterns or replacement rules. Depending on the nature and precision of that input, techniques of substructure searching fall into two categories: exact search and similarity search.

Exact Search Finding all exact instances of a graph structure reduces to the subgraph isomorphism problem, which is NP-complete. Isomorphisms are mappings of node and edge labels that preserve the connections in the subgraph.

Ullmann [68] has proposed a well-known algorithm to number the isomorphisms with a refinement procedure that overcomes brute-force tree-search enumeration. Cordella *et al.* [69] include more selective feasibility rules to prune the state search space of their VF algorithm.

A faster algorithm, GraphGrep [70], builds an index of a database of graphs, then uses filtering and exact matching to find isomorphisms. The database is indexed with paths, which are easier to manipulate than trees or graphs. As an alternative, GIndex [71] relies on frequent substructures to index a graph database.

Similarity Search A more flexible search is to find graphs that are similar but not necessarily identical to the query. Two kinds of similarity search seem interesting in the context of network abstraction. The first one is the K-Nearest-Neighbors (K-NN) query that reports the K substructures which are the most similar to the user's input; the other is the range query which returns subgraphs within a specific dissimilarity range to user's input.

These definitions of the problem imply computation of a similarity measure between two subgraphs. The edit distance between two graphs has been used for that purpose [72]: it generally refers to the cost of transforming one object into the other. For graphs, the transformations are the insertion and removal of vertices and edges, and the changing of attributes on vertices and edges. As graphs have mappings, the edit distance between graphs is the minimum distance over all mappings.

Tian *et al.* [73] propose a distance model containing three components: one measures the structural differences, a second component is the penalty associated with matching two nodes with different labels, and the third component measures the penalty for the gap nodes, nodes in the query that cannot be mapped to any nodes in the target graph.

Another family of similarity measures is based on the maximum common subgraph of two graphs [74]. Fernandez and Valiente [75] propose a graph distance metric based on both maximum common subgraph and minimum common supergraph. The maximum percentage of edges in common has also been used as a similarity measure [76].

Processing pairwise comparisons is very expensive in term of computational time. Grafil [76] and PIS [77] are both based on GIndex [71], indexing the database by frequent substructures.

The concept of graph closure [72] represents the union of graphs, by recording the union of edge labels and vertex labels, given a mapping. The derived algorithm, Closure-tree, organizes graphs in a hierarchy where each node summarizes its descendants by a graph closure: efficiency of similarity query may improve, and that may avoid some disadvantages of path-based and frequent substructure methods.

The authors of SAGA (Substructure Index-based Approximate Graph Alignment) [73] propose the FragmentIndex technique, which indexes small and frequent substructures. It is efficient for small graph queries, however, processing large graph queries is much more expensive. TALE (Tool for Approximate Subgraph Matching of Large Queries Efficiently) [78] is another approximate subgraph matching system. The authors propose to use NH-Index (Neighborhood Index) to index and capture the local graph structure of each node. An alternative approach uses structured graph decomposition to index a graph database [79].

4 Conclusion

There is a large literature on methods suitable for network abstraction. We reviewed some of the most important approaches, classified by whether they allow user focus or not, as well as by the graph modification operations used by them. Even though we did not cover the literature exhaustively, we can try to propose areas for further research based on the gaps and issues observed in the review.

First, we noticed that different node ranking measures (Sections 2.1–2.1) are useful for picking out important nodes, as evidenced by search engines, but the result is just that – a set of nodes. How to better use those ideas to find a connected, relevant subnetwork is an open question.

Second, while there are lots of methods for partitioning a graph (Section 2.2), the computational complexity usually is prohibitive for large graphs such as Biomine, with millions of nodes and edges. Obviously, partitioning would be a valuable tool for network abstraction there.

Third, we observed that some more classical graph problems have been researched much more intensively for graph databases consisting of a number of graphs, rather than for a single large graph. This holds especially for frequent subgraphs (Section 2.3) and subgraph search (Section 3.3).

Fourth, the most obvious gap is for partitioning methods that could be guided by the user (Section 3.2). Constrained or supervised clustering might be provide useful starting points here.

Finally, a practical exploration system needs an integrated approach to abstraction, using several of the techniques reviewed here to complement each other in producing a simple and useful abstract network.

Acknowledgements This work has been supported by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland and by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898.

References

1. Toussaint, G.T.: The relative neighbourhood graph of a finite planar set. Pattern Recognition **12**(4) (1980) 261–268
2. Jaromczyk, J., Toussaint, G.: Relative neighborhood graphs and their relatives. Proceedings of the IEEE **80**(9) (1992) 1502–1517
3. Freeman, L.C.: Centrality in social networks: Conceptual clarification. Social Networks **1**(3) (1979) 215–239
4. Stephenson, K.Z.M.: Rethinking centrality: Methods and examples. Social Networks (1989) 1–37
5. Wasserman, S., Faust, K.: Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences). Cambridge University Press (November 1994)
6. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1) (March 1953) 39–43

7. Everett, M., Borgatti, S.P.: Ego network betweenness. Social Networks **27**(1) (January 2005) 31–38
8. Brandes, U.: A faster algorithm for betweenness centrality. Journal of Mathematical Sociology **25**(163) (2001)
9. Freeman, L.C.: A set of measures of centrality based upon betweenness. Sociometry **40** (1977) 35–41
10. Friedkin, N.E.: Theoretical foundations for centrality measures. American Journal of Sociology **96**(6) (1991) 1478–1504
11. Gert, S.: The centrality index of a graph. Psychometrika **31**(4) (December 1966) 581–603
12. Bonacich, P.: Factoring and weighting approaches to status scores and clique identification. Journal of Mathematical Sociology **2**(1) (1972) 113–120
13. Lawrence, P., Sergey, B., Rajeev, M., Terry, W.: The pagerank citation ranking: Bringing order to the web. Stanford Digital Library Technologies Project (1998)
14. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems **30** (1998) 107–117
15. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM **46** (1999) 604–632
16. Li, L., Shang, Y., Zhang, W.: Improvement of hits-based algorithms on web documents. In: WWW '02: Proceedings of the 11th international conference on World Wide Web, ACM Press (2002) 527–535
17. Haveliwala, T.H.: Topic-sensitive pagerank. In: WWW '02: Proceedings of the 11th international conference on World Wide Web, New York, NY, USA, ACM (2002) 517–526
18. Birnbaum, Z.W.: On the importance of different components in a multicomponent system. In: Multivariate Analysis - II. (1969) 581–592
19. Hong, J., Lie, C.: Joint reliability-importance of two edges in an undirected network. IEEE Transactions on Reliability **42** (1993) 17–23,33
20. Fjällström, P.O.: Algorithms for graph partitioning: A Survey. In: Linköping Electronic Atricles in Computer and Information Science, 3. (1998)
21. Elsner, U.: Graph partitioning - a survey. Technical Report SFB393/97-27, Technische Universität Chemnitz (1997)
22. Pothen, A., Simon, H.D., Liu, K.P.: Partitioning sparse matrices with eigenvectors of graphs. SIAM J. Matrix Anal. Appl. **11**(3) (July 1990) 430–452
23. Hendrickson, B., Leland, R.: An improved spectral graph partitioning algorithm for mapping parallel computations. SIAM Journal on Scientific Computing **16**(2) (1995) 452–469
24. Miller, G.L., Teng, S.H., Thurston, W., Vavasis, S.A.: Geometric separators for finite-element meshes. SIAM J. Sci. Comput. **19**(2) (1998) 364–386
25. Berger, M.J., Bokhari, S.H.: A partitioning strategy for nonuniform problems on multiprocessors. IEEE Transactions on Computers **36**(5) (1987) 570–580
26. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing **20** (1998) 359–392
27. Hendrickson, B., Leland, R.: A multi-level algorithm for partitioning graphs. In: Supercomputing. (1995)
28. Newman, M.E.J.: Detecting community structure in networks. The European Physical Journal B-Condensed Matter **38**(2) (2004) 321–330
29. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. The Bell system technical journal **49**(1) (1970) 291–307

30. Fiduccia, C.M., Mattheyses, R.M.: A linear-time heuristic for improving network partitions. In: DAC '82: Proceedings of the 19th conference on Design automation, Piscataway, NJ, USA, IEEE Press (1982) 175–181
31. Diekmann, R., Monien, B., Preis, R.: Using helpful sets to improve graph bisections. In: Interconnection Networks and Mapping and Scheduling Parallel Computations. (1995) 57–73
32. Scott, J.: Social Network Analysis: A Handbook. SAGE Publications (January 2000)
33. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E **69** (2004) 026113
34. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc Natl Acad Sci U S A **99**(12) (June 2002) 7821–7826
35. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks (Feb 2004)
36. Wu, F., Huberman, B.A.: Finding communities in linear time: A physics approach (2003)
37. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Physical Review E **70** (2004) 066111
38. Borgatti, S.P.: Identifying sets of key players in a social network. Computational and Mathematical Organization Theory **12**(1) (April 2006) 21–34
39. Dehaspe, L., Toivonen, H., King, R.D.: Finding frequent substructures in chemical compounds. In Agrawal, R., Stolorz, P., Piatetsky-Shapiro, G., eds.: 4th International Conference on Knowledge Discovery and Data Mining, AAAI Press. (August 1998) 30–36
40. Holder, L.B., Cook, D.J., Djoko, S.: Substructure discovery in the subdue system. In: Proceedings of the AAAI Workshop on Knowledge Discovery in Databases. (1994) 169–180
41. Cook, D.J., Holder, L.B.: Substructure discovery using minimum description length and background knowledge. Journal of Artificial Intelligence Research **1** (1994) 231–255
42. Yoshida, K., Motoda, H.: Clip: Concept learning from inference patterns. Artif. Intell. **75**(1) (1995) 63–92
43. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In Bocca, J.B., Jarke, M., Zaniolo, C., eds.: Proc. 20th Int. Conf. Very Large Data Bases, VLDB, Morgan Kaufmann (September 1994) 487–499
44. Inokuchi, A., Washio, T., Motoda, H.: An apriori-based algorithm for mining frequent substructures from graph data. In: PKDD '00: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, London, UK, Springer-Verlag (2000) 13–23
45. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. Data Mining, IEEE International Conference on **0** (2001) 313
46. Kuramochi, M., Karypis, G.: An efficient algorithm for discovering frequent subgraphs. IEEE Trans. on Knowl. and Data Eng. **16**(9) (2004) 1038–1051
47. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), Washington, DC, USA, IEEE Computer Society (2002) 721
48. Yan, X., Han, J.: Closegraph: mining closed frequent graph patterns. In: KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2003) 286–295

49. Huan, J., Wang, W., Prins, J., Yang, J.: Spin: mining maximal frequent subgraphs from graph databases. In: KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2004) 581–586
50. Bringmann, B., Nijssen, S.: What is frequent in a single graph? In: Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference (PAKDD). (2008) 858–863
51. Fiedler, M., Borgelt, C.: Subgraph support in a single large graph. In: ICDM '07: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, Washington, DC, USA, IEEE Computer Society (2007) 399–404
52. Fiedler, M., Borgelt, C.: Support computation for mining frequent subgraphs in a single graph. In: Proc. 5th Int. Workshop on Mining and Learning with Graphs (MLG 2007, Florence, Italy), Florence, Italy (2007) 25–30
53. Kuramochi, M., Karypis, G.: Finding frequent patterns in a large sparse graph. Data Mining and Knowledge Discovery **11**(3) (2005) 243–271
54. Grötschel, M., Monma, C.L., Stoer, M.: Design of survivable networks. In: Handbooks in Operations Research and Management Science. (1993)
55. Ramakrishnan, C., Milnor, W.H., Perry, M., Sheth, A.P.: Discovering informative connection subgraphs in multi-relational graphs. SIGKDD Explor. Newsl. **7**(2) (2005) 56–63
56. Faloutsos, C., McCurley, K.S., Tomkins, A.: Fast discovery of connection subgraphs. In: KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2004) 118–127
57. Tong, H., Faloutsos, C.: Center-piece subgraphs: problem definition and fast solutions. In: KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2006) 404–413
58. Sevon, P., Eronen, L., Hintsanen, P., Kulovesi, K., Toivonen, H.: Link discovery in graphs derived from biologican databases. In Leser, U., Naumann, F., Eckmann, B., eds.: 3rd International Workshop on Data Integration in the Life Sciences 2006 (DILS'06). Volume LNBI 4705., Berlin/Heidelberg, Germany, Springer–Verlag (2006) 35–49
59. Hintsanen, P., Toivonen, H.: Finding reliable subgraphs from large probabilistic graphs. In: ECML PKDD '08: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I, Berlin, Heidelberg, Springer-Verlag (2008) 15–15
60. Raedt, L.D., Kimmig, A., Toivonen, H.: Problog: a probabilistic prolog and its application in link discovery. In: Proceedings of 20th International Joint Conference on Artificial Intelligence, AAAI Press (2007) 2468–2473
61. Raedt, L., Kersting, K., Kimmig, A., Revoredo, K., Toivonen, H.: Compressing probabilistic prolog programs. Mach. Learn. **70**(2-3) (2008) 151–168
62. Lin, S., Chalupsky, H.: Unsupervised link discovery in multi-relational data via rarity analysis. In: ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining, Washington, DC, USA, IEEE Computer Society (2003) 171
63. White, S., Smyth, P.: Algorithms for estimating relative importance in networks. In: KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2003) 266–275

64. Jeh, G., Widom, J.: Scaling personalized web search. In: WWW '03: Proceedings of the 12th international conference on World Wide Web, New York, NY, USA, ACM (2003) 271–279
65. Forgaras, D., Rácz, B., Csalogány, K., Sarlós, T.: Towards scaling fully personalized pagerank: algorithms, lower bounds and experiments. Internet Mathematics **2**(3) (2005) 335–358
66. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Eighteenth International Conference on Machine Learning (ICML 2001), Williamstown, MA, USA, Morgan Kaufmann (2001) 577–584
67. Eick, C.F., Zeidat, N.M., Zhao, Z.: Supervised clustering – algorithms and benefits. In: IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), Boca Raton, FL, USA, IEEE Computer Society (2004) 774–776
68. Ullmann, J.R.: An algorithm for subgraph isomorphism. J. ACM **23**(1) (1976) 31–42
69. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: A (sub)graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(10) (2004) 1367–1372
70. Shasha, D., Wang, J.T.L., Giugno, R.: Algorithmics and applications of tree and graph searching. In: PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, New York, NY, USA, ACM (2002) 39–52
71. Yan, X., Yu, P.S., Han, J.: Graph indexing: a frequent structure-based approach. In: SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (2004) 335–346
72. He, H., Singh, A.K.: Closure-tree: An index structure for graph queries. In: ICDE '06: Proceedings of the 22nd International Conference on Data Engineering, IEEE Computer Society (2006)
73. Tian, Y., Mceachin, R.C., Santos, C., States, D.J., Patel, J.M.: Saga: a subgraph matching tool for biological graphs. Bioinformatics **23**(2) (2007) 232–239
74. Bunke, H., Shearer, K.: A graph distance metric based on the maximal common subgraph. Pattern Recogn. Lett. **19**(3-4) (1998) 255–259
75. Fernández, M.L., Valiente, G.: A graph distance metric combining maximum common subgraph and minimum common supergraph. Pattern Recogn. Lett. **22**(6-7) (2001) 753–758
76. Yan, X., Yu, P.S., Han, J.: Substructure similarity search in graph databases. In: SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (2005) 766–777
77. Yan, X., Zhu, F., Han, J., Yu, P.S.: Searching substructures with superimposed distance. In: ICDE '06: Proceedings of the 22nd International Conference on Data Engineering, Washington, DC, USA, IEEE Computer Society (2006)
78. Tian, Y., Patel, J.M.: Tale: A tool for approximate large graph matching. In: Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on. (2008) 963–972
79. Williams, D., Huan, J., Wang, W.: Graph database indexing using structured graph decomposition. In: Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on. (April 2007) 976–985

Probabilistic and Logical Inference for Network Mining

Luc de Raedt

Katholieke Universiteit Leuven, Belgium
luc.deraedt@cs.kuleuven.be

Abstract. In this talk I shall analyze network mining and bisociation from a logical and probabilistic inference point of view. This is inspired by the work on ProbLog for link mining [De Raedt, Kimmig, Toivonen, IJCAI 2007], which is - in turn -based on Biomine [Sevonen et al. DILS 06]. The talk shall introduce a probabilistic semantics for networks and databases and use it to clarify notions of deduction, abduction and explanations, induction, analogy, abstraction and spread of influence. All notions will be illustrated in the context of the Biomine network.

Finding representative nodes in probabilistic graphs

Laura Langohr and Hannu Toivonen

Department of Computer Science and
Helsinki Institute for Information Technology HIIT,
University of Helsinki, Finland
laura.langohr, hannu.toivonen@cs.helsinki.fi

Abstract. We introduce the problem of identifying representative nodes in probabilistic graphs, motivated by the need to produce different simple views to large networks. We define a probabilistic similarity measure for nodes, and then apply clustering methods to find groups of nodes. Finally, a representative is output from each cluster. We report on experiments with real biomedical data, using both the k-medoids and hierarchical clustering methods in the clustering step. The results suggest that the clustering based approaches are capable of finding a representative set of nodes.

1 Introduction

Information contained in large networks is difficult to view and handle by users. The problem is obvious for networks of hundreds of nodes, but the problems start already with dozens of nodes.

In this paper, we propose identification of a few representative nodes as one approach to help users make sense of large networks. As an example scenario of the approach, consider link discovery. Given a large number of predicted links, it would be useful to present only a small number of representative ones to the user. Or, representatives could be used to abstract a large set of nodes, e.g., all nodes fulfilling some user-specified criteria of relevance, into a smaller but representative sample.

Our motivation for this problem comes from genetics, where current high-throughput techniques allow simultaneous analysis of very large sets of genes or proteins. Often, these wet lab techniques identify numerous genes (or proteins, or something else) as potentially interesting, e.g., by the statistical significance of their expression, or association with a phenotype (e.g., disease). Finding representative genes among the potentially interesting ones would be useful in several ways. First, it can be used to remove redundancy, when several genes are closely related and showing all of them adds no value. Second, representatives might be helpful in identifying complementary or alternative components in biological mechanisms.

The network in our application is Biomine [1], an integrated network database currently consisting of about 1 million biological concepts and about 10 million links between them. Concepts include genes, proteins, biological processes, cellular components, molecular functions, phenotypes, articles, etc.; weighted links mostly describe their known relationships. The data originates from well known public databases such as Entrez[1], GO[2], and OMIM[3].

The problem thus is to identify few representative nodes among a set of them, in a given weighted network. The solutions proposed in this paper are based on defining a probabilistic similarity measure for nodes, then using clustering to group nodes, and finally selecting a representative from each cluster.

In this framework, two design decisions need to be made: how to measure similarities or distances of nodes in a probabilistic network (Section 3), and which clustering method to use on the nodes (Section 4). Experimental results with real datasets are reported in Section 5, and we conclude in Section 6 with some notes about the results and future work.

2 Related work

Representatives are used to reduce the number of objects in different applications. In an opposite direction to our work, clustering can be approximated by finding representative objects, clustering them, and assigning the remaining objects to the clusters of their representatives. Yan et al. [2] use k-means or RP trees to find representative points, Kaufman and Rousseeuw [3] k-medoids, and Ester et al. [4] the most central object of a data page.

Representatives are also used to reduce the number of datapoints in large databases, i.e., to eliminate irrelevant and redundant examples in databases to be tested by data mining algorithms. Riquelme et al. [5] use ordered projections to find representative patterns, Rozsypal and Kubat [6] genetic algorithms, and Pan et al. [7] measure the representativeness of a set with mutual information and relative entropy.

DeLucia and Obraczaka [8] as well as Liang et al. [9] use representative receivers to limit receiver feedback. Only representatives provide feedback and suppress feedback from the other group members. Representatives are found by utilizing positive and negative acknowledgments in such a way that each congested subtree is represented by one representative.

The cluster approximation and example reduction methods use clustering algorithms to find representatives, but are not applied on graphs. The feedback limitation methods again use graph structures, but not clustering to find representatives. Other applications like viral marketing [10], center-piece subgraphs [11], or PageRank [12] search for special node(s) in graphs, but not for representative nodes. The authors are not aware of approaches to find representatives by clustering nodes and utilizing the graph structure.

[1] www.ncbi.nlm.nih.gov/Entrez/

[2] www.geneontology.org/

[3] www.ncbi.nlm.nih.gov/omim/

3 Similarities in probabilistic graphs

Probabilistic graphs offer a simple yet powerful framework for modeling relationships in weighted networks. A probabilistic graph is simply a weighted graph $G = (V, E)$ where the weight associated with an edge $e \in E$ is probability $p(e)$ (or can be transformed to a probability). The interpretation is that edge e exists with probability $p(e)$, and conversely e does not exist, or is not true, with probability $1 - p(e)$. Edges are assumed mutually independent.

The probabilistic interpretation of edge weights $p(e)$ gives natural measures for indirect relationships between nodes. In this paper we call these similarity measures, as is conventional in the context of clustering.

Probability of a path Given a path P consisting of edges $e_1, \ldots, e_k$, the probability $p(P)$ of the path is the product $p(e_1) \cdot \ldots \cdot p(e_k)$. This corresponds to the probability that the path exists, i.e., that all of its edges exist.

Probability of the best path Given two nodes $u, v \in V$, a measure of their connectedness or similarity is the probability of the best path connecting them:

$$s(u, v) = \max_{P \text{ is a path from } u \text{ to } v} p(P).$$

Obviously, this is not necessarily the path with the least number of edges. This similarity function $s(\cdot)$ is our choice for finding representatives.

Network reliability Given two nodes s and t, an alternative measure of their connectivity is the probability that there exists at least one path (not necessarily the best one) between s and t. This measure is known as the (two-terminal) network reliability (see, e.g., [13]). A classical application of reliability is in communication networks, where each communication link (edge) may fail with some probability. The reliability then gives the probability that s and t can reach each other in the network.

Network reliability is potentially a more powerful measure of connectedness than the probability of the best path, since reliability uses more information — not only the best path. The reliability measure considers alternative paths between s and t as independent evidence for their connectivity, and in effect rewards for such parallelism, while penalizing long paths. The reliability is always at least as high as the probability of the best path, but can also be considerably higher.

However, computing the two-terminal network reliability has been shown to be NP-hard [14]. Fortunately, the probability can be estimated, for instance, by using a straightforward Monte Carlo approach: generate a large number of realizations of the random graph and count the relative frequency of graphs where a path from s to t exists. For very large graphs, we would first extract a smaller neighborhood of s and t, and perform the computation there. More information about our techniques can be found, e.g., in [1, 15]. Due to the complexity of computing the network reliability, we stick to the simpler definition of similarity $s(\cdot)$ as the probability of the best path.

4 Clustering and representatives in graphs

Our approach to finding representatives in networks is to cluster the given nodes, using the similarity measure defined above, and then select one representative from each cluster (Algorithm 1). The aim is to have representatives that are similar to the nodes they represent (i.e., to other members of the cluster), and also to have diverse representatives (from different clusters). In clustering, we experiment with two methods: k-medoids and hierarchical clustering. Both are well-known and widely used methods which can be applied to our problem of finding representatives; k-medoids is an obvious choice, since it directly produces representatives.

Algorithm 1 Find representative nodes

```
Input: Set $S$ of nodes, graph $G$, number $k$ of representatives
Output: $k$ representative nodes from $S$
Find $k$ clusters of nodes in $S$ using similarities $s(\cdot)$ in graph $G$
For each of the $k$ clusters, output its most central node (the node with the maximum
  similarity to other nodes in the cluster)
```

k-medoids k-medoids is similar to the better known k-means method, but better suited for clustering nodes in a graph. Given k, the number of clusters to be constructed, the k-medoids method iteratively chooses cluster centers (medoids) and assigns all nodes to the cluster identified by the nearest medoid. The difference to the k-means clustering method is that instead of using the mean value of the objects within a cluster as cluster center, k-medoids uses the best object as a cluster center. This is a practical necessity when working with graphs, since there is no well defined mean for a set of nodes. The k-medoids method also immediately gives the representatives. See, e.g., [16, 3] for more information about the methods.

For very large graphs, a straight forward implementation of k-medoids is not necessarily the most efficient. In our applications we use the Biomine database and tools to facilitate faster clustering. Given a set S of nodes, i.e., biological entities, to be clustered, and k, the number of clusters to be constructed, the method proceeds as follows. First, the Biomine system is queried for a graph G of at most 1000 nodes cross-connecting nodes in S as strongly as possible. The pairwise similarities between nodes are then calculated as the best path probabilities in G.

The Biomine system uses a heuristic to obtain G, details are omitted here. As the Biomine network consists of a million nodes, querying it for a graph exceeds by far the computational complexity of running k-medoids on the extracted graph. For brevity, we here omit discussion of the computational complexities of k-medoids and other approaches.

To start the actual clustering, k nodes from S are chosen randomly as initial medoids. Each remaining node in S is then clustered to the most similar medoid. If the pairwise similarity between a node and all medoids equals zero, the node will be considered an outlier and is not assigned to any medoid in this iteration. Then, a new medoid is calculated for each cluster. The node that has a maximal product of similarities between each other node in the cluster and itself is chosen as the new medoid. The last two steps are then repeated until the clustering converges or the maximum number of iterations is reached.

Example As an example, k-medoids was run with $k = 3$ and a set of nine genes. The genes belong to three known groups, each group of three genes being associated to the same phenotype. The three OMIM phenotypes used in the example are a pigmentation phenotype (MIM:227220), lactase persistence (MIM:223100), and Alzheimer disease (MIM: 104300).

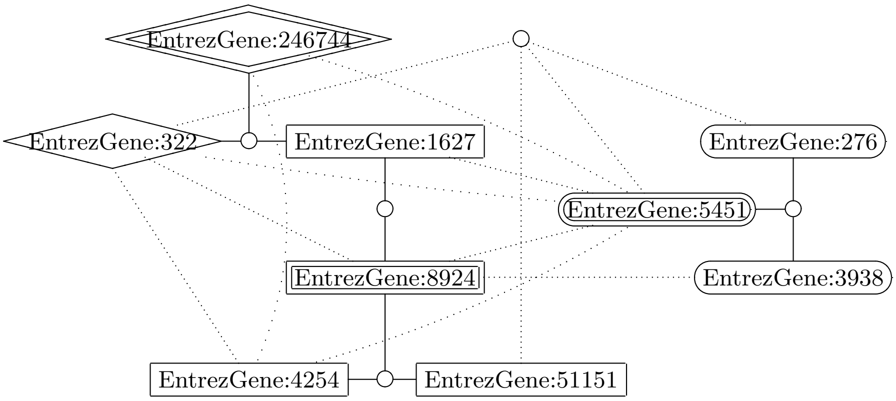


Fig. 1: Clusters (diamonds, boxes, ellipses) and representatives (double borders) of nine given nodes, and some connecting nodes (circles) on best paths between them. Lines represent edges between two nodes, dotted lines represent best paths with several nodes.

The algorithm converged in this case after two iterations. The result of the example run is shown in Figure 1. Looking at the quality of clustering, only one gene (EntrezGene:1627) was assigned to another cluster than it should with respect to the OMIM phenotypes. Apart from this gene, the clustering produced the expected partitioning: each gene was assigned to a cluster close to its corresponding phenotype. The three representatives (medoids) are genes assigned to different phenotypes. Hence, the medoids can be considered representative for the nine genes.

Hierarchical clustering As an alternative clustering method we use hierarchical clustering. (Again, see, e.g., [16, 3] for more information.) A possible problem with the k-medoids approach is that it may discover star-shaped clusters, where cluster members are connected mainly through the medoid. To give more weight on cluster coherence, we use the average linkage method, as follows.

In the practical implementation, we again start by querying the Biomine system for a graph G of at most 1000 nodes connecting the given nodes S, and compute similarities of nodes in S as the probabilities of the best paths connecting them in G.

The hierarchical clustering proceeds in the standard, iterative manner, starting with having each node in a cluster of its own. In each iteration, those two clusters are merged that give the best merged cluster as a result, measured by the average similarity of nodes in the merged cluster. The clustering is finished when exactly k clusters remain.

After the clusters have been identified, we find the medoid in each cluster (as in the k-medoids method) and output them as representatives.

Random selection of representatives For experimental evaluation, we also consider a method that selecst representatives randomly. We again query the Biomine system for a graph G of at most 1000 nodes connecting the given nodes S, and compute similarities of nodes in S as the probabilities of the best paths connecting them in G.

We randomly select k medoids and cluster the remaining nodes of S to the most similar medoid. If the pairwise similarity between a node and all medoids equals zero, the node will be considered an outlier, as in k-medoids.

5 Experiments

Our goal in this section is to evaluate how successful the method is in finding representative nodes.

5.1 Test setting

Test data We used data published by Köhler et al. [17], who defined 110 disease-gene families based on the OMIM database. The families contain three to 41 genes each; each family is related to one disease. Köhler et al. originally used the families in their experiments on candidate gene prioritization. Given a list of candidate genes they used a protein-protein interaction network to score the given genes by distance to all genes that are known to be related to a particular disease. Then they set up a ranking of the candidate genes based on their scores. Although their aim was different from ours, and the network they used was only a protein interaction network, the data sets give a natural real test case for our problem, too.

Test setting In each test run, k gene families were randomly chosen as the nodes to find k representatives for. We performed 100 test runs for $k = 3$ and $k = 10$ of all three variants (k-medoids, hierarchical, random) of the method, and report averages over the 100 runs. As k-medoids is sensitive to the randomly selected first medoids, we applied k-medoids five times in each run and selected the best result. We applied the random selection of representatives 20 times in each run and used average values of the measures in order to compensate the random variation.

Measures of representativeness We use two measures of representativeness of the selected nodes. The first one is based on the similarity of nodes to their representatives, the second one on how well the k (known) families of nodes are covered by the k representatives.

The first measure is directly related to the objective of the k-medoids method. The idea is that each node is represented by its nearest representative, and we simply measure the average similarity of objects to their closest representative (ASR):

$$ASR = \frac{1}{|S| - K} \sum_{x \in S, x \neq m(x)} s(x, m(x))$$

where S is the set of given vertices, K is the number of clusters, $m(x)$ is the medoid most similar to x, and $s(x, m(x))$ denotes the similarity (probability of best path) between node x and medoid $m(x)$.

The second measure takes advantage of the known families of genes in our test setting. The rational here is that a representation is better if it covers more families, i.e., contains a representative in more families. For this purpose, we calculate the fraction of non-represented classes (NRC):

$$NRC = \frac{1}{K} |\{k \mid \nexists j : m_j \in H_k,\ j = 1..K\}|,$$

where K is the number of classes and clusters (equal in our current test setting), m_j is the medoid of the jth cluster, and H_k is the kth original class.

For the k-medoids variant, we also report the number of outliers. Recall that the method outputs as outliers those nodes that are not connected (in the extracted subnetwork) to any medoid.

As additional characteristics of the methods we measure how good the underlying clusterings are. Again, we have two measures, one for the compactness of clusters, and one based on the known classification.

The first additional measure is the average compactness of clusters (ACC), where the compactness of a given cluster is defined as the minimum similarity of two objects in the cluster. The average is computed over clusters having at least two members:

$$ACC = \frac{1}{k'} \sum_{k=1}^{K} \min_{x,y \in C_k} s(x, y), \text{ where } k' = |\{k \mid |C_k| > 1, k = 1..K\}|,$$

i.e., k' is the number on non-trivial clusters. This measure is sensitive to outliers, and thus may favor the k-medoids variant.

The second additional measure compares the clustering to the known classes and measures their difference. We first identify the class best represented by each cluster, and then calculate how many objects were "wrongly assigned" (WAO):

$$WAO = \frac{1}{|S|} \sum_{k=1}^{K} \min_{k'=1..K} |C_k \setminus H_{k'}|.$$

Rand index could have been used here just as well.

5.2 Results

In terms of average similarity of nodes to their representative (ASR), the k-medoids method slightly but clearly outperforms the hierarchical method (Figure 2, left panels). The hierarchical method, in turn, is clearly superior to the random selection of representatives (Figure 2, right panels). For the k-medoids variant and $k = 3$, average similarities in the 100 test runs range from 0.3 to 0.8, and the total average is 0.51. For $k = 10$ the average is 0.55 and range is 0.4 to 0.8. For the hierarchical variant and $k = 3$, the average is 0.48 and range is 0.1 to 0.8. For $k = 10$ the average is 0.51 and range is 0.3 to 0.7. For the random variant and $k = 3$, average is 0.36 and range is 0.2 to 0.7. For $k = 10$ average is 0.43 and range is 0.3 to 0.6. These differences are no big surprise, since the k-medoids method more directly aims to maximize this measure than the hierarchical method, which however performs better than random choice of representatives. Further, the k-medoids method may output some nodes as outliers. The average fraction of outliers in the experiments was 1.9 % for $k = 3$ and 4.5 % for $k = 10$.

The fraction of non-represented classes is a more neutral measure of performance since neither variant directly maximizes this. The results indicate that the k-medoids variant is slightly better with respect to this measure for $k = 3$ (Table 1), but for $k = 10$ the hierarchical variant is clearly superior. Both methods clearly outperform the random selection of representatives.

To gain a better understanding of the performance of the methods, we look at the quality of clusterings produced. It is not surprising that clusters produced by the hierarchical method are on average more compact than those produced by the k-medoids method (Figure 3), as the hierarchical method more directly optimizes this measure. It is however somewhat surprising that k-medoids performs only slightly better than the random variant. The average compactness (minimum similarity within a cluster) is 0.20 ($k = 3$) and 0.23 ($k = 10$) for k-medoids, 0.33 ($k = 3$) and 0.48 ($k = 10$) for the hierarchical variant, and 0.16 ($k = 3$) and 0.21 ($k = 10$) for the random variant, with considerable spread and variance in all results.

In terms of wrongly assigned objects, the hierarchical variant clearly outperforms k-medoids (Table 2). The k-medoids variant outperforms the random selection of representatives, but for $k = 10$ only by a small difference.

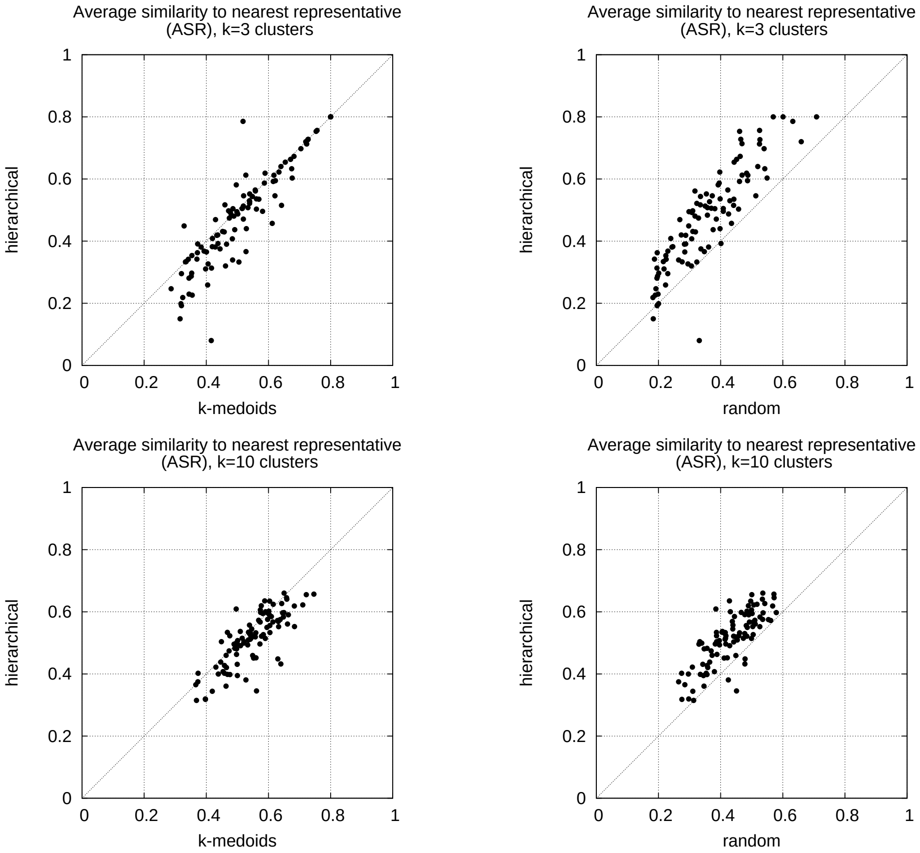


Fig. 2: Average similarity of objects to their nearest representative (ASR). In each panel 100 runs are visualized. Each point represents one run, thereby comparing ASR values of two variants (see x- and y-axis).

	k=3	k=10
k-medoids	14 %	29 %
hierarchical	16 %	21 %
random	34 %	39 %

Table 1: Fraction of non-represented classes (NRC).

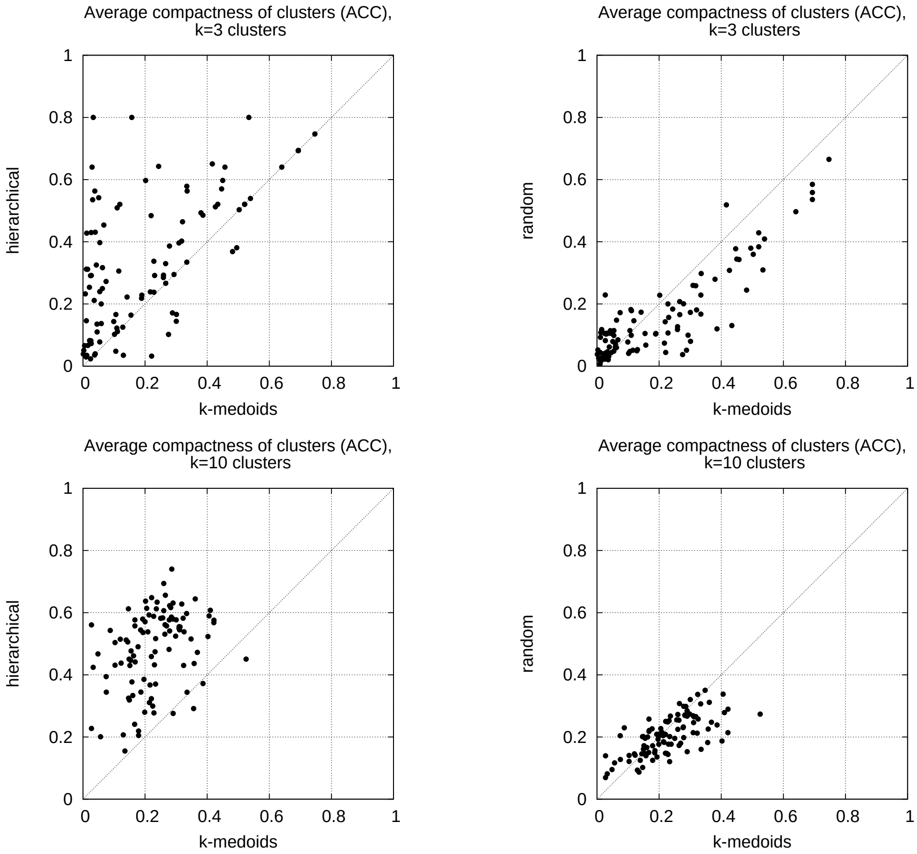


Fig. 3: Average compactness of nontrivial clusters (ACC). In each panel 100 runs are visualized. Each point represents one run, thereby comparing ACC values of two variants (see x- and y-axis).

	k=3	k=10
k-medoids	18 %	44 %
hierarchical	15 %	25 %
random	27 %	46 %

Table 2: Wrongly assigned objects (WAO).

6 Conclusions

We have described the problem of finding representative nodes in large probabilistic graphs. We based our definition of node similarity on a simple probabilistic interpretation of edge weights. We then gave a clustering-based method for identifying representatives, with two variants: one based on the k-medoids methods, one on the hierarchical clustering approach.

We performed a series of 100 experiments on a real biomedical data, using published gene families [17] and the integrated Biomine network [1]. We measured the success of finding representatives with two measures: the similarity of nodes to their representatives, and the fraction of classes represented by the output.

In our experimental comparison, the k-medoids based variant and the hierarchical method are promising approaches. A look at the quality of the clusterings indicates that the success of the methods in identifying the underlying clusters depends on the measure used, and may also depend on the number of clusters to be constructed. According to the results, the hierarchical method is more robust, especially when looking for more than just couple of representatives.

More work is needed to understand the reasons for the differences of the two approaches. Further, the problem of finding representative nodes needs to be validated in real applications. Based on the simple methods introduced here, and the initial experimental results, the clustering approach seems to be capable of reliably identifying a high quality set of representatives.

Acknowledgements We would like to thank Lauri Eronen for his help with the test data and hierarchical clustering.

This work has been supported by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland and by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898.

References

1. Sevon, P., Eronen, L., Hintsanen, P., Kulovesi, K., Toivonen, H.: Link discovery in graphs derived from biological databases. In: 3rd International Workshop on Data Integration in the Life Sciences 2006 (DILS'06), Hinxton, UK (2006) 35–49
2. Yan, D., Huang, L., Jordan, M.I.: Fast approximate spectral clustering. In: 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09), New York, NY, USA, ACM (2009) 907–916
3. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley Inc., New York, NY (1990)
4. Ester, M., Kriegel, H.P., Xu, X.: Knowledge discovery in large spatial databases: focusing techniques for efficient class identification. In: 4th International Symposium on Advances in Spatial Databases (SDD'95), London, UK, Springer-Verlag (1995) 67–82
5. Riquelme, J.C., Aguilar-Ruiz, J.S., Toro, M.: Finding representative patterns with ordered projections. Pattern Recognition **36**(4) (2003) 1009–1018

6. Rozsypal, A., Kubat, M.: Selecting representative examples and attributes by a genetic algorithm. Intelligent Data Analysis **7**(4) (2003) 291–304
7. Pan, F., Wang, W., Tung, A.K.H., Yang, J.: Finding representative set from massive data. In: The 5th IEEE International Conference on Data Mining (ICDM'05), Washington, DC, USA, IEEE Computer Society (2005) 338–345
8. DeLucia, D., Obraczka, K.: Multicast feedback suppression using representatives. In: 16th Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution (INFOCOM'97), Washington, DC, USA, IEEE Computer Society (1997) 463–470
9. Liang, C., Hock, N.C., Liren, Z.: Selection of representatives for feedback suppression in reliable multicast protocols. Electronics Letters **37**(1) (2001) 23–25
10. Domingos, P., Richardson, M.: Mining the network value of customers. In: 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01), New York, NY, USA, ACM (2001) 57–66
11. Tong, H., Faloutsos, C.: Center-piece subgraphs: problem definition and fast solutions. In: 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'06), New York, NY, USA, ACM (2006) 404–413
12. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1999)
13. Colbourn, C.J.: The Combinatiorics of Network Reliability. Oxford University Press (1987)
14. Valiant, L.: The complexity of enumeration and reliability problems. SIAM Journal on Computing **8** (1979) 410 – 421
15. Hintsanen, P., Toivonen, H.: Finding reliable subgraphs from large probabilistic graphs. Data Mining and Knowledge Discovery **17**(1) (2008) 3 – 23
16. Han, J., Kamber, M.: Data Mining. Concepts and Techniques. 2nd edn. Morgan Kaufmann (2006)
17. Köhler, S., Bauer, S., Horn, D., Robinson, P.: Walking the interactome for prioritization of candidate disease genes. American Journal of Human Genetics **82**(4) (April 2008) 949 – 958

Pure Spreading Activation is Pointless

Michael R. Berthold, Ulrik Brandes, Tobias Kötter,
Martin Mader, Uwe Nagel, Kilian Thiel

Department of Computer and Information Science,
University of Konstanz, 78457 Konstanz, Germany
firstname.lastname@uni-konstanz.de

Abstract. Spreading activation is a popular technique for retrieving and ranking indirectly related information by activating query items and spreading their activation along relatedness links. Almost every use of the technique is accompanied by its own set of restrictions on the dynamics, though, and the usual motivation is a reduced computational demand or an improved fit to specific types of data. We show that in linear, constraint-free scenarios spreading activation would actually yield query-independent results, so that applications crucially depend on the imposed restrictions. To avoid this undesirable behavior, we study natural modifications that ensure query-dependent results even without heuristic restrictions and provide experimental evidence for their effectiveness.

1 Introduction

Spreading activation methods have been introduced first by Quillian and Collins [1, 2] to query networks. These methods allow relevant subgraphs, nodes, or edges according to a given query to be extracted. For this purpose specific nodes of a network are activated and the activation is spread iteratively to adjacent nodes until the process is terminated. The result is determined from the subset of activated nodes, their activation level, and induced subgraph.

Initially spreading activation techniques have been applied to semantic networks, and later on in fields of Information Retrieval (IR) as well [3–5], originating from past work in associative retrieval [6] based on associative networks. The fundamental idea of associative retrieval is that related information is connected in the network. It is assumed that therefore relevant information can be retrieved by considering the associations to concepts known to be relevant or specified by the user. In the fields of IR the associated units of information are usually documents, parts of documents, concepts, index terms, keywords, or authors. Spreading activation is used to retrieve, for example, relevant documents, authors or terms related to a given query. Besides IR, spreading activation methods have been applied to other areas, such as trust propagation [7], ontology extension [8], word sense disambiguation [9], or recommender systems [10].

To our knowledge, all these methods share the usage of constraints to control the spread of activation inside a network, such as distance constraints to terminate the spreading procedure after a certain number of iterations or fan-out constraints to affect the path of spreading. We show that, without these constraints, spreading activation models converge to specific fixed points. These convergence points are usually independent of the initial query. This behavior turns pure (constraint free) spreading activation into an inadequate technique to answer queries. This drawback can be overcome by applying constraints or other methods, that induce convergence points depending on the initial query. We show that the cumulation of single iteration results of spreading activation, both converges and relates the fixed point to the query. For this orthogonal approach to constraints, three different cumulation strategies are compared, which turn out to be very similar when the iteration process continues infinitely. Additionally the behavior of convergence for normalized systems is shown and an outlook to non-linear systems is given.

The document is organized as follows: The next section defines the underlying framework of spreading activation. It describes and formalizes the basic functionality all spreading activation methods have in common, the basic constraints and drawbacks, as well as a matrix and vector representation of the spreading activation procedure. Based on this representation we show in Section 3 the convergence of pure spreading activation to a fixed point, which makes it inadequate to answer queries. A solution to overcome this drawback is the accumulation of iteration results, described in Section 4. In Section 4.2 the accumulation and the convergence behavior of normalized systems is examined. We complete this section with a short inspection of convergence in non-linear systems. In Section 5 we present two application scenarios of spreading activation, a simple query expansion and an iterative cosine similarity method based on a term-document network. We empirically demonstrate the behavior of convergence for these application scenarios in Section 6. For this purpose we used a network created from parts of the SMART document collection. Section 7 concludes the document.

2 Preliminaries

The functionality of spreading activation is related to neural networks. Both methods have in common that units can be activated through their incoming activation, and edges spread the outgoing activation to adjacent units. In [11] eight major aspects of a neural network are defined. Based on these aspects the functionality of spreading activation can be specified in terms of three components described below.

2.1 Framework

The activation is spread on a graph $G = (V, E, w)$, with weights $w : E \to \mathbb{R}$. For ease of exposition we assume that $V = \{1, \ldots, n\}$ and that G is undirected, but our results easily generalize to directed graphs. We extend w to $V \times V$ by letting $w(u, v) = 0$ if $(u, v) \notin E$. The set of neighbors of $v \in V$ is denoted by $N(v) = \{u : \{u, v\} \in E\}$. The trisection consists of input, activation, and output.

For each part a function and a state exists. The function specifies the transition to a certain state at a time k. The state at time k of all nodes is represented by a vector $\mathbf{a}^{(k)} \in \mathbb{R}^V$, where $\mathbf{a}_v^{(k)}$ is the activation of $v \in V$. The state at time $k > 0$ is obtained from the state at time $k-1$ via the following three families of functions.

- *Input function* $\text{in}_v : \mathbb{R}^n \to \mathbb{R}$. The input function aggregates the incoming activation of a node v and defines the input $\mathbf{i}_v^{(k)} = \text{in}_v(\mathbf{o}^{(k-1)})$ of that node at time k. A prototypical choice is the sum of the weighted outgoing activation of the adjacent nodes

$$\text{in}_v(\mathbf{o}^{(k-1)}) = \sum_{u \in N(v)} \mathbf{o}_u^{(k-1)} w(u, v),$$

  with $\mathbf{o}_u^{(k-1)}$ as the output activation of node u at time $k-1$.
- *Activation function* $\text{act}_v : \mathbb{R} \to \mathbb{R}$. The activation function determines the level of activation $\mathbf{a}_v^{(k)} = \text{act}_v(\mathbf{i}_v^{(k)})$ of a node v at time k and thus whether a node is activated, i.e. whether its activation is spread to adjacent nodes in the next iteration. To introduce non-linearity into the system e.g. sign, threshold, sigmoid functions are used.
- *Output function* $\text{out}_v : \mathbb{R} \to \mathbb{R}$. The output function determines the outgoing activation $\mathbf{o}_v^{(k)} = \text{out}_v(\mathbf{a}_v^{(k)})$ of a node v at time k. Output functions can be used to normalize the output in certain ways. This could have systematic reasons, for example, to ensure that the sum of activation in the whole network is constant, or could be a necessity by design.

Initially some nodes, usually those representing the query, are activated first. The activation spreads across incident edges to adjacent nodes and activates these nodes as well. This process is usually terminated after a certain number of iterations, activated nodes or other abort criteria. The subgraph induced by the activated nodes represents the result of the query. A ranking of the nodes as often required in IR can be obtained by sorting the nodes according to their output activation henceforth simply denoted as activation.

2.2 Constraints

In general not only constraints like termination criteria are used to regulate the networks spread. The most common constraints according to Crestani [12] are listed below:

- *Distance constraint*: the activation decreases based on the distance to the initially activated nodes and finally stops at a certain distance, based on the argument that the strength of the relation decreases with their semantic distance.
- *Fan-out constraint*: the spread of activation expires at nodes with a high out-degree, in order to avoid a too wide spreading.

- *Path constraint*: the activation spreads across preferential paths, reflecting specific inference rules.
- *Activation constraint*: to avoid the activation of all nodes which receive activation at all, a threshold function can be applied. In this case a certain degree of input activation is required to activate the node.

In our examination we do not consider threshold functions as a constraint. The reason is that in contrast to the other constraints the activation constraint is not based on structural properties of the network but solely on the activation level itself.

Crestani [12] argues that constraints are neccessary, because pure (constraint-free) spreading activation has three serious drawbacks:

1. Without control the activation will spread all over the network.
2. Semantics of relations, represented as edge labels can hardly be interpreted and considered.
3. The implementation of an inference procedure based on the semantics of relations (edge labels) is difficult.

In this paper we propose a fourth important drawback, in case of linear activation functions, that is *convergence* to one fixed points which is independent of the query.

2.3 Matrix notation

In linear systems a common activation and output function is the identity function. In this context we now point out how to express the corresponding spreading activation process in matrix notation. This will then be used to examine the convergence behavior.

Given a graph $G = (V, E, w)$ and an initial activation $\mathbf{a}^{(0)}$ we can express a complete activation step by omitting the (identity) activation and output functions as

$$\mathbf{a}_v^{(k)} = \sum_{u \in N(v)} \mathbf{a}_u^{(k-1)} w(u, v).$$

Define the weight matrix $W \in \mathbb{R}^{n \times n}$ with $(W)_{uv} = w(u, v)$ and $w(u, v) = 0$ if $(u, v) \notin E$ then this can be written as

$$\mathbf{a}_v^{(k)} = \sum_{u \in V} \mathbf{a}_u^{(k-1)} (W)_{uv}.$$

Consequently the complete activation vector $\mathbf{a}^{(k)}$ indexed by the nodes of V can be written in matrix notation as $\mathbf{a}^{(k)} = W\mathbf{a}^{(k-1)}$. Recursive substitution yields

$$\mathbf{a}^{(k)} = W^k \mathbf{a}^{(0)}.$$

We will use this notation to examine different variants of constraint-free spreading activation.

3 Query Independence

A simple method of eigenvector calculation is power iteration. The elements of the sequence $\mathbf{e}^{(k)} = \frac{W\mathbf{e}^{(k-1)}}{\|W\mathbf{e}^{(k-1)}\|}$ are computed until the change between two consecutive elements is sufficiently small. Under some conditions on W and $\mathbf{e}^{(0)}$ this series will converge to the principal eigenvector corresponding to the eigenvalue of W having maximum absolute value. The conditions for this convergence can be derived from the Perron-Froebenius Lemma (see e.g. [13]) and basically guarantee the uniqueness of the eigenvalue with maximum absolute value.

In our context it is sufficient to assume that G is connected and not bipartite and $\mathbf{e}^{(0)}$ is not orthogonal to the principal eigenvector. For practical reasons most spreading activation systems have an underlying network which is sufficiently connected and therefore these requirements are usually met. In this case the principal (in terms of absolute value) eigenvalue of W, called spectral radius $\rho(W)$ is positive and has exactly one associated eigenvector with non-negative components.

Consequently after a sufficient number of iterations, the activation vector will approach eigenvector centrality. While this may actually be a reasonable default answer, it is not at all related to the query.

4 Avoiding Query Independence

Since a ranking procedure with a result independent of the initial activation is not useful we will have a look at some approaches to overcome this problem without adding constraints. An initial idea can be derived from Katz' [14] status centrality. For a (directed) graph with adjacency matrix W it is defined as:

$$c_{\text{Katz}}(W) = \sum_{k=1}^{\infty} \left(\alpha W^T\right)^k \mathbf{1},$$

where $\mathbf{1}$ is the vector of all ones and α is a decay factor. For α limited by $0 < \alpha < \rho(W)^{-1}$ this series converges analog to the geometric series and can be written in a closed form as $c_{\text{Katz}}(W) = \left(\left(I - \alpha W^T\right)^{-1} - I\right)\mathbf{1}$. Katz' results where stated for adjacency matrices with entries being either 0 or 1 but they also hold for the more general case of W being a matrix with real numbers as edge weights.

4.1 Iteration with Memory

We will examine three different approaches to combine or modify the activations of the single iterations in order to obtain a final activation significantly depending on the initial activations, that is the query:

1. *Accumulation*: the final activation is defined as the sum of all intermediate activation states.

2. *Activation renewal*: the initial activations are renewed in each step.
3. *Inertia*: activations of the previous state are partially retained.

The main idea for the combinations of the iteration steps as presented here is interpretable as an approach to balance between a local and a global factor. The transition between the local and the global factor corresponds to the transition from direct via indirect node similarities to the principal eigenvector of the system. The accumulation process of the iteration steps is then supposed to achieve a balance between the two factors by an appropriate weighting of the single steps.

1. Accumulation In the first approach the final activation $\mathbf{a}^*$ of nodes relative to the initial activation is determined as a linear combination of the individual iteration steps:

$$\mathbf{a}^* = \sum_{k=0}^{\infty} \lambda(k)\mathbf{a}^{(k)} = \left(\sum_{k=0}^{\infty} \lambda(k)W^k\right)\mathbf{a}^{(0)},$$

where $\lambda(k)$ is a decay function. Generally we use $\lambda(k) = \alpha^k$.

If λ is limited to the form $\lambda(k) = \alpha^k$, constraint-free spreading activation can be seen as a generalization of Katz status with an added bias to a starting vector consisting of initially activated nodes. In this case $\mathbf{a}^*$ can be determined directly by the matrix multiplication

$$\mathbf{a}^* = (I - \alpha W)^{-1}\mathbf{a}^{(0)},$$

as long as $0 < \alpha < \rho(W)^{-1}$.

2. Activation renewal Another variant of spreading activation is obtained by renewing the activation of the initially activated nodes to strengthen their influence:

$$\mathbf{a}^{(k)} = \mathbf{a}^{(0)} + W\mathbf{a}^{(k-1)}.$$

By recursive substitution we can give a closed form for $\mathbf{a}^{(k)}$:

$$\mathbf{a}^{(k)} = \left(\sum_{i=0}^{k} W^i\right)\mathbf{a}^{(0)}.$$

This is identical to the first variant with $\lambda(k) = 1$ therefore convergence can only be guaranteed for graphs G with $\rho(G) < 1$.

3. Inertia The third method we will examine is obtained by partially retaining the previous state:

$$\mathbf{a}^{(k)} = \mathbf{a}^{(k-1)} + W\mathbf{a}^{(k-1)}.$$

This approach can also be expressed by adding self loops to each node with a weight of 1 which can be seen in the respective closed form:

$$\mathbf{a}^{(k)} = (I + W)^k\mathbf{a}^{(0)}.$$

The result is a ranking which corresponds to the principal eigenvector of $(I+W)$. Extending the approach by cumulating the activations of the iterations up to infinity analog to the first variant with $\lambda(k) = \alpha^k$ yields:

$$\mathbf{a}^* = ((1-\alpha)I - \alpha W)^{-1} \mathbf{a}^{(0)}.$$

Again this result only holds if $0 < \alpha < \rho(I+W)^{-1}$. Note further that this modification of the adjacency matrix shifts the spectrum of W by adding 1 to each eigenvalues but results in the same eigenvectors. The spectrum shift may also influence convergence speed which is dependent on the ratio of the two eigenvalues with greatest absolute value. Another result of this modification affects the usability on bipartite graphs. Those have a symmetric spectrum (each positive eigenvalue has a negative complement) and therefore convergence of the power iteration method can not be guaranteed, which can be bypassed by the spectrum shift from the added self loops.

Since all three presented approaches are only slight modifications of the first by either using a fixed decay factor or modifying the graph we will concentrate our experiments on the first variant.

4.2 Normalization

There are two arguments for the introduction of normalization into spreading activation. On the one hand spreading functions designed for a specific purpose could inherently demand normalization, like the application proposed in Section 5.3. On the other hand we have seen in the previous section that in the accumulation method, the influence of the single iteration steps is a combination of the decay factor α and the spectral radius of the network. Normalizing the actvitation vectors in each iteration is a way to eleminate the restriction posed on α by the spectral radius. Therefore α can be chosen liberally between 0 and 1 in linear systems with normalization.

Following these arguments we take a look at systems where the activation vector is normalized in each iteration:

$$\mathbf{a}^{(k)} = \frac{W\mathbf{a}^{(k-1)}}{\|W\mathbf{a}^{(k-1)}\|}.$$

We restrict our analysis to the case in which $\|\cdot\|$ denotes the l_2 norm, analog to the power iteration method discussed above. Different norms can be discussed with the same arguments. Apparently the sequence of $\mathbf{a}^{(k)}$ converges under the assumptions of the Perron-Froebenius Lemma to the eigenvector of W corresponding to the eigenvalue $\rho(W)$. Again we look at the limit of the series $\sum_{k=0}^{\infty} \alpha^k \mathbf{a}^{(k)}$ in the hope that the influence of $\mathbf{a}^{(0)}$ yields a useful ranking of the fixed point of the series. Here we show that such a fixed point always exists, even if W does not satisfy the Perron-Froebenius conditions. In a later section we will analyze the quality of the resulting ranking empirically.

Theorem 1. *Given a matrix $W \in \mathbb{R}^{n \times n}$ and $\alpha \in [0,1)$ there exists a vector $\mathbf{a}^* \in \left[-(1-\alpha)^{-1}, (1-\alpha)^{-1}\right]^n$ such that*

$$\mathbf{a}^* = \lim_{m \to \infty} \sum_{k=0}^{m} \alpha^k \mathbf{a}^{(k)}$$

with

$$\mathbf{a}^{(k)} = \frac{W\mathbf{a}^{(k-1)}}{\|W\mathbf{a}^{(k-1)}\|}.$$

Proof. Since $\|\mathbf{a}^{(k)}\| = 1$ it is apparent that $|\mathbf{a}_i^{(k)}| \leq 1 \; \forall i \in V$. Therefore each $\mathbf{a}_i^*$ is bounded by the geometric series:

$$\mathbf{a}_i^* \leq \sum_{k=0}^{\infty} \alpha^k = (1-\alpha)^{-1}$$

and

$$\mathbf{a}_i^* \geq -\sum_{k=0}^{\infty} \alpha^k = -(1-\alpha)^{-1}.$$

Since $\lim\limits_{k \to \infty} \alpha^k \mathbf{a}_i^{(k)} = 0$ this concludes the proof.

4.3 A Glimpse at Non-Linear Systems

To emphasize node activations some authors make use of special activation functions. The simplest approach is to use sign or threshold functions [2, 15, 16]. In threshold functions a constant is subtracted from the activation value of a node and afterwards the sign function (sgn) is applied to the result, mapping it to $\{-1, 1\}$. Apparently a sign function is a special case of thresholds with a zero constant. Other authors use sigmoid functions [12, 17] as a continuous approximation. When using the threshold function as activation method, spreading activation equals the processes of discrete Hopfield networks with parallel node processing. In our framework parallel processing is always assumed while in Hopfield networks also strategies of processing nodes in arbitrary orders have been examined.

In this method convergence is not completely characterizable as for the systems presented above. In the following we will apply the most general result about convergence in this field to spreading activation. Then we will have a look at the number of fixed points in such systems. This is not intended as a full review of the current state of art but aims to give a context for comparison with the systems analyzed hitherto.

Convergence In this part we will concentrate on the results of Kwong and Xu presented in [18] who generalize preliminary results of Hopfield [19] for sequential activation and Goles, Fogelman and Pellegrin [20] for parallel activation.

A discrete, parallel Hopfield network is defined by a tuple $(W, \mathbf{t})$ where W is a matrix of connection strengths (weights) and $\mathbf{t}$ is a vector of thresholds. In each iteration step all neurons of the network are simultaneously (in parallel) activated depending on the output of the adjacent neurons in the previous iteration. In difference to the linear systems described earlier the activation function of the neurons applies a threshold: $\mathbf{a}_v^{(k)} = \operatorname{sgn}\left(\mathbf{i}_v^{(k)} - \mathbf{t}_v\right)$. In spreading activation applications the thresholds are usually equal for all nodes, such that $\mathbf{t} = t \cdot \mathbf{1}$ for some threshold value t.

For the most general convergence criterion a Matrix W^* is derived from W by modifying its diagonal:

$$(W^*)_{ij} = \begin{cases} (W)_{ii} - \frac{1}{2} \sum\limits_{k=1}^{n} \left|(W)_{ki} - (W)_{ik}\right| & \text{, if } i = j \\ (W)_{ij} & \text{, if } i \neq j \end{cases} .$$

Using this definition we can present the following theorem proven by Kwong and Xu in [18].

Theorem 2. *Let (W, T) be a Hopfield network, W not necessarily symmetric. Given that W^* is positive semidefinite (W, T) will converge to a stable state.*

Restated in the context of spreading activation this can be interpreted analogous to the third variant of linear systems described in Section 4. It is known that if W^* is positive semidefinite then $\operatorname{trace}(W^*) = \sum_{i=0}^{n} (W^*)_{ii} \geq 0$ and all principal minors of W^* are positive semidefinite. Under these properties it is apparent that a necessary condition for Theorem 2 is that $(W)_{ii} \geq \frac{1}{2} \sum_{k=1}^{n} |(W)_{ki} - (W)_{ik}|$. In the graph the spreading process is applied to, nodes necessarily have self loops with a weight that dominates the asymmetry of their directed edges.

For the symmetric case it is therefore sufficient that W itself is positive semidefinite, which was already shown in [20]. Goles et al. further show that even without this condition the network will converge to a cycle with a maximum length of 2. We can therefore reason that activations as presented here will in undirected graphs either converge to a single state or flip between two final states. Note further that Theorem 1 can be applied here directly.

Fixed Points We characterized some situations in which spreading activation systems using threshold functions will converge to fixed points and we showed that some retrieval systems are pointless because they only have a single fixed point. This raises the question if systems based on Hopfield networks have enough fixed points to be practicably usable in information retrieval. Since a thorough examination of this points is beyond the scope of this paper we will restrict ourself to some hints on previous work. Results regarding the number of fixed points in the context of Hopfield networks are generally concerned with methods building matrices with a predetermined number of fixed points. A good introduction on this topic in the context of Hopfield networks can be found in Haykin [21, Chapter 14]. Still, even if matrices are designed in such ways there also exist additional spurious fixed points. However, results in this field apply only to the predetermined fixed points and make no statement on their overall number. Therefore, they are not applicable to investigate the number of fixed points in our scenario.

Yet, there is an analogy to Hopfield networks appearing in physics, namely spin-glass systems. Those are also examined for their convergence but, as opposed to results in Hopfield networks, in a way that no distinction between wanted and spurious fixed points is made. The number of possible fixed points to expect in these systems is given by McEliece and Posner in [22]. Here a detailed analysis of the average number of fixed points in symmetric networks without self loops is given. The average number of fixed points is roughly $2^{\frac{n}{4}}$, with n being the number of nodes in the network.

5 Application

For empirical examinations of the theoretical results derived above we define two application scenarios.

The first is an example of a simple linear system. In [23, 24] spreading activation techniques are used for concept exploration and query expansion. We will follow this approach by using a term co-occurrence network to expand a given query in Section 5.2. Here the knowledge about domain specific term similarities, encoded in the co-occurrence network, is utilized by spreading activation.

An application of normalized systems is given in Section 5.3 where spreading activation is used to query a term-document network for documents based on cosine similarity. The approach is similar to those in [25, 26]. In this model, alternating cosine similarities between (virtual) documents and (virtual) terms contribute to the result.

5.1 Preliminaries

The basis for both application scenarios is a bipartite document-term-graph. The nodes of the graph $G = (V, E)$ are partitioned into $V = D \uplus T$, where partition D has a node for each document and partition T one for each term. The edge set consists of $E = \{\{d, t\} : d \in D, t \in T, w(d, t) > 0\}$, where $w(d, t)$ is a function describing the relation between documents and terms. Without loss of generality, we assume that G is connected, that is, there exists a path between any two nodes in the graph.

Throughout our experiments, we will use the tf-idf values of terms and documents as weighting function $w(d, t)$. A standard definition of tf-idf values is

$$w(d, t) = \mathrm{tfidf}_{d,t} = \frac{f(t, d)}{n_d} \cdot \log\left(1 + \frac{|D|}{n_t}\right)$$

where $f(t, d)$ is the absolute frequency of term t in document d, n_t is the number of documents in which term t appears and n_d is the number of terms contained in document d.

Typical text-mining approaches work with the so called *vector space model*, where each document is a vector in the $|T|$-dimensional space spanned by the $|T|$ terms in the collection. Using this representation, the documents d of the collection are often ranked by calculating a similarity measure between their corresponding vector representations $\mathbf{d}$ and the vector representation $\mathbf{d_q}$ of a query document d_q. Most prominent among these is the *cosine similarity*, given by

$$\cos(d, d_q) = \frac{\mathbf{d}\mathbf{d_q}}{\|\mathbf{d}\|\|\mathbf{d_q}\|}$$

Note that, vice versa the terms of the collection can be represented as a vector in the vector space spanned by the documents.

5.2 A Simple Linear System: Query Expansion on a Term Co-occurrence Network

The first application we examine is a case of a linear system without normalization. We construct a co-occurrence graph of terms G_c from the original bipartite graph G. Each term in G is also a node in G_c. The weighting function between pairs of terms is defined as

$$w_c(i,j) = \sum_{d \in D} w(d,i)w(d,j),$$

that is, the weight between two terms $i, j \in T$ is the inner product of their corresponding vectors in the vector space model. The edge set of G_c can then be defined appropriately by $E_c = \{\{i,j\} : w(i,j) > 0\}$. Based on the co-occurrence graph G_c one could try to perform query expansion by a spreading activation technique on the graph. This could be achieved by providing a simple activation function as

$$\mathbf{a}_i^{(k)} = \sum_{j \in N_c(i)} w_c(i,j) \cdot \mathbf{a}_j^{(k-1)}$$

where k is the current iteration, the input and output functions are the identity functions and $N_c(i) = \{j : \{j,i\} \in E_c\}$. It is easy to see, that this method can be directly transformed into a matrix power iteration of the adjacency matrix $W_c \in \mathbb{R}^{T \times T}$ of G_c, where $(W_c)_{ij} = w_c(i,j)$. Also note that G_c is not bipartite. Thus, independent of the initial activated terms, the resulting activation will converge to the principal eigenvector of W_c corresponding to $\rho(W_c)$.

After the query terms have been expanded by spreading activation, the resulting term vector can be used to rank the documents, for example, by calculating the cosine similarity between the documents and the expanded term vector. We will use this ranking to investigate the performance of the accumulation strategy in Section 6.

5.3 A System with Normalization: Alternating Cosine Similarity

To examine the statements about systems with normalization, let us now consider the bipartite network proposed in Section 5.1. Every given activation of the term-partition corresponds to a vector of a virtual document in the vector space model; vice versa, every activation of the vertices in the document partition may be seen as a virtual term. The spreading activation framework proposed in the following is designed to yield as activation at a given time the cosine similarity to the virtual document, resp. term given by the current activation levels of the other partition.

As in [25], a single spreading of activation on the bipartite network yields the standard cosine similarity measure between a virtual query document and the documents in the network. To achieve the desired activation, spreading functions for each partition have to be defined appropriately. In the following, let k be the index for the current iteration, where one iteration is a spreading of activation from the document partition to the term partition and backwards. Initially, i.e. $k = 0$, the query terms are activated and activation is just spread from the term to the document partition.

For the term partition T the combined activation function is:

$$\mathbf{a}_t^{(k)} = \frac{\sum\limits_{d \in N(t)} \mathbf{a}_d^{(k-1)} w(d,t)}{\sqrt{\sum\limits_{d \in N(t)} w(d,t)^2} \cdot \|\mathbf{a}_d^{(k-1)}\|}.$$

For the partition containing the documents D the combined activation function is defined similarly; the only difference is that documents receive their activation from the activation state of terms in the same iteration k:

$$\mathbf{a}_d^{(k)} = \frac{\sum\limits_{t \in N(d)} \mathbf{a}_t^{(k)} w(d,t)}{\sqrt{\sum\limits_{t \in N(d)} w(d,t)^2} \cdot \|\mathbf{a}_t^{(k)}\|}.$$

Given a query document d_q, let $\mathbf{d_q}$ be the corresponding normalized vector space representation of d_q. Then the cosine similarity of d_q to every document d in the network is calculated by activating the terms contained in d_q, i.e. $\mathbf{a}_t^{(0)} := (\mathbf{d_q})_t$, and spreading the activation to the document partition using the spreading function as defined above, since

$$\mathbf{a}_d^{(0)} = \frac{\sum\limits_{t \in N(d)} \mathbf{a}_t^{(0)} w(d,t)}{\sqrt{\sum\limits_{t \in N(d)} w(d,t)^2} \cdot \|\mathbf{a}_t^{(0)}\|} = \frac{\mathbf{d}\mathbf{d_q}}{\|\mathbf{d}\|\|\mathbf{d_q}\|} = \cos(d, d_q).$$

As stated above, the overall resulting activation of the document partition, represented as a vector $\mathbf{a}_d(0)$ can be seen as a virtual term. When the next spreading is executed, i.e. the spreading function for the term partition is applied, each term will receive its cosine similarity to this virtual term as activation, and so on.

To analyze the convergence properties of the proposed framework, it will now be transformed into a matrix representation, such that the statements of the previous section can be applied. Let $W \in \mathbb{R}^{D \times T}$ be the matrix constituted by the weighting function $w(d,t)$, i.e. $(W)_{dt} = w(d,t)$. Let $W_D \in \mathbb{R}^{D \times T}$ be the l_2-row-normalization of W, that is

$$(W_D)_{dt} = w(d,t)/\|\mathbf{d}\|.$$

Analogous, let $W_T \in \mathbb{R}^{T \times D}$ be the l2-row-normalized version of the transposed matrix W, i.e.

$$(W_T)_{td} = w(d,t)/\|\mathbf{t}\|.$$

Furthermore, let $\mathbf{a}_D^{(k)}$ denote the output vector at time k, restricted to the components corresponding to documents, and $\mathbf{a}_T^{(k)}$ analogous for the restriction on terms, i.e.

$$\mathbf{a}_D^{(k)} = \mathbf{a}^{(k)}\Big|_D \text{ and } \mathbf{a}_T^{(k)} = \mathbf{a}^{(k)}\Big|_T .$$

Now, the spreading functions as given above can be reformulated in matrix notation as

$$\mathbf{a}_D^{(k)} = \frac{W_D \cdot \mathbf{a}_T^{(k)}}{\|\mathbf{a}_T^{(k)}\|}$$

where

$$\mathbf{a}_T^{(k)} = \frac{W_T \mathbf{a}_D^{(k-1)}}{\|\mathbf{a}_D^{(k-1)}\|}.$$

When put together this yields

$$\mathbf{a}_D^{(k)} = W_D \frac{W_T \mathbf{a}_D^{(k-1)}}{\|\mathbf{a}_D^{(k-1)}\|} \cdot \left\| \frac{W_T \mathbf{a}_D^{(k-1)}}{\|\mathbf{a}_D^{(k-1)}\|} \right\|^{-1} = \frac{W_D W_T \mathbf{a}_D^{(k-1)}}{\|W_T \mathbf{a}_D^{(k-1)}\|}.$$

By recursive substitution we have a power iteration representation of the spreading activation framework presented here, i.e.

$$\mathbf{a}_D^{(k)} = W_D \frac{(W_T W_D)^k \mathbf{a}_T^{(0)}}{\|(W_T W_D)^k \mathbf{a}_T^{(0)}\|}$$

The matrix $W_T W_D$ represents a connected, non-bipartite graph and therefore satisfies the conditions for convergence by the Perron-Froebenius Lemma. Hence, the statements of the previous section apply to the proposed spreading activation framework. Thus again, the resulting activation will converge to the principal eigenvector of $W_T W_D$ transformed to the document space by a multiplication with W_D. The strategy of accumulating the activation of the individual iterations will be reviewed experimentally in the next section.

6 Experimental Results

To demonstrate exemplarily that pure spreading activation does not work to find relevant items, and how, in contrast, the cumulation of iteration results and

Test collection	TIME	MED
Number of documents	425	1033
Number of queries	83	30
Average number of relevant documents	3.9	23.2
Spectral radius $\rho(G_c)$	1.81	1.28

Table 1. SMART test collection statistics

additional constraints affect the search results, we applied spreading activation to parts of the SMART test collection[1]. For the experiments we used two corpora, TIME and MED, with the given queries and reference rankings. The number of documents, queries and average relevant documents of the test sets, as well as spectral radius $\rho(G_c)$ of their co-occurrence graphs, are listed in Table 1. For the sake of brevity we only give illustrations for the TIME dataset and state differences encountered in MED where necessary.

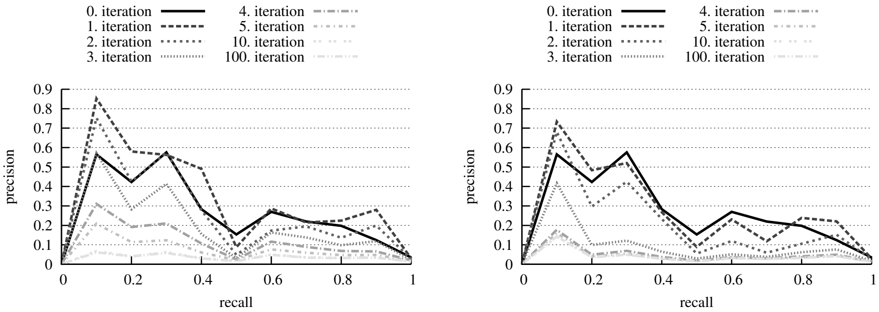
To build the bipartite document-term graph described in section 5.1, the data went through common preprocessing steps: stop words were filtered, words were stemmed and converted to lower case. Furthermore the tf-idf values have been computed for each word and used as edge weights. The queries have been preprocessed in the same way. A query was applied by activating the nodes representing the query terms with an initial activation of 1. Activation was then spread across the graph according to the two procedures described in section 5.2 and 5.3. The resulting documents were ordered according to their final activation value. Precision and recall was measured in order to compare the results of the two methods and of different numbers of iterations. For each test collection the precision was measured at 11 equidistant points of recall for each query result. Finally, the average precision over all queries for each of the 11 recall points was computed.

We investigate two strategies of obtaining a final result based on the individual iterations: First, we look at pure iteration results. Afterwards, we evaluate the cumulation strategy *Accumulation*, as described in Section 4.1. We want to stress that the purpose of the following experiments is not to test whether the presented applications perform well compared to others. In contrast, we demonstrate the failure of pure spreading activation and the effectiveness of cumulation regarding query-dependent results.

6.1 Pure Iteration

The final result in these tests is the activation of the last iteration. Previous iterations are not cumulated or taken into account. Figure 1a shows the precision-recall results of the query expansion application, which forms a linear system with no normalization. Figure 1b displays the results of the alternating cosine similarity application, which is a system with normalization. In all precision recall plots, the continuous line represents the precision of the initial state. This

[1] ftp://ftp.cs.cornell.edu/pub/smart



(a) Query expansion application

(b) Alternating cosine similarity application

Fig. 1. Precision and recall results of single iterations of spreading activation.

corresponds exactly to the cosine similarity measure between the query vector and the documents in both applications, and serves as a baseline for our investigations. The dotted lines depict the precision corresponding to the results of iterations 1-5, 10, and 100.

As predicted by the theoretical results, the empirical tests show that convergence to the query-independent fixed point is not adequate in order to generate satisfying query results. The precision curves of the 10^{th} and the 100^{th} iteration are almost identical, due to the convergence of the system. In both systems, only the first and second iteration outperform the cosine measure for some recall ranges. The performance decreases with each additional iteration. One could now put a constraint on the maximum number of iterations of the spreading method to terminate the process early enough before the fixed point is reached. However, the determination of the optimal number of iterations is complicated. In case of the TIME dataset, the first iteration performs best; but for MED, the results of the first iteration are already worse than the standard cosine measure. This shows that the constraint when to abort spreading can vary for different queries and datasets.

To illustrate the convergence behavior Figure 2 shows the trajectories of the iteration process for some sample queries. Activations of each step are projected into 2 dimensions with multi-dimensional scaling (MDS, [27]). MDS is a projection of data into a lower-dimensional space, such that dissimilarities of vectors in the high-dimensional space are conserved in the projection as good as possible. As dissimilarities for the MDS we used an inverted cosine between the activation vectors in document space. The dissimilarity of two vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ is $\mathrm{dis}(\mathbf{r}_1, \mathbf{r}_2) = 1 - \cos(\mathbf{r}_1, \mathbf{r}_2)$. We used this angular dissimilarity, to aid the geometric interpretation of the progress of intermediate results. Clearly, for comparison of two rankings, a metric based on ranking inversions would be more appropriate since activation vectors resulting in equal rankings would be projected onto the same coordinates with this measure. In contrast, the dissimilarity

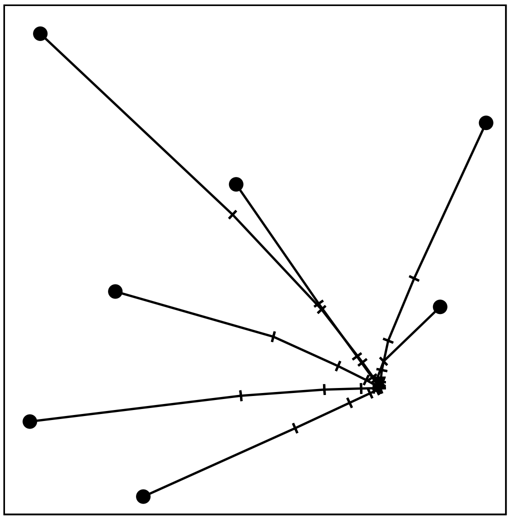


Fig. 2. Trajectories of 7 queries of the alternating cosine similarity application with pure spreading activation. Coordinates are obtained by projecting intermediate results into 2D with MDS, such that angular distances between activation vectors are conserved. Each query starts at a dot, marking the result of iteration 0, subsequent iterations are marked by dashes and finally meet at the principal eigenvector of the system.

based on the cosine has the ability to differentiate between different activation vectors implying the same ranking. This is more suiting to illustrate how fast and to what extend the individual intermediate results approach the common fixed point. In the figure, consecutive intermediate results are connected by a line, starting at the activation of direct cosine distance depicted as a dot. It is evident from the trajectories that all activation results converge very quickly to the same fix point, that is, the principal eigenvector of the system.

6.2 Accumulation

The second method we examine is *Accumulation* as described in Section 4.1. Here, the activation of each iteration is cumulated, with a particular decay, to a final result. This assures convergence to a fixed point dependent on the query, if the decay rate meets certain conditions.

As stated in Section 4, the decay factor α for non-normalized systems must obey $\alpha < \rho(G_c)^{-1}$ to ensure convergence. In our case we use as denominator a value that is just higher than the spectral radius, that is, $\alpha = 1/1.9 = 0.526$ for the TIME collection and $\alpha = 1/1.3 = 0.769$ for the MED collection. The precision and recall values of the query expansion application in Section 5.2 are displayed in Figure 3. In this and subsequent precision-recall figures, the

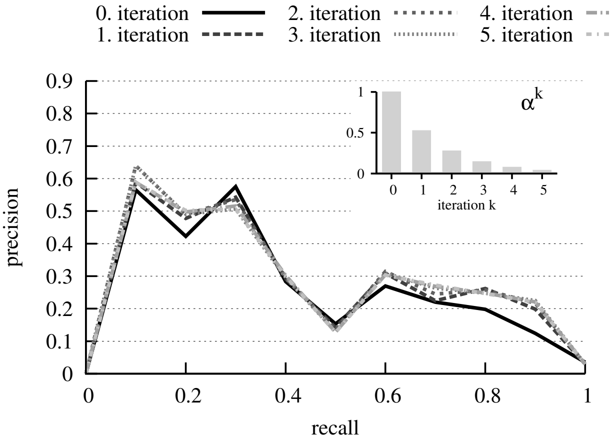


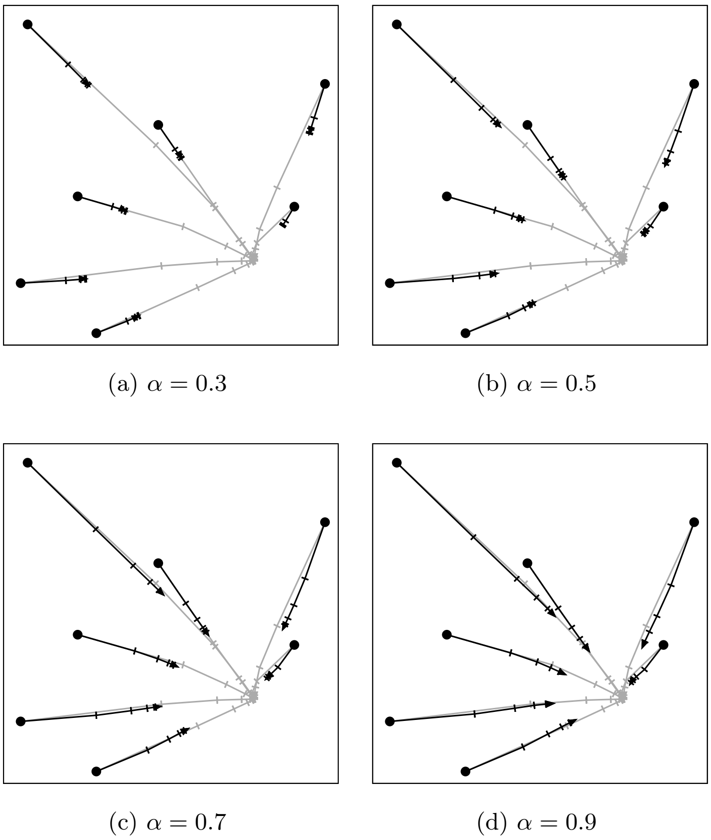
Fig. 3. Precision and recall results of the query expansion application with accumulated spreading activation. From initial state (continuous line) accumulated up to 5th iteration (dotted lines), with a decay of $\alpha = 0.526$.

embedded bar chart in the top-right corner displays the decay of α^k to show the influence of the single iteration results on the final result.

The performance of accumulation is better than that of pure spreading activation. Yet, depending on the possible decay value, the first few iterations contribute most, and later iterations less. We argued for systems with normalization for exactly this reason, that is, being able to choose the decay factor α independently of the spectral radius. Also, the computation of the spectral radius can be very time-consuming.

For normalized systems like the alternating cosine distance application, we have shown that there are no constraints on α except that it must obey $0 < \alpha < 1$. When α is set to a low value, the single iterations loose influence on the final results very quickly, whereas later iterations still contribute quite significantly when α is set to a high value. The trajectories of the intermediate results, displayed in Figure 4, illustrate the influence of α on the results. The trajectory of intermediate results of pure spreading activation is outlined in gray in each figure. The dark arrows depict the trajectories of intermediate results for each value of α. The accumulated results converge to a fixed point, which is dependent on the query, and different from the principal eigenvector of the system. Still, the trajectory of angles between intermediate results point in the direction of this eigenvector. Therefore, the greater the value of α, the more will the fixed point of an accumulation system approach the fixed point of the pure system with respect to their angular distance in the high-dimensional space. This implies that with a high value of α, the system will give more “general” answers to a query, whereas, with a low value of α, it will return more “specific” answers.

The precision and recall results corresponding to the alternating cosine similarity application are shown in Figure 5 for two values of α. Again, the results of



(a) $\alpha = 0.3$

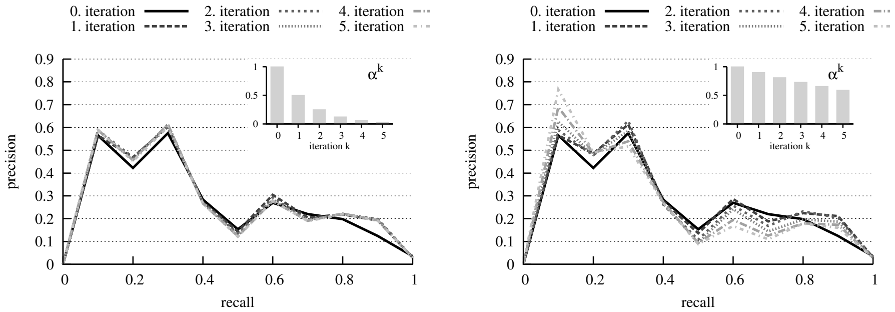
(b) $\alpha = 0.5$

(c) $\alpha = 0.7$

(d) $\alpha = 0.9$

Fig. 4. Trajectories of intermediate results of the alternating cosine similarity application with accumulated spreading activation and different decay factors. The trajectories of Figure 2 are depicted in gray for comparison, while trajectories of accumulated results with given decays are shown in black.

accumulation perform significantly better than those of pure spreading activation. With a low value of α, the resulting rankings do not differ very much from the result of just calculating the cosine similarities of documents to the query. In contrast, a high value of α significantly changes the ranking. For the TIME dataset, accumulation with a high value of α yields a good precision for low recall ranges, but performs worse than the baseline in higher recall ranges. For the MED dataset, a high value of α also significantly changes the ordering, but more in the intermediate recall ranges. However, in MED a low value of α results in better performance. This demonstrates that an optimal value for α can not be determined without knowledge about the underlying dataset. Nevertheless, the continuity of results corresponding to continuous choices for α, and the natural



(a) $\alpha = 0.5$

(b) $\alpha = 0.9$

Fig. 5. Precision and recall results of the alternating cosine similarity application with accumulated spreading activation and two different decay factors. From initial state (continuous line) accumulated up to 5th iteration (dotted lines).

interpretability of this parameter gives the accumulation strategy an advantage over the choice of a discrete constraint to control spreading.

7 Conclusions

Commonly used spreading activation strategies are based on linear systems with additional constraints. We pointed out that, without constraints, these systems can be directly transformed to a power iteration method. Therefore some set constraints is essential to avoid convergence to query-independent fixed points. We proposed accumulation of single iteration results as an orthogonal approach to constraints and showed that this leads to query dependent results. To ensure convergence in this method, a decay factor decreasing the influence of subsequent iterations, has to be introduced. We confirmed our theoretical findings empirically using two application scenarios. Furthermore, the influence of the decay factor on the results was shown exemplarily with these applications. The experiments suggest that the decay factor could be a promising tool to control the degree of generality of the results.

The effect of non-linear activation functions with regard to their convergence behavior remains to be analyzed more thoroughly in the context of spreading activation. Furthermore, other decay and combination functions can be examined. Additionally, traditional constraints can be combined with the proposed methods.

8 Acknowledgment

The work presented in this paper was supported in part by DFG under grant GRK 1042 (Research Training Group "Explorative Analysis and Visualization of

Large Information Spaces") and the European Commission under the 7th Framework Programme FP7-ICT-2007-CFET-Open, contract no. BISON-211898.

References

1. Quillian, M.: Semantic memory. In Minsky, M., ed.: Semantic Information Processing. MIT Press (1968)
2. Collins, A., Loftus, E.: A spreading-activation theory of semantic processing. Psychological Review **82**(6) (1975) 407–428
3. Cohen, P., Kjeldsen, R.: Information retrieval by constrained spreading activation in semantic networks. Information Processing and Management: an International Journal **23**(4) (1987) 255–268
4. Belew, R.: A connectionist approach to conceptual information retrieval. In: Proceedings of the 1st international conference on Artificial intelligence and law (ICAIL '87), New York, NY, USA (1987) 116–126
5. Salton, G., Buckley, C.: On the use of spreading activation methods in automatic information retrieval. In: Proc. 11th Ann. Intl. ACM SIGIR Conf. Research and Development in Information Retrieval. (1988) 147–160
6. Salton, G.: Automatic Information Organization and Retrieval. McGraw Hill (1968)
7. Ziegler, C., Lausen, G.: Spreading activation models for trust propagation. In: Proceedings of 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04), Washington, DC, USA (2004) 83–97
8. Liu, W., Weichselbraun, A., Scharl, A., Chang, E.: Semi-automatic ontology extension using spreading activation. Journal of Universal Knowledge Management **0**(1) (2005) 50–58
9. Tsatsaronis, G., Vazirgiannis, M., Androutsopoulos, I.: Word sense disambiguation with spreading activation networks generated from thesauri. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07). (2007)
10. Kovacs, A., Ueno, H.: Recommending in context: A spreading activation model that is independent of the type of recommender system and its contents. In: Proceedings of the Workshop on Recommender Systems and Intelligent User Interfaces at the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2006). (2006)
11. Rumelhart, D., McClelland, J.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition. MIT Press (1986)
12. Crestani, F.: Application of spreading activation techniques in information retrieval. Artificial Intelligence Review **11**(6) (1997) 453–482
13. Godsil, C., Royle, G.: Algebraic Graph Theory. Springer-Verlag (2001)
14. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**(1) (1953) 39–43
15. Belew, R.K.: Adaptive information retrieval: using a connectionist representation to retrieve and learn about documents. In: SIGIR '89: Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA (1989) 11–20
16. Berger, H., Dittenbach, M., Merkl, D.: An adaptive information retrieval system based on associative networks. In: Proceedings of the first Asian-Pacific conference on Conceptual modelling, Darlinghurst, Australia (2004) 27–36

17. Crestani, F., Lee, P.: Searching the Web by constrained spreading activation. Information processing and Management **36**(4) (2000) 585–605
18. Kwong, C., Xu, Z.B.: Global convergence and asymptotic stability of asymmetric hopfield neural networks. Journal of Mathematical Analysis and Applications **191** (1995) 405–427
19. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences **79**(8) (1982) 2554–2558
20. Goles, E., Fogelman, F., Pellegrin, D.: Decreasing energy functions as a tool for studying threshold networks. Discrete Applied Mathematics **12** (1986) 261–277
21. Haykin, S.: Neural Networks A Comprehensive Foundation. Prentice Hall, Upper Saddle River, New Jersey (1994)
22. McEliece, R.J., Posner, E.C.: The number of stable points of an infinite-range spin glass memory. Telecommunications and Data Acquisition Progress Report **42–83** (1985) 209–215 Jet Propulsion Laboratory, California Institute of Technology, Pasadena.
23. H. Chen, T.N.: An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): symbolic branch-and-bound search vs. connectionist hopfield net activation. Journal of the American Society for Information Science **46**(5) (1995) 348–369
24. Strzalkowski, T.: Natural Language Information Retrieval. Text, Speech and Language Technology. Kluwer Academic Publishers (1999)
25. Wilkinson, R., Hingston, P.: Using the cosine measure in a neural network for document retrieval. In: Proc. 14th Ann. Intl. ACM SIGIR Conf. Research and Development in Information Retrieval. (1991) 202–210
26. Cunningham, S., Holmes, G., Littin, J., Beale, R., Witten, I.: Applying connectionist models to information retrieval. Brain-like computing and intelligent information systems (1997) 435–457
27. Cox, T.F., Cox, M.A.: Multidimensional Scaling. Volume 88 of Monographs on Statistics and Applied Probability. Chapman&Hall/CRC (2001)

Interactive Visualization of Continuous Node Features in Graphs

Stefan Haun, Marcus Nitsche, Andreas Nürnberger

Data and Knowledge Engineering Group,
Departement of Computer Science,
Institute of Technical and Business Information Systems,
Otto-von-Guericke-University of Magdeburg,
Universitätsplatz 2, 39106 Magdeburg, Germany

{stefan.haun, marcus.nitsche, andreas.nuernberger}@ovgu.de

Abstract: Ordinary graphs only support discrete structures. In this paper we present an approach towards embedding continuous data – like time stamps or series of measurements – in discrete graph models. These continuous meta-information implicitly define relations between vertices which are not explicitly defined in the graph itself. We call this an induced Non-Discrete Graph Structure (NoDeS). The model is helpful for visualization of time-dependent models or values from physical domains. We provide a formal definition of NoDeS based on graphs and two mappings, instance and annotation based, to already known graph structures and visualizations. A visualization of multi-partite projection provides a representation of information from several contexts, enabling NoDeS for a generic context switching mechanism which is used for interaction with these structures. Finally, we introduce an application concept for agent-driven event scheduling using NoDeS.

1 Introduction

In discrete mathematics graphs are a solid theoretical approach to model and visualize relations between items and to visualize complex structures in many different areas (cf. [DL07]). While these structures are static and discrete we thought about adding continuous data to classic graph structures. This supports visualizing relations between instances as well as sticking continuous node features like time measurements for instance. In this paper we present the formal description of this enhanced graph structure as well as interaction sequences of end-users.

We first give a brief overview of related work in the area of graph visualization and systems related to web searching and the visualization of hypermedia structures. Then we introduce our approach towards embedding continuous domains into graphs. A concept for the interactive visualization of the presented structure is discussed in Section 4. Finally, in Section 5, we present an application example based on the problem of Agent-driven event scheduling to show how our approach can be used.

2 Related Work

Related work can be found in the field of graph visualization and graph layouting. Cook and Holder, although mostly concerned with graph mining, provide a good overview on the state of the art and current systems, [CH07]. For a general overview there are several surveys on graph visualization available (c.f. [HMM00], [DBETT94], [Tam99]). According to [DL07], there are three major methods for graph layouting: *force-directed*, *hierarchical* and *topology-shape-metrics*, where the force directed method introduced by [Ead84] is most used today, despite its disadvantageous behavior in interactive systems [HMM00], such as presented in Sec. 5. Special visualizations can be used to accommodate data specific features such as time lines: [DRS04] introduces a 2.5D time-series data visualization, which uses stacks to represent time-dependent advances in data series. However, all existing layout and visualization methods do not take continuous domains into account and rather rely on the inherent relations of the continuous data, such as ordering and distance, being encoded into the discrete graph structure.

A large number of visualization systems is available. Approaches tailored to web searching and the visualization of hypermedia structures can be found among the web meta-search clustering engines (Vivísmo[1], iBoogie[2], SnakeT[3], WhatsOnWeb[4]) and in the field of semantic wikis (iMapping Wiki [HKV06]).

Combining visualization methods become more interesting in the context of heterogeneous information networks as they are heavily used throughout the BISON project[5].

3 A Structure for Visualizing Continuous Node Features

For our approach, we use the graph definition as proposed in [Die06]:

> A *graph* is a pair $G = (V, E)$ of disjoint sets with $E \subseteq V^2$, i.e. the elements of E are two-element subsets of V, where elements from V are *vertices* and elements from E are *edges* connecting the vertices.

Often the term *node* is used when referring to vertices and edges may be called *links*, especially in hyperlink structures.

[1] `http://vivismo.com/`

[2] `http://www.iboogie.com/`

[3] `http://snaket.di.unipi.it`

[4] `http://whatsonweb.diei.unipg.it`

[5] `http://www.bisonet.eu`

We call a vertex v continuous if meta-data from a continuous domain is mapped to v. There are two different mappings to ordinary graphs:

Mapping with meta-data. In the first mapping, continuous information is attached as meta-data to existing vertices (cf. Fig. 1). If there is a complete linkage between a continuous and a discrete domain, i.e. for each vertex from the continuous domain there is a matching vertex from the discrete domain, the continuous information can be attached to the discrete vertices. This resembles ordinary graphs with embedded meta-information. However, since the continuous vertices are only attached to other vertices, they are not a genuine part of the graph structure. It is not possible to distinguish between the continuous vertex and the mapping vertex in the graph structure and it is not possible to label the edges as they do not explicitly exist. Also, algorithms need to take this mapping into account and have to recognize attached meta-information as distinct vertices.

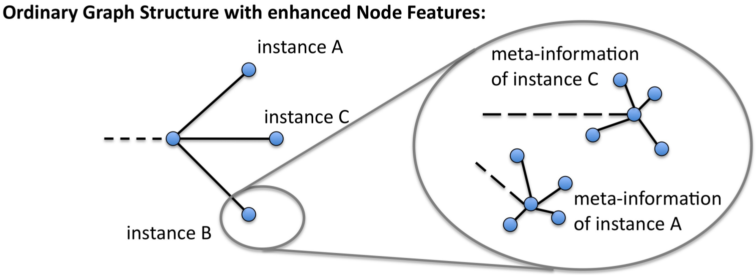

Ordinary Graph Structure with enhanced Node Features:
instance A
instance C
instance B
meta-information of instance C
meta-information of instance A

Figure 1: Graph with attached meta-information.

Mapping with meta-nodes. The second – preferred – mapping uses discrete nodes as instances from the continuous domain (cf. Fig. 2). Instead of attaching meta-information to existing nodes, we create meta-nodes which serve as a discrete instance of a value from the continuous domain. In contrast to the first mapping, these nodes do not represent any information other than the value from the continuous domain and yet can be treated as ordinary graph elements, i.e. meta-information can be attached and labeled edges can be added. However, two nodes representing the same value must be considered equal, i.e. before adding a meta-node, the graph must be searched for already existing nodes representing the desired value. As an advantage of this approach the resulting graph structure may contain continuous domains and still is an ordinary graph where existing algorithms can be applied. As a disadvantage meta-nodes do not represent relationships between elements from the continuous domain, i.e. two distinct nodes may have a very large distance, but also represent two very similar values. A distinction function needs to be used to decide whether two nodes should be considered equal to avoid cluttering the graph with similar value representations (with the penalty of imprecise representation of the continuous domain), which can introduce model knowledge about the underlying data, e.g. by using fuzzy sets.

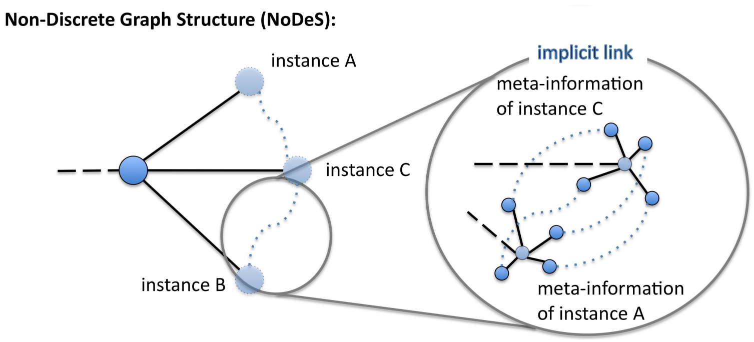

Non-Discrete Graph Structure (NoDeS):
instance A
instance C
instance B
implicit link
meta-information of instance C
meta-information of instance A

Figure 2: Interconnected meta-information, visualized as implicit links between vertices.

While both approaches result in an ordinary, annotated graph, the continuous graph structure differs in its semantics. The continuous vertices are bridges into the continuous domain rather than independent discrete elements. There may be an indirect relationship between those continuous vertices, if it is implied by the continuous domain, e.g. some distance measure. When calculating distances in the graph, those measures may be taken into account.

To introduce continuous vertices, the set V is built by the union of r multiple, arbitrary domains

$$V = V_1 \cup V_2 \cup \ldots \cup V_r$$

where each domain $V_1, \ldots, V_r$ may be either discrete or continuous, i.e. values from a continuous domain are embedded by adding a vertex as its discrete representative. The resulting graph not only contains vertices from discrete domains, but also elements from the continuous domain and their implicit connections, which are formed by the intrinsic ordering and distance function of the respective domain. For example, a pair of numbers in $\mathbb{R}$ is implicitly linked by the distance between the numbers (their difference) and has an ordering which implies a directed edge in the graph. The set of edges in the non-discrete graph structure is built from the set of explicit edges and the set of all edges induced by the continuous domain:

$$E = E_{Discrete} \cup E_{Implicit}$$

This type of graph we call **No**n-**D**iscret**e** graph **S**tructure (NoDeS) induced by a domain.

4 Interactive Visualization of NoDeS

Interactive variants of a NoDeS visualization can be easily implemented by setting single nodes or sets of nodes sensitive for user interactions like *selecting*, *dragging* and *dropping*. With slight adaptations, the NoDeS structure is not only interactive, but also allows to act context sensitive in a users perspective. Therefore NoDeS provide a generic context switching concept which allows to transfer the graph structure to different types of applications. Discrete vertices and meta-vertices, representing continuous data, are linked to each other (cf. Fig. 3).

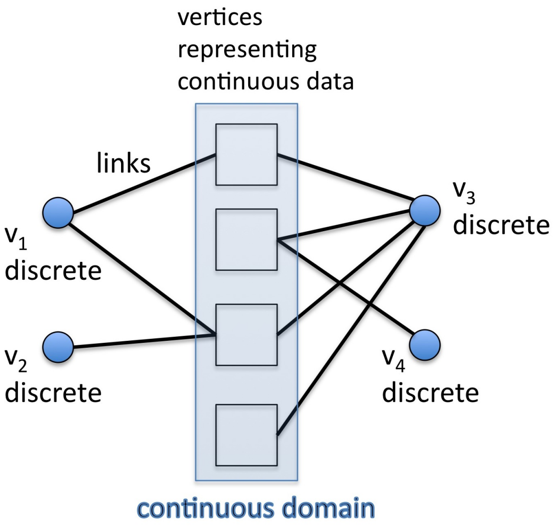


Figure 3: Generic structure of *NoDeS*.

Besides the indirect connections within the continuous domains, NoDeS can be defined as multi-partite graph structures if connections are only made between distinct domains from $V_1, \ldots, V_2$ (cf. Sec. 3). The resulting graph is visualized using methods for multipartite graphs, comparable to flow diagrams, like shown in [vdVvW00]: To interact in this information network we use *direct manipulation*, more precisely *drag-and-drop*, to support vague user requests. The laws of form like the *principle of proximity* can be applied to create context proximity and to map users intentions. Imaginable are user queries like in scheduling applications: *"I tend to not using this room for our meeting"* or *"It is not necessary for this special person to take part at our meeting"*. This creates *implicit context proximity* comparable to [TS00].

Furthermore visualization and interaction are integrated in the same user interface, which supports *directness*. It is left to be evaluated whether this technique enhances usability and user experience. A good evaluation result will show the benefit of direct interaction of this information network.

5 Application Example

In this chapter we discuss an application, which benefits from NoDeS as the underlying graph structure: Agent-driven event scheduling (cf. [Pay93], [PG02]).

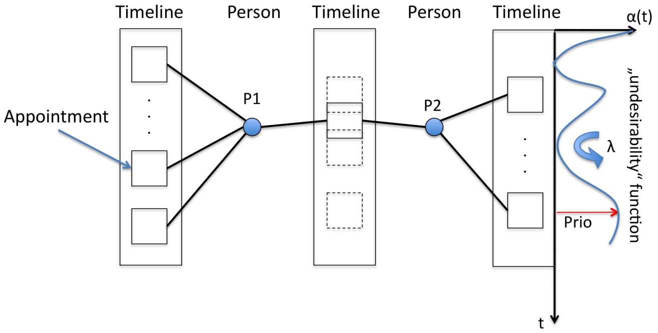


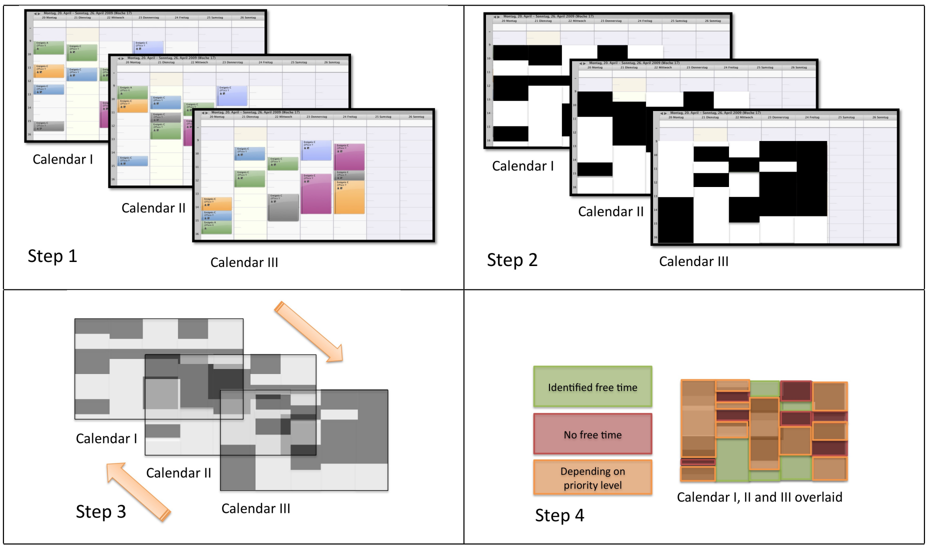
Figure 4: NoDeS used as schedule structure. Appointments are positioned along the continuous timelines but part of the graph structure and linked with the participants. The "undesirability function" steers priorities and the "leeway" between appointments.

From an application point of view, we refer to examinations of groupware aspects at the example of calendar applications like done in [GS86], on the work of Beard (cf. [BPH^{+}90]), who proposed a visual calendar for group meetings. Transferring the single user scheduling (cf. Personal Information Management (PIM) Systems, e.g. [JT06]) into multiple user applications, we also consider Computer Supported Cooperative Work (CSCW) topics like Community Building and user management. This transfer of a single into a multiuser mode is also discussed in [Eri06]. To our knowledge there is no approach using a graph of appointments and participants to represent the scheduling problem. We show a solution which uses NoDeS to formulate event scheduling as a minimization problem.

According to [LD89] we need to consider the prospect that calendars are a special form of continuous data integration. Contrary to time measurements we do not need to care about every second. When dealing with calendar scheduling problems we deal with appointments, which have a start time, an end time and a consequential duration time. The time line can be discretized by dividing the continuous time line into subsequent slots. The length of these slots decides the solution of the calendar, i.e. the granularity in which appointments can be made. However, the discretization yields some major disadvantages:

- The granularity of the calendar has to be chosen prior to its initial creation, i.e. whenever the resolution needs to be changed, the graph structure must be recalculated.
- The graph structure is bloated by nodes representing empty slots or subsequent nodes belonging to the same appointment.

Table 1: Steps for calendar preparation: From multiple calendars to a map showing free slots.



- From a semantic point of view, the calendar graph represents a sequence of time slots, where some of these slots may have special states denoting them as real appointments, rather than nodes representing appointments themselves. Especially when linking to a distinct appointment, it is difficult to decide which of the time slots contained in this appointment should be considered the *major vertex* for this appointment, i.e. to which of the slots the edge should be connected. Solving this issue by connecting any time slot to the appointment only shifts the problem, as all slots belonging to the appointment must be evaluated to obtain its neighbors.

- When the appointment is moved, the whole link structure must be adapted, i.e. edges removed and re-inserted in other vertices in the graph. Moving an appointment should be possible by just changing its start and end time.

To remedy these disadvantages we propose a graph representation which contains only existing appointments as vertices, which can have an arbitrary start and end on the continuous time line (cf. Fig. 4). Continuous node features are tagged to the classes “Appointments”, which are linked to discrete vertices like persons (e.g. P1, P2) taking part in the specific appointment. An “undesirability function” denotes the fitness of a specific time for scheduling a meeting: α represents the grade of “undesirability” of a date, where a value of zero represents a free time slot and greater values the grade to which the slot is blocked, i.e. its priority over other appointments. The slope of the graph, λ, specifies the “leeway” between appointments. The higher the value, the more will an appointment block time beyond its boundaries, leading to more free time between the specific dates. A λ-value of zero leads to subsequent appointments with no free time between them. This is

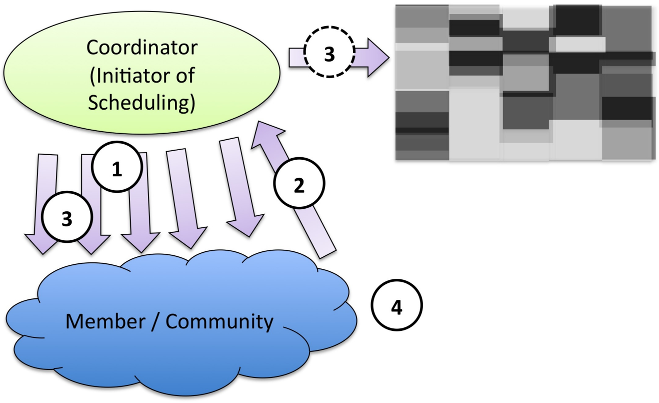


Figure 5: System architecture for scheduling application.

important to adapt vague scheduling. For optimal scheduling of a meeting the sum of the undesirability functions for all participants needs to be minimal.

Before using various calendars they first need to be prepared: By overlaying them (cf. Table 1) different steps of gray types appear, which allow a categorization of time slots: black (blocked, inevitable), gray (normal), white (free) and arbitrary fine-grained levels between them.

A possible technical realization of this special example is shown in in Fig. 5. First we assume the user maintains an online accessible calendar with priorities for each appointment. Similar to [JFR$^+$01] and [ESTT97] our system architecture provides a central coordinator, who initiates the scheduling. Given these information the following steps will be accomplished by the system:

1. A distinct ID is generated by the system.
2. Community members send and update their calendar data.
3. Estimated time slots are published.
4. The coordinator sends the first free time slots as a request to the user community.
5. The appointment has been accepted, otherwise go back to step 2.

Further examples of possible applications can be found in physics for instance: In thermodynamics and quantum physics usual discrete iterations are thus extended into a continuous flow (cf. [AF98]).

In future work one can also consider social aspects of scheduling and group meetings like thought about in [Pal99]. Also enhancements of our scheduler in a more personalized way like done in [TGW06] are possible. For readers who would like to read up more precisely on calendar use we suggest Brush's "Taking a closer look to calendar use" ([BT05]).

6 Conclusion

In this paper we proposed a conceptual design approach towards graph structures which is able to support continuous node features, called NoDeS. We presented an application, the agent-driven event scheduling, to provide a use-case for embedding continuous information in form of time-dependent nodes into the discrete structure of a network comprising human participants and resources such as meeting rooms.

This approach can be transferred to different domains. For example, to support creative exploration and discovery by switching between contexts (BISON) or to provide visualization and exploration of huge and complex data structures in energy and logistic networks (ViERforES).

7 Acknowledgement

Stefan Haun is funded by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898.

Marcus Nitsche is funded by the German Ministry of Eductaion and Science (BMBF) within the ViERforES project (no. 01IM08003C).

References

[AF98] R. Aldrovandi and L. P. Freitas. Continuous iteration of dynamical maps. *Journal of Mathematical Physics*, 39(10):5324–5336, 1998.

[BPH+90] David Beard, Murugappan Palaniappan, Alan Humm, David Banks, Anil Nair, and Yen-Ping Shan. A visual calendar for scheduling group meetings. In *CSCW '90: Proceedings of the 1990 ACM conference on Computer-supported cooperative work*, pages 279–290, New York, NY, USA, 1990. ACM.

[BT05] A.J. Bernheim Brush and Tammara Combs Turner. A survey of personal and household scheduling. In *GROUP '05: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pages 330–331, New York, NY, USA, 2005. ACM.

[CH07] Diane J. Cook and Jawrence B. Holder, editors. *Mining Graph Data*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2007.

[DBETT94] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. Algorithms for drawing graphs: An annotated bibliography. *Computational Geometry: Theory and Applications*, 4(5):235 – 282, 1994.

[Die06] Reinhard Diestel. *Graphentheorie*, volume 3. Springer, Heidelberg, September 2006.

[DL07] Walter Didimo and Guiseppe Liotta. *Mining Graph Data (Graph Visualization and Data Mining)*, chapter 3, pages 35 – 63. John Wiley & Sons, Inc., Hoboken, New Jersey, 2007.

[DRS04] T. Dweyer, H. Rolletschek, and F. Schreiber. Representing experimental biological data in metabolic networks. In *2nd Asia-Pacific Bioinformatics Conference (APBC'04)*, volume 29 of CRPIT, pages 13 – 20, Sydney, 2004. ACS.

[Ead84] P. Eades. A heuristic for graph drawing. *Congr. Numer.*, 42:149 – 160, 1984.

[Eri06] Thomas Erickson. From PIM to GIM: personal information management in group contexts. *Commun. ACM*, 49(1):74–75, 2006.

[ESTT97] Keith Edwards, Mike J. Spreitzer, Douglas B. Terry, and Marvin M. Theimer. Designing and Implementing Asynchronous Collaborative Applications with Bayou, 1997.

[GS86] Irene Greif and Sunil Sarin. Data sharing in group work. In *CSCW '86: Proceedings of the 1986 ACM conference on Computer-supported cooperative work*, pages 175–183, New York, NY, USA, 1986. ACM.

[HKV06] Heiko Haller, Felix Kugel, and Max Völkel. iMapping Wikis - Towards a Graphical Environment for Semantic Knowledge Management. In *SemWiki*, 2006.

[HMM00] I. Herman, G. Melancon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24 – 43, 2000.

[JFR$^+$01] Daniel Ford Joann, Daniel A. Ford, Joann Ruvolo, Stefan Edlund, Jussi Myllymaki, James Kaufman, Jared Jackson, and Martin Gerlach. Tempus Fugit: A system for making semantic connections, 2001.

[JT06] M. Juntumaa and V.K. Tuunainen. Pim applications - an explorative study on benefits and barriers. In *Proceedings of the 19th Bled Econference: Evalues*, Bled, Slovenia, 2006.

[LD89] K. K. Gordon Lan and David L. Demets. Group sequential procedures: Calendar versus information time. *Statistics in Medicine*, 8:1191–1198, 1989.

[Pal99] Leysia Palen. Social, individual and technological issues for groupware calendar systems. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 17–24, New York, NY, USA, 1999. ACM.

[Pay93] Stephen J. Payne. Understanding calendar use. *Hum.-Comput. Interact.*, 8(2):83–100, 1993.

[PG02] Leysia Palen and Jonathan Grudin. Discretionary Adoption of Group Support Software: Lessons from Calendar Applications. In *In B.E. Munkvold (Ed.), Organizational*, pages 159–179. Springer Verlag, 2002.

[Tam99] R. Tamassia. Advances in the theory and practice of graph drawing. *Theoretical Computer Science*, 17:235 – 254, 1999.

[TGW06] Martin Tomitsch, Thomas Grechenig, and Pia Wascher. Personal and private calendar interfaces support private patterns: diaries, relations, emotional expressions. In *NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction*, pages 401–404, New York, NY, USA, 2006. ACM.

[TS00] Albrecht Schmidt Telecooperation and Albrecht Schmidt. Implicit Human Computer Interaction Through Context. Technical report, Personal Technologies, 2000.

[vdVvW00] Gerrit van der Veer and Martijn van Welie. Task based groupware design: putting theory into practice. In *DIS '00: Proceedings of the 3rd conference on Designing interactive systems*, pages 326–337, New York, NY, USA, 2000. ACM.

Creative Knowledge Discovery by Literature Outlier Detection

Ingrid Petrič[1], Bojan Cestnik[2,3], Nada Lavrač[3,1], Tanja Urbančič[1,3]

[1] University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia
[2] Temida, d.o.o., Dunajska 51, 1000 Ljubljana, Slovenia
[3] Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
ingrid.petric@p-ng.si, bojan.cestnik@temida.si, nada.lavrac@ijs.si, tanja.urbancic@p-ng.si

Abstract. This paper investigates the role of outliers in literature-based knowledge discovery. It shows that detecting interesting outliers that appear in the literature about a given phenomenon can help generate novel plausible scientific hypotheses. The underlying assumption is that whereas the majority of domain literatures describe matters related to common understanding of the domain, some particular observations that appear rarely in the literature can indicate a promising direction towards novel discoveries. This rarity principle is used in our method called RaJoLink to guide the knowledge discovery process. The presented method focuses on the role of outliers in the closed discovery process as implemented in the RaJoLink literature mining methodology.

Keywords: rarity, outliers, bisociations, literature mining, knowledge discovery.

1 Introduction

In statistics, an outlier is an observation that is numerically distant from the rest of the data, or more formally, it is an observation that lies outside the overall pattern of a distribution [1]. While in many data sets outliers may be due to data measurement errors (therefore it would be best to discard them from the data), there are also several examples where outliers actually led to important discovery of intriguing information. Outlier mining has proved to have important applications in fraud detection and network intrusion detection [2], [3], [4]. Similarly, much attention to the study of outliers is paid in economics, particularly in finance and business, where rare events can be a sign of interesting unusual activities or observations like, for instance, potential sales opportunities [5]. A specifically challenging aspect of outlier detection is emerging within the climate research and extreme weather events prediction. There has been much interest in investigating the impacts, intensity and distribution of rare extreme events over a certain period of time [6]. Current attention to rare weather phenomena is driven by their possibility to become regionally more variable or

[1] Correspondence to: Ingrid Petrič, University of Nova Gorica, Vipavska 13, SI-5000 Nova Gorica, Slovenia. Fax number: +386 5 331 5240. E-mail address: ingrid.petric@p-ng.si

extreme menace to human life, civil infrastructure and natural ecosystems, which may have significant socioeconomic impacts [7]. Rarity as a principle has been extensively researched in the field of ecology statistics [8]. These investigations include the rarity driven by biodiversity and conservation policies [9]. Considering this, special concern of ecologists has been devoted to studying rare species [10]. They recognized two syndromes of rarity: habitat-limited species that were rare because their habitat was rare and dispersal-limited species that were rare because they stayed behind due to a catastrophic turnover of old growth. While ecologists' primary concern was preventing the extinction of rare species, they also identified the potential of dispersal-limited species to adapt to changed environment.

Data analysis with machine learning methods applied to large collections of textual data in databases enables discovering pieces of knowledge, which - when put together - might describe still unknown connections between phenomena. In this way, they contribute to the formation of new hypotheses in different fields. Connectivity of numerous large data sets that may include textual data and their computer-supported analysis contributes also in a methodological sense to the development of e-science. In particular, information that is related across different contexts is difficult to identify with the conventional associative approaches. It is, however, these kinds of context-crossing associations, called *bisociations* [11], that are often needed for innovative discoveries.

The aim of this paper is to present an approach to rarity-based knowledge discovery from text documents that can be used to explore implicit relationships across different domains of expertise. The approach upgrades the RaJoLink method [12] which provides a novel approach to the knowledge discovery from literature, based on the principle of rare terms from scientific articles together with the notion of bisociation. RaJoLink is intended to support experts in their overall process of open knowledge discovery, where hypotheses have to be generated, and in the closed knowledge discovery process, where hypotheses are tested. It was demonstrated in [12], [13], and [14] that this method can successfully support the user-guided knowledge discovery process.

The motivation for our focus on rare items/terms in the literature has grounds in the associationist creativity theory [15], and pays special attention to the category of context-crossing associations, called bisociations [11]. Bisociation implicates the literal processes of mind when making completely new connections between concepts from contexts or categories that are usually considered separate contexts or categories. Mednick [15] defines creative thinking as the faculty of generating new combinations of distant associative elements (e.g. words). He explicates how thinking of concepts that are not strictly related to the elements under research inspires unforeseen useful connections between elements. In this manner, bisociations considerably improve the creative process. Through the history of science, this mechanism has been the crucial element of progressive insights and paradigm shifts. Nevertheless, no comprehensive ICT methodology has yet been developed on this basis. Hence, we firmly believe that our method contributes to this particular approach to scientific discovery, which is based on an existing, but hitherto not computationally implemented notion of bisociation.

This paper describes an upgrade of the RaJoLink methodology, focusing on rare terms in the literature from different domains/contexts, aimed at creative knowledge discovery through bisociative reasoning. The detected rare terms may indicate the discovery of the so-called bridging terms/concepts, enabling the exploration of the potentially interesting bisociative links between the domains, which may be indicative of new insights/discoveries. The methodology has been applied to a challenging medical domain: the set of records for our studies was selected from the domain of autism. Autism belongs to a group of pervasive developmental disorders that are portrayed by an early delay and abnormal development of cognitive, communication and social interaction skills of a person [16]. It is a very complex and not yet sufficiently understood domain, where precise causes are still unknown, hence we have chosen it as our experimental testing domain.

This paper is organized as follows. Section 2 presents the related work in the area of literature mining. Section 3 introduces the literature-based knowledge discovery process and further explores rarity as a principle for guiding the knowledge discovery in the upgraded RaJoLink method. Section 4 presents the RaJoLink approach by focusing on outliers in the closed discovery process. Section 5 illustrates the application of outlier detection to the autism literature. Section 6 provides discussion and conclusions.

2 Related Work in Literature Mining

Novel interesting connections between disparate research findings can be extracted from the published literature. Analysis of implicit associations hidden in scientific literature can guide the hypotheses formulation and lead to the discovery of new knowledge. To support such literature-based discoveries in medical domains, Swanson has designed the ABC model approach [17] that investigates whether an agent *A* influences a phenomenon *C* by discovering complementary structures via interconnecting phenomena *B*. Two literatures are complementary if one discusses the relations between *A* and *B*, while a disparate literature investigates the relations between *B* and *C*. If combining these relations suggests a previously unknown meaningful relation between *A* and *C*, this can be viewed as a new piece of knowledge that might contribute to a better understanding of phenomenon *C*.

Weeber and colleagues [18] defined the hypothesis generation approach as an open discovery process and the hypothesis testing as a closed discovery process. In the *open discovery process* only the phenomenon under investigation (*C*) is given in advance, while the target agent *A* is still to be discovered. In the *closed discovery process*, both *C* and *A* are known and the goal is to search for linking phenomena *B* in order to support the validation of the hypothesis about the connection between *A* and *C*. Smalheiser and Swanson [19] developed an online system named ARROWSMITH, which takes as input two sets of titles from disjoint domains *A* and *C* and lists terms *b* that are common to literature *A* and *C*; the resulting terms *b* are used to generate novel scientific hypotheses.[2] As stated by Swanson [20], his major

[2] Here we use the notations *A*, *B*, and *C* that are written in uppercase letter symbols to represent a set of terms (e.g., literature or collection of records), while with *a*, *b*, and *c* (lowercase symbols) we represent a single term.

focus in literature-based discovery has been on the closed discovery process, where both *A* and *C* have to be specified in advance.

Several researchers have continued Swanson's line of research. An on-line literature-based discovery tool called BITOLA has been designed by Hristovski [21]. It uses association rule mining techniques to find implicit relations between biomedical terms. Weeber and colleagues [22] developed Literaby, the concept-based Natural Language Processing tool. The units of analysis that are essential for their approach are UMLS Metathesaurus concepts. The open discovery approach developed by Srinivasan and colleagues [23], on the other hand, relies almost completely on Medical Subject Headings (MeSH). Yetisgen-Yildiz and Pratt [24] proposed a literature-based discovery system called LitLinker. It mines biomedical literature by employing knowledge-based and statistical methods. All the pointed systems use MeSH descriptors [25] as a representation of scientific medical documents, instead of using title, abstract or full-text words. Thus, problems arise since MeSH indexers normally use only the most specific vocabulary to describe the topic discussed in a document [25] and therefore some significant terminology from the documents' content may not be covered. The Swanson's literature-based discovery approach has been extended also by Lindsay and Gordon [26], who used lexical statistics to determine relative frequencies of words and phrases. In their open discovery approach they search for words on the top of the list ranked by these statistics. However, their approach fails when applied to Swanson's first discoveries and extensive analysis has to be based on human knowledge and judgment.

3 The Upgraded RaJoLink Knowledge Discovery Process

The aim of knowledge discovery investigated in this paper is to detect the previously unnoticed concepts (chances) at the intersections of multiple meaningful scenarios. As a consequence, tools for indicating rare events or situations prove to play a significant role in the process of research and discovery [27]. From this perspective, researchers have to be sensitive to curious or rare observations of phenomena in order to provide novel possible opportunities for reasoning [28] and be aware of the powerful support that data mining tools can have for choosing meaningful scenarios [27].

Outliers actually attract a lot of attention in the research world and are becoming increasingly popular in text mining applications as well. Detecting interesting outliers that rarely appear in a text collection can be viewed as searching for the needles in the haystack. This popular phrase illustrates the problem with rarity since identifying useful rare objects is by itself a difficult task [28].

The rarity principle that we apply in the RaJoLink literature-based discovery is a fundamental difference from the previously proposed methods and represents a unique contribution of the RaJoLink method. In our earlier work [12], [13], and [14] we presented the idea of extending the Swanson's ABC model to handle the open discovery process with rare terms from the domain literature. For that purpose we employed the Txt2Bow utility from the TextGarden library [29] in order to compute total frequencies of terms in the entire text corpus/corpora.

Already in the original RaJoLink method the rarity principle is employed as a means to find new interesting pieces of knowledge that were previously unrelated in the available literature. The rationale behind it is that if a piece of information is abundant in the set of articles, it may be speculated that its impact to the field under study is well-covered; however, if it appears rarely, not many researchers are acquainted with it, so it might be worth exploring it further. Similarly to dispersal-limited species from ecology, such pieces of information might be either on their way to extinction or might embody a potential for new development in the field. In order to distinguish between the two options, expert guidance is needed in the process.

To support the extraction of information from scientific articles and to simplify the processing and analysis of such information we have designed the RaJoLink method. RaJoLink provides a framework for literature-based discovery from texts written in English. The entire RaJoLink method involves three principal steps, *Ra*, *Jo* and *Link*, which have been named after the key elements of each step: Rare terms, Joint terms and Linking terms. Note that the steps *Ra* and *Jo* implement the open discovery, while the step *Link* corresponds to the closed discovery. The methodological description of the three steps has been provided in our previous publications [12], [13], and [14].

We developed a software tool that implements the RaJoLink method and provides decision support to experts. It can be used to find scientific articles in MEDLINE database [30], to compute statistics about the data, and to analyze them to discover eventually new knowledge. By such exploration, massive amounts of textual data are automatically collected from databases, and text mining methods are employed to generate and test hypotheses. In the step *Ra*, a specified number (set by user as a parameter value) of interesting rare terms in literature about the phenomenon *C* under investigation are identified. In the step *Jo*, all available articles about the selected rare terms are inspected and interesting joint terms that appear in the intersection of the literatures about rare terms are identified and selected as the candidates for *A*. In order to provide explanation for hypotheses generated in the step Jo, our method searches for links between the literature on joint term *a* and the literature on term *c*.

The upgraded RaJoLink methodology for creative knowledge discovery consists of the following steps.

- The crucial step in the RaJoLink method is to identify rare elements within scientific literature, i.e., terms that rarely appear in articles about a certain phenomenon.
- Sets of literature about rare terms are then identified and considered together to formulate one or more initial hypotheses in the open discovery process.
- Next, in the closed discovery process, RaJoLink focuses on outlying and their neighbouring documents in the documents' similarity graphs. We construct such graphs with the computational support of a semi-automatic tool for ontology construction, called OntoGen [31].
- Outlying documents are then used as a heuristic guidance to speed-up the search for the linking terms (bridges) between different domains of expertise and to alleviate the burden from the expert in the process of hypothesis testing.

4 Outlier Detection in the RaJoLink Knowledge Discovery Process

This paper focuses mainly on the steps of the closed discovery process. The closed discovery process is supported by using the OntoGen tool [31]. One of its features is its capacity of visualizing the similarity between the selected documents of interest. The main novelty of the upgraded RaJoLink methodology is to visualize outlying (and their neighbouring) documents in the documents' similarity graph (Figure 1) to find bisociations in the combined set of literatures *A* and *C*. Our argumentation is that outlying documents of two implicitly linked subjects can be used to search for relevant linking terms (bridges) between the two subjects. The idea of this paper of representing instances of literature *A* together with instances of literature *C* in the same similarity graph with the purpose of searching for their bisociative links is a unique aspect of our method in comparison to the literature-based discovery investigated by other researchers.

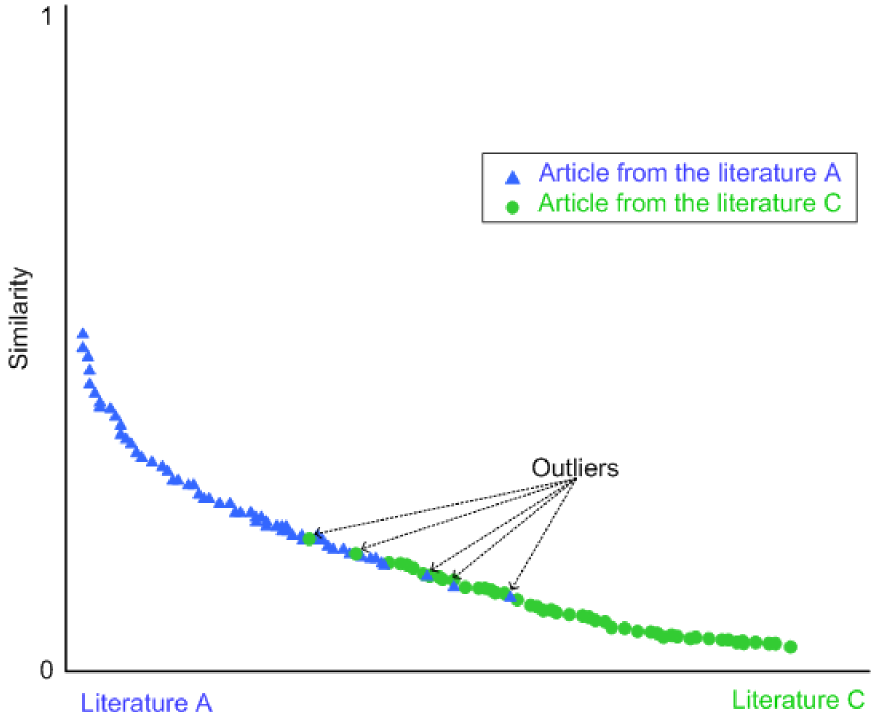


Fig. 1. A graph representing instances of literature *A* and instances of literature *C* according to their content similarity. Outliers are positioned far enough away from the most typical representatives of the two heretofore unrelated literatures *A* and *C*.

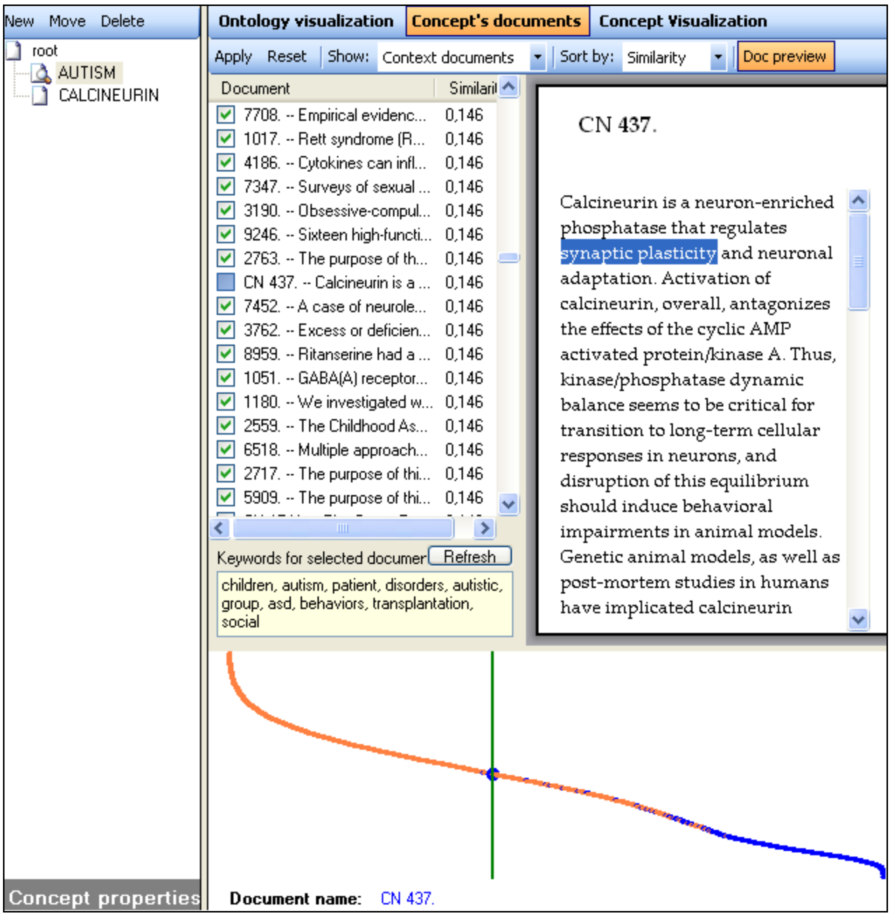
In the closed discovery process of the RaJoLink method, linking terms *b* that bridge the literature *A* and the literature *C* can be considered as outliers. Having disparate literatures *A* and *C*, both domains are examined by the combined dataset of literatures *A* and *C* in order to assess whether they can be connected by implicit relations. Within the whole corpus of texts consisting of literatures *A* and *C*, which acts as input for step Link (i.e. the closed discovery) of RaJoLink, each text document represents a single record.

Each document from the two literatures is represented by a set of words using frequency statistics based on Bag of Words (BoW) text representation [32]. BoW representation and the appearance of co-occurring words are employed as a measure of content similarity between documents. Its computation is performed with OntoGen, which was designed for interactive data-driven construction of topic ontologies [31]. The content similarity is based on the textual description of documents and is measured using the standard TF*IDF (term frequency inverse document frequency) weighting method [33]. This way, all the records are sorted according to similarity and the content related documents are obtained by comparing neighbouring documents from the list. The similarity between documents is visualized with OntoGen in the document's similarity graph, as illustrated in Figure 1.

5 Application of Outlier Detection in the Autism Literature

Figure 2 shows the similarity graph representing instances of literature *A* and instances of literature *C (AUTISM context)* according to their content similarity, where *A* denotes a set of documents containing term calcineurin *(i.e., the so-called CALCINEURIN context*), and *C* denotes a set of documents containing term autism (i.e., the so-called *AUTISM context*).

The presented linking approach suggests a novel way to improve the evidence gathering phase when analyzing individual *a* terms in their potential connection with the term *c*. In fact, even Srinivasan and colleagues, who declared to have developed the algorithms that require the least amount of manual work in comparison with other studies [23], still need significant time and human effort for collecting evidence relevant to the hypothesized connections. In the comparable upgraded RaJoLink approach, the domain expert should be involved only in the conclusive actions of the Link step to accelerate the choice of significant linking terms. In this step, similarity graph visualization proves to be extremely beneficial for speeding the process of discovering the bridging concepts. Not only that the documents detected as outliers are visualized and their contents presented on the screen by simply clicking on the pixel representing the document (see Figure 2), but also the keywords are listed, explicitly indicating a set of potential bridging concepts (terms) to be explored by the domain experts.



Document	Similarit
☑ 7708. -- Empirical evidenc...	0,146
☑ 1017. -- Rett syndrome (R...	0,146
☑ 4186. -- Cytokines can infl...	0,146
☑ 7347. -- Surveys of sexual ...	0,146
☑ 3190. -- Obsessive-compul...	0,146
☑ 9246. -- Sixteen high-functi...	0,146
☑ 2763. -- The purpose of th...	0,146
□ CN 437. -- Calcineurin is a ...	0,146
☑ 7452. -- A case of neurole...	0,146
☑ 3762. -- Excess or deficien...	0,146
☑ 8959. -- Ritanserine had a ...	0,146
☑ 1051. -- GABA(A) receptor...	0,146
☑ 1180. -- We investigated w...	0,146
☑ 2559. -- The Childhood As...	0,146
☑ 6518. -- Multiple approach...	0,146
☑ 2717. -- The purpose of thi...	0,146
☑ 5909. -- The purpose of thi...	0,146

Keywords for selected documer

children, autism, patient, disorders, autistic, group, asd, behaviors, transplantation, social

CN 437.

Calcineurin is a neuron-enriched phosphatase that regulates synaptic plasticity and neuronal adaptation. Activation of calcineurin, overall, antagonizes the effects of the cyclic AMP activated protein/kinase A. Thus, kinase/phosphatase dynamic balance seems to be critical for transition to long-term cellular responses in neurons, and disruption of this equilibrium should induce behavioral impairments in animal models. Genetic animal models, as well as post-mortem studies in humans have implicated calcineurin

Document name: CN 437.

Fig. 2. OntoGen's similarity graph representing instances of the literature *A (CALCINEURIN context)* and instances of the literature *C (AUTISM context)* according to their content similarity. The distinctive article about the substance calcineurin (*CN 437)* is visualized among the autism context documents.

6 Conclusions

Current literature-based approaches depend strictly on simple, associative information search. Commonly, literature-based association is computed using measures of similarity or co-occurrence. Because of their 'hard-wired' underlying criteria of co-occurrence or similarity, these methods often fail to discover relevant information, which is not related in obvious associative ways. Especially information related across separate contexts is hard to identify with the conventional associative approach. In such cases the context-crossing connections, called bisociations, can help generate creative and innovative discoveries. The RaJoLink method has the potential for bisociative relation discovery as it allows switching between contexts by exploring rare observations in intersections between contexts.

Similar to Swanson's closed discovery approach [17], the search for linking terms consists of looking for terms *b* that can be found in separate sets of records, namely in the literature *A* as well as in the literature *C*. However, our focusing is on outliers from the two sets of records and their neighbouring documents. Thus we show how outlying documents in the similarity graphs yield useful information in the closed discovery, where connections have to be found between the literatures *A* and *C*. In fact, such visual analysis can show direction to the previously unseen relations, which provide new knowledge. This is an important aspect and significant contribution of our method to literature-based discovery research.

Most of the data analysis research is focused on discovering mainstream relations. These relations are well statistically supported; findings usually confirm the conjectured hypothesis. However, searching for rare items can be beneficial to finding new, previously unseen relations. This paper provides insight into the relationship between outliers and the literature-based knowledge discovery. The focus is on the use of the associationist creativity theory in the literature-based discovery with particular attention to the context-crossing associations, called bisociations. An important feature of our approach is the way of detecting the bridging concepts connecting unrelated literatures, which we have performed by the OntoGen's similarity graphs. We used them for representing instances of the literature *A* together with instances of the literature *C* according to their content similarity with the goal to find out outliers from the two sets of literatures and their neighbouring documents. We showed that with the similarity graphs that enable the visual analysis of the literature it is easier to detect the documents, which are very interesting for a particular link analysis investigation, for the reason that such outlying documents often represent particularities in domain literature. Therefore, to test whether the hypothetical observation could be related to the phenomenon under investigation or not, we compare the sets of literature about the initial phenomenon with the literature about the hypothetically related one in the documents' similarity graphs. By our original discovery of linking terms between the literature on autism and the literature on calcineurin we proved that such combination of two previously unconnected sets of literatures in a single content similarity graph can be very effective and useful [12], [13], and [14]. From the similarity graphs that we drew with OntoGen we could quickly notice, which documents from the observed domain are semantically more related to another context. They were positioned in the middle portions of the similarity curves.

Acknowledgments. This work was partially supported by the project Knowledge Technologies (grant P2-0103) funded by the Slovene National Research Agency, and the EU project FP7-211898 BISON (Bisociation Network for Creative Information Discovery).

References

1. Moore, D. S., McCabe, G. P.: Introduction to the Practice of Statistics, 3rd ed. New York: W. H. Freeman (1999)
2. Aggarwal, C. C., Yu, P. S.: An effective and efficient algorithm for high-dimensional outlier detection. Int J Very Large Data Bases 14(2), pp. 211--221 (2005)
3. Lazarevic, A., Kumar, V., Srivastava, J.: Intrusion detection: A survey. In: Kumar, V., Srivastava, J., and Lazarevic, A. (eds.) Massive Computing, Managing Cyber Threats. Springer, US, pp. 19--80 (2005)
4. Singhal, A., Jajodia, S.: Data warehousing and data mining techniques for intrusion detection systems. Distrib Parallel Databases 20(2), pp. 149--166 (2006)
5. Leung, C. K.-S., Thulasiram, R. K., Bondarenko, D. A.: An Efficient System for Detecting Outliers from Financial Time Series. In: Bell D and Hong J (eds.) Flexible and Efficient Information Handling. Lect Notes Comput Sci 4042, Springer, Berlin, pp. 190--198 (2006)
6. IPCC Climate change 2007: The physical science basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change. 996 pp (2007)
7. Frei, C., Schär, C.: Detection probability of trends in rare events: Theory and application to heavy precipitation in the Alpine region. J Clim 14(7), pp 1568--1584 (2001)
8. Ellison, A. M., Agrawal, A. A.: The Statistics of Rarity. Ecology 86(5), pp. 1079--1080 (2005)
9. Carney, R. S.: Basing conservation policies for the deep-sea floor on current-diversity concepts: A consideration of rarity. Biodiversity and Conservation 6(11), pp. 1463--1485 (1997)
10. Boughton, D.: Paradoxes in science: A new view of rarity. Science findings of Pacific Northwest Research Station 35, (2001)
11. Koestler, A. The act of creation. MacMillan Company, New York (1964)
12. Petrič, I., Urbančič, T., Cestnik, B., Macedoni-Lukšič, M.: Literature mining method RaJoLink for uncovering relations between biomedical concepts. J. Biomed. Inform. 42(2), pp. 219--227 (2009)
13. Petrič, I., Urbančič, T., Cestnik, B.: Discovering hidden knowledge from biomedical literature. Informatica 31(1), pp. 15--20 (2007)
14. Urbančič, T., Petrič, I., Cestnik, B., Macedoni-Lukšič, M.: Literature mining: towards better understanding of autism. In: Bellazzi, R., Abu-Hanna, A., Hunter, J. (eds.) AIME 2007. Proceedings of the 11th Conference on Artificial Intelligence in Medicine in Europe. pp. 217--226. Amsterdam, The Netherlands (2007)
15. Mednick, S. A.: The associative basis of the creative process. Psychol. Rev. 69(3), pp. 220--232 (1962)
16. American Psychiatric Association: Diagnostic and Statistical Manual of Mental Disorders, 4th Edition, Text Revision. Washington, DC (2000)
17. Swanson, D. R.: Undiscovered public knowledge. Library Quarterly 56(2), pp. 103--118 (1986)
18. Weeber, M., Vos, R., Klein, H., de Jong-van den Berg, L. T. W.: Using concepts in literature-based discovery: Simulating Swanson's Raynaud–fish oil and migraine–magnesium discoveries. J. Am. Soc. Inf. Sci. Tech. 52(7), pp. 548--557 (2001)
19. Smalheiser, N. R., Swanson, D. R.: Using ARROWSMITH: a computer-assisted approach to formulating and assessing scientific hypotheses. Comput. Methods Programs Biomed. 57(3), pp. 149--153 (1998)
20. Swanson, D. R., Smalheiser, N. R., Torvik, V. I.: Ranking indirect connections in literature-based discovery: The role of Medical Subject Headings (MeSH). J. Am. Soc. Inf. Sci. Tech. 57(11), pp. 1427--1439 (2006)

21. Hristovski, D., Peterlin, B., Mitchell, J. A., Humphrey, S. M.: Using literature-based discovery to identify disease candidate genes. Int. J. Med. Inform. 74(2-4), pp. 289--298 (2005)
22. Weeber, M.: Drug discovery as an example of literature-based discovery. In: Džeroski, S., Todorovski, L. (eds.) Computational Discovery of Scientific Knowledge. LNCS 4660, pp. 290--306 Springer, Berlin, Heidelberg (2007)
23. Srinivasan, P., Libbus, B.: Mining MEDLINE for implicit links between dietary substances and diseases. Bioinformatics 20(Suppl 1), pp. I290--296 (2004)
24. Yetisgen-Yildiz, M., Pratt, W.: Using statistical and knowledge-based approaches for literature-based discovery. J. Biomed. Inform. 39(6), pp. 600--611 (2006)
25. Nelson, S. J., Johnston, D., Humphreys, B. L.: Relationships in Medical Subject Headings. In: Bean, C. A., Green, R. (eds.) Relationships in the organization of knowledge. pp. 171--184. Kluwer Academic Publishers, New York (2001)
26. Lindsay, R. K., Gordon, M. D.: Literature-based discovery by lexical statistics. J. Am. Soc. Inf. Sci. 50(7), pp. 574--587 (1999)
27. Ohsawa, Y.: Chance discovery: the current states of art. Chance Discoveries in Real World Decision Making 30, pp. 3--20 (2006)
28. Magnani, L.: Chance discovery and the disembodiment of mind. In: Khosla, R., Howlett, R. J., Jain L. C. (eds.) Knowledge-Based Intelligent Information and Engineering Systems: 9th International Conference, KES 2005. pp. 547--553 Melbourne, Australia (2005)
29. Grobelnik, M., Mladenić, D.: Extracting human expertise from existing ontologies. In: EU-IST Project IST-2003-506826 SEKT (2004)
30. MEDLINE Fact Sheet [Online], URL:http://www.nlm.nih.gov/pubs/factsheets/medline.html
31. Fortuna, B., Grobelnik, M., Mladenić, D.: Semi-automatic data-driven ontology construction system. In: Bohanec, M., Gams, M., Rajkovič, V., Urbančič, T., Bernik, M., Mladenić, D., Grobelnik, M., Heričko, M., Kordeš, U., Markič, O., Musek, J., Osredkar, M. J., Kononenko, I., Novak Škarja, B. (eds.) Proceedings of the 9th International multi-conference Information Society. pp. 223--226. Ljubljana, Slovenia (2006)
32. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34(1), pp. 1--47 (2002)
33. Salton, G., Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24(5), pp. 513--523 (1988)

Characterizing Semantic Relatedness of Search Query Terms

Dominik Benz, Beate Krause, G. Praveen Kumar,
Andreas Hotho, and Gerd Stumme

Knowledge & Data Engineering Group, University of Kassel,
34121 Kassel, Germany
{benz,krause,praveen,hotho,stumme}@cs.uni-kassel.de

Abstract. Mining for semantic information in search engine query logs bears great potential for both the optimization of search engines and bootstrapping Semantic Web applications. The interaction of a user with a search engine (more specifically clicklog information) has recently been viewed as *implicit tagging* of resources by query terms. The resulting structure – previously called a *logsonomy* – exhibits structural similarities to folksonomies, which evolve during the *expclicit* process of annotating resources with freely chosen keywords in social bookmarking systems. For the folksonomy case, appropriate measures of relatedness have shown to be capable to harvest the emerging semantics inherent in the tripartite graph of users, tags and resources. Motivated by the reported structural similarities, in this work we extend this methodology to logsonomies. More specifically, we apply several measures of query term relatedness to the logsonomy graph and provide a semantic characterization for each measure by grounding it against user-validated relatedness measures based on WordNet. Comparing the outcome with prior results of analyzing folksonomy data we find that the formalization of log data in logsonomies retains the semantic information. Some relatedness measures we applied prove to be able to capture these emergent semantics similarly to the folksonomy case, while others exhibit different characteristics. In this way we provide a novel and systematic approach to compare the emergent semantics of user interactions with search engines and social bookmarking systems. We conclude that the *type* of semantic information inherent in both emerging structures is similar, and inform the choice of an appropriate measure of query term relatedness for a given task.

1 Introduction

Folksonomies are complex systems consisting of user-defined labels added to web content such as bookmarks, videos or photographs by different users. In contrast to classical search engines, which index the web and offer a simple user interface to search in this index, a folksonomy can be explored in different dimensions taking users, tags and resources into account. With logfiles containing queries and clicks of search engine users, a similar relation between users, query terms and a resource can be found: a user submits a query and clicks on a specific URL. The resulting structure of this process, previously called *logsonomy* [1], is a tripartite graph of a set of users, queries and clicked URLs with hyperedges, each connecting one query, one clicked URL and one specific user.

While folksonomies aggregate *explicit* lightweight metadata annotations (namely tags), logsonomies can hence be seen as *implicit* annotations by user clicks. Previous work [1] revealed that both show similar structural characteristics, e. g., small world properties, a power law distribution of tags and users, and a similar co–occurrence behaviour of tags. These first insights indicate the possibility that logsonomies contain – similar to the folksonomy graph – inherent semantics emerging from the "collaborative" process of searching similar information and being interested in the same resources.

In prior work, we found that measures of semantic relatedness based on the folksonomy graph are able to extract these *emerging semantics* [2]. Motivated by the structural analogies mentioned above, we explore in this paper the potential of logsonomies to extract semantic relations between different queries or query parts. In [2], different relatedness measures considering statistical, distributional and structural characteristics of a folksonomy have been applied to learn about related tags on a large-scale snapshot of the social bookmarking system del.icio.us. These measures are the co-occurrence count, three distributional measures which use the cosine similarity in the vector spaces spanned by users, tags and resources, respectively, and FolkRank [3], a graph-based measure which is an adaptation of the well-known PageRank [4] to folksonomies.

We will apply these measures in the same manner to a logsonomy built from an AOL click dataset. This allows for a direct comparison of the findings of tag relatedness in folksonomies to the ones in logsonomies. Especially, the semantic grounding based on a comparison of related tags / query terms to semantic relations of terms in the lexical database WordNet helps to characterize the major differences of relatedness measures between folk- and logsonomies. We follow the choice of [2] and measure the semantic term relatedness within WordNet by using both the taxonomic path length and a similarity measure by Jiang and Conrath [5]. The latter resembles most closely to what humans perceive as semantically related [6], while the first allows the inspection of the edge composition of paths leading from one tag to the corresponding related tags, which has proven to be especially insightful.

Learning about the hidden structure of a search engine's query vocabulary will be interesting for a variety of applications: similar or related tags can be used to refine and expand search queries, to correct spelling errors or to improve a search engine's ranking. For example, first results show that the tag context relatedness is often able to extract synonyms of a given query term. Other measures (like FolkRank) seem to point to more general terms, which can be useful for broadening a search. Another application of our work is harvesting the usage-driven semantics of search query logs by ontology learning procedures based on logsonomies. This would directly tackle the knowledge acquisition problem of many Semantic Web applications. Our work establishes hereby the connection between prior work (like [7] on mining for semantics in search query logs) and recent approaches of bridging the gap between the "Web 2.0" and the Semantic Web.

In this paper, we bring together two research branches: First, work targeted on harvesting semantic information from social annotations (i. e., by appropriate measures of semantic relatedness), and second work on structural similarities between interactions of users with social bookmarking systems and search engines. We expect synergies for both directions by posing the following research questions:

- Is there evidence for emergent semantics in logsonomies as it is in folksonomies?
- Can we infer further structural similarities or differences between folksonomies and logsonomies by comparing the output of relatedness measures on both structures?
- Does a given measure of relatedness exhibit the same semantic characteristics when applied to a folksonomy and to a logsonomy graph?

The rest of the paper is organised as follows. Section 3 briefly defines folk- and logsonomies, describes the construction of logsonomies and the datasets used for computing tag and query term relatedness. In Section 4, we introduce the applied relatedness measures. Some qualitative insights are presented in Section 5, while a thorough analysis of query term relatedness is conducted in Section 6. Finally, implications of this work to other fields are discussed and an outlook on future work given.

2 Related Work

In this section, we discuss related work which considers the analysis of semantics in query click logs. To the best of our knowledge, a comprehensive comparison of the semantics extracted from folksonomies and search engine logs has not been conducted before, but each data structure has been considered individually.

Besides a variety of analytical studies about the nature of clickdata, extensive reseach has been conducted in the area of information retrieval where click data was used to expand queries and to improve the retrieval performance. For a detailed discussion of folksonomies, social bookmarking systems and the extraction of semantics the reader may refer to [2].

The transformation of clickdata to a tripartite hypergraph as well as a comparison to folksonomy properties has been carried out in [1]. The work could show similar structural properties of logsonomies and folksonomies. Given these results, the analysis of query term similarity seems to be very promising.

A further consideration of the tripartite structure of query logs has been presented in [8], where an algorithm to rank resources based on the relationships among users, queries and resources was proposed. In [7], Baeza-Yates and Tiberi proposed to present query-logs as an implicit folksonomy where queries can be seen as tags associated to documents clicked by people making those queries. The authors extracted semantic relations between queries from a query-click bipartite graph where nodes are queries and an edge between nodes exists when at least one equal URL has been clicked after submitting the queries. As an extension of the above work, Francisco et al. [9] cluster these bipartite graphs using clique percolation and priori induced cliques and consequently extract semantic relations between queries. However, the semantic grounding of the relations is not in the core of this work. By constructing a folksonomy-alike structure, we can build on systematic investigations of various topological characteristics of the well-established folksonomy model. Our tag-tag–co-occurrence analysis is closely related to the graph analysis of [9], but operates on a different kind of dataset created by splitting the original search queries into single search terms. This dataset is compared to results of the del.icio.us folksonomy. Overall, our contribution is a comparison between hypergraphs constructed of real-world folksonomy and logsonomy datasets. We are not aware of other work which examines differences and commonalities of user interactions with folksonomy and search engine systems as we do in this paper.

3 Folksonomies and Logsonomies

As mentioned above, social bookmarking systems contain *explicit* annotations while search engine clicklogs provide *implicit* annotations of resources. In this section, we provide a formal model of folksonomies and show how search engine clicklogs can be adapted to this model, resulting in *logsonomies*. We consequently detail on the datasets used to evaluate our approach.

3.1 Formal Model of a Folksonomy

The central data structure of a social bookmarking system is called *folksonomy*. It can be seen as a lightweight classification structure which is built from *tag* annotations (i. e., freely chosen keywords) added by different users to their resources. A folksonomy consists thus of a set of users, a set of tags, and a set of resources, together with a ternary relation between them.

Following [3], we formally define a *folksonomy* as a tuple $\mathbb{F} := (U, T, R, Y)$ where

- U, T, and R are finite sets, whose elements are called *users*, *tags* and *resources*, resp., and
- Y is a ternary relation between them, i. e., $Y \subseteq U \times T \times R$, whose elements are called tag assignments (TAS for short).

For convenience we also define, for all $u \in U$ and $r \in R$, $\textsc{tags}(u, r) := \{t \in T \mid (u, t, r) \in Y\}$, i. e., $\textsc{tags}(u, r)$ is the set of all tags that user u has assigned to resource r. The set of all *posts* of the folksonomy is $P := \{(u, S, r) \mid u \in U, r \in R, S = \textsc{tags}(u, r), S \neq \emptyset\}$. Thus, each *post* consists of a user, a resource and all tags that the user has assigned to the resource.

3.2 Adaptation to Search Engine Query Logs: Logsonomies

In order to apply our established folksonomy analysis techniques [2] to search engine logs, we need similar structures on both sides. We adhere to the approach described in [1] and transform a search engine log into a folksonomy alike structure, called *logsonomy*. User IDs represent the users of a folksonomy,[1] and the *clicked* URLs represent resources. The latter are implicitly annotated by the given

[1] In datasets other than the AOL data at hand, one may need to switch to session IDs, if there are no explicit user IDs in the log files.

query; in order to mimic most closely social annotations, we split composed queries into single words. These *query terms* are thus all substrings of a query that are separated by whitespaces. They correspond to the tags in a folksonomy. For sake of simplicity, we will also use the term “tag” when addressing “query words” in the remainder of the paper. This means when we talk about relatedness of “tags” in a logsonomy, we do talk about relatedness of query words. The decision to use the splitted queries instead of complete ones was motivated by the findings of [1] that the resulting network structure comes closer to an actual folksonomy.

More formally, this transformation of a search engine log to a logsonomy can be described as follows:

- Let U be the set of *users* of the search engine.
- T be the set of *query terms* contained in the queries the users gave to the search engine,
- R be the set of *URLs* which have been clicked by the search engine users.

We add a tuple (u, t, r) to Y whenever user u clicked on resource r of a result set after having submitted the query term t (eventually with other terms). The resulting relation $Y \subseteq U \times T \times R$ corresponds to the tag assignments in a folksonomy.

The process of creating a logsonomy shows similarities to the creation of a folksonomy. Users describe an information need by means of a query. They then restrict the result set of the search engine by clicking on those URLs whose snippets indicate that the web page has some relation to the query. These query/click combinations result in the logsonomy. However, one needs to keep some major differences in mind, when applying folksonomy techniques to logsonomies:

- Users have a bias towards clicking the top URLs of a result list. In query log analysis, these clicks are usually discounted.
- While tagging a specific resource can be seen as an indicator for relevance, users may click on a resource to check if the result is important and then disappointedly return to the initial search list. We nevertheless assume that the act of clicking already indicates an association between query and resource in the logsonomy, since the log data under study did not contain any explicit user feedback (which could have been used for further differentiation).
- In logsonomies, we interpret the query as the description of the underlying, clicked resource. Splitting these descriptions in single words may destroy or change the intended meaning.
- Queries are processed by search engines which do not publish the techniques applied. One does not know to which extent the query terms serve as a description of search results. They may be ignored or enhanced with similar query terms.
- When a resource never comes up in a search result, it cannot be tagged as such.

3.3 Datasets

In order to make our results comparable to prior work on semantic relatedness measures on folksonomies, we detail here on the social bookmarking data which was

Table 1. Folksonomy and Logsonomy Datasets

dataset	$\|T\|$	$\|U\|$	$\|R\|$	$\|Y\|$
Del.icio.us	10,000	476,378	12,660,470	101,491,722
AOL split queries	10,000	463,380	1,284,724	26,227,550

the basis of [2] before describing the click data we used to build the logsonomies. Table 1 summarizes some statistics about both datasets.

Social Bookmarking Data. For our experiments, we used data from the social bookmarking system del.icio.us, collected in November 2006. In total, data from $667,128$ users of the del.icio.us community were collected, comprising $2,454,546$ tags, $18,782,132$ resources, and $140,333,714$ tag assignments. As one main focus of this work is to characterize tags by their properties of co–occurrence with other tags, we restricted our dataset to the $10,000$ most frequent tags of del.icio.us, and to the resources/users that have been associated with at least one of those tags. One could argue that tags with low frequency have a higher information content in principle — but their inherent sparseness makes them less useful for the study of both co-occurrence and distributional measures. The size of the restricted folksonomy is shown in Table 1.

Click Data. We used a click dataset from the AOL search engine. The data was collected from March, 1st to May, 31st 2006. The original dataset consists of 657,426 unique user IDs, 10,154,742 unique queries, and 19,442,629 click-through events [10]. We constructed the logsonomy as described in Section 3.2. Again, we apply the restriction of only using the 10,000 most frequent tags (i. e., query words) to the dataset. The resulting sizes are shown in Table 1. Since the AOL data was only available with truncated URLs, we reduced the URLs to host-only URLs, i. e., we removed the path of each URL leaving only the host name.

4 Measures of Relatedness

The underlying structure of a logsonomy can — analogously to a folksonomy — also be regarded as an undirected tri-partite hyper-graph (see section 3.2). As measures of similarity and relatedness are not well-developed for this kind of data yet, we follow the approach of [2] and stick to two- and one-mode views on the data. These views are complemented by a graph-based approach for discovering related tags (FolkRank), which makes direct use of the three-mode structure. Please note that the remaining paragraphs of this section summarize the measures of relatedness used in our prior work [2]; we include their description in order to explain their adaption to logsonomies. For a detailed description of the computational complexity of each measure, we refer to [2].

Co-Occurrence. Given a logsonomy (U, T, R, Y), we define the *query word co-occurrence graph* as a weighted undirected graph whose set of vertices is the set T of query words. Two query words t_1 and t_2 are connected by an edge, iff there is at least one query (u, T_{ur}, r) with $t_1, t_2 \in T_{ur}$. The *weight* of this edge is given by the number of queries that contain both t_1 and t_2, i. e.,

$$w(t_1, t_2) := \mathrm{card}\{(u, r) \in U \times R \mid t_1, t_2 \in T_{ur}\} \ . \tag{1}$$

Co-occurrence relatedness between query words is given directly by the edge weights. For a given query word $t \in T$, the tags that are most related to it are thus all the tags $t' \in T$ with $t' \neq t$ such that $w(t, t')$ is maximal. We will denote this co-occurrence relatedness by *co-occ*.

Distributional Measures. We introduce three distributional measures of query word relatedness that are based on three different vector space representations of query words. The difference between the representations – and thus between the measures – is the feature space used to describe the tags, which varies over the possible three dimensions of the logsonomy. Specifically, for $X \in \{U, T, R\}$ we consider the vector space $\mathbb{R}^X$, where each query word t is represented by a vector $\boldsymbol{v}_t \in \mathbb{R}^X$, as described below.

Tag Context Similarity. The Tag Context Similarity (TagCont) is computed in the vector space $\mathbb{R}^T$, where, for tag t, the entries of the vector $\boldsymbol{v}_t \in \mathbb{R}^T$ are defined by $v_{tt'} := w(t, t')$ for $t \neq t' \in T$, where w is the co-occurrence weight defined above, and $v_{tt} = 0$. The reason for giving weight zero between a node and itself is that we want two tags to be considered related when they occur in a similar context, and not when they occur together.

Resource Context Similarity. The Resource Context Similarity (ResCont) is computed in the vector space $\mathbb{R}^R$. For a tag t, the vector $\boldsymbol{v}_t \in \mathbb{R}^R$ is constructed by counting how often a tag t is used to annotate a certain resource $r \in R$: $v_{tr} := \mathrm{card}\{u \in U \mid (u, t, r) \in Y\}$.

User Context Similarity. The User Context Similarity (UserCont) is built similarly to ResCont, by swapping the roles of the sets R and U: For a tag t, the vector $\boldsymbol{v}_t \in \mathbb{R}^U$ is defined as $v_{tu} := \mathrm{card}\{r \in R \mid (u, t, r) \in Y\}$.

In all three representations, we measure vector similarity by using the cosine measure, as is customary in Information Retrieval [11]: If two tags t_1 and t_2 are represented by $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^X$, their cosine similarity is defined as: $\mathrm{cossim}(t_1, t_2) := \cos \measuredangle(\boldsymbol{v}_1, \boldsymbol{v}_2) = \frac{\boldsymbol{v}_1 \cdot \boldsymbol{v}_2}{||\boldsymbol{v}_1||_2 \cdot ||\boldsymbol{v}_2||_2}$.

FolkRank. FolkRank employs the principle of the PageRank algorithm [12] for folksonomies [3]: a resource which is tagged with important tags by important users becomes important itself. The same holds, symmetrically, for tags and users. By modifying the weights for a given tag in the random surfer vector, FolkRank can compute a ranked list of relevant tags.

To apply the FolkRank to a logsonomy, we assigned high weights to a specific query term t in the random surfer vector. The final outcome of the FolkRank is then (among others) a ranked list of tags which FolkRank judges as related to t. Refer to [2] for a more detailed description of the experimental procedure and to [3] for a detailed description of the FolkRank algorithm.

Table 2. Examples of most related tags for each of the presented measures.

rank	tag	measure	1	2	3	4	5
5	lyrics	*co-occurrence*	song	love	music	day	songs
		folkrank	song	love	music	myspace	songs
		tag context	titles	listen	lyric	called	theme
		resource context	wanna	lyric	gonna	ya	goodbye
		user context	song	music	songs	school	center
37	news	*co-occurrence*	channel	daily	fox	paper	newport
		folkrank	channel	fox	daily	newspaper	county
		tag context	news.com	newspaper	weather	obituaries	newspapers
		resource context	news.com	arrested	killed	accident	local
		user context	county	center	edging	state	city
399	guitar	*co-occurrence*	tabs	chords	tab	free	bass
		folkrank	tabs	chords	lyrics	tab	music
		tag context	banjo	drum	piano	acoustic	bass
		resource context	tabs	tab	tablature	chords	acoustic
		user context	chords	tabs	tab	guitars	chord
474	gun	*co-occurrence*	smoking	paintball	parts	laws	control
		folkrank	guns	rifle	paintball	parts	sale
		tag context	guns	pistol	rifles	rifle	handgun
		resource context	smoking	pistol	rifle	handgun	guns
		user context	safes	guns	pistol	holsters	pellet
910	brain	*co-occurrence*	tumor	stem	injury	symptoms	tumors
		folkrank	cancer	symptoms	tumor	blood	disease
		tag context	pancreas	intestinal	liver	thyroid	lungs
		resource context	tumor	tumors	syndrome	damage	complications
		user context	stem	feline	tumor	acute	urinary
4764	vest	*co-occurrence*	herb	life	patterns	hd	pattern
		folkrank	herb	motorcycle	vests	patterns	shooting
		tag context	vests	sweaters	jacket	sweater	knit
		resource context	jacket	set	bag	shorts	stainless
		user context	herb	hd	vests	sec	lawsuits

5 Qualitative Insights

A first natural question that arises when trying to compare tag relatedness measures on both logsonomies and folksonomies is to which extent both vocabularies overlap. We found that $4,451$ out $10,000$ tags[2] were present in both datasets (i. e., roughly 44%). Looking up these tags in an English dictionary showed that 92% of them are proper English words; this confirms the intuition that the vocabulary used for tagging and searching is substantially different, but has an overlap of "generic" terms. Figure 1 plots the tag rank for each overlapping tag in the folksonomy (del.icio.us) against its rank in the logsonomy (AOL). Please note that a low tag rank corresponds to a high usage frequency. One can see that high-frequency (i. e., low-rank) tags in the folksonomy tend to be frequently used in a logsonomy as well (roughly the top 1,000 tags); apart from this, there seems to be no special correlation between the usage frequency of tags in both datasets.

Following the methodology of [2], our next step was to compute, for each of the $10,000$ most frequent tags of the AOL log, its most closely related tags using each of the measures described above.

Table 2 provides a few examples of the related tags returned by the measures under study. A first observation is that the cooccurrence relatedness seems to

[2] Please recollect that we use for sake of simplicity the term "tag" to subsume tags in a folksonomy and query words in a logsonomy.

Table 3. Overlap between the 10 most closely related tags.

	co-occurrence	*FolkRank*	*tag context*	*resource context*
user context	*2.28*	*2.16*	0.71	1.11
resource context	1.93	*2.25*	1.5	
tag context	0.88	1.1		
FolkRank	**5.91**			

often "restore" compound expressions like *news channel, guitar tabs, brain tumor.* This can be naturally attributed to the way how the logsonomy was constructed, namely by splitting queries (and consequently also compound expressions) using whitespace as delimiter. We could observe this behaviour also partially in our last study on folksonomy data (see [2]), however to a much lesser extent. Another observation which is identical to the folksonomy data is that cooccurrence and folkrank relatedness seem to often return the same related tags.

The tag context relatedness seems to yield substantially different tags. Our experience from folksonomy data (where this measure discovered preferentially synonym or "sibling" tags) seems to also prove true for logsonomy data: The most related by tag context relatedness is often a synonym (e. g., *gun* – *guns*, *vest* – *vests*), whereas the remaining tags can be regarded as "siblings". For example, for the tag *brain* it gives other organs of the body, whereas for the tag *guitar* it gives other music instruments. When we talk about "siblings" we mean that these tags could be subsumed under a common parent in some suitable concept hierarchy; in this case, e. g., under *organs* and *music instruments*, respectively. In our folksonomy analysis, this effect was even stronger for the resource context relatedness – a finding which does not seem to hold for logsonomy data, based on this first inspection. The resource context relatedness does exhibit some similarity to the tag context relatedness, but gives in general a mixed picture. User context relatedness is even more blurred – the latter observation is again in line with the folksonomy side.

These first observations suggest that despite the reported differences, especially the tag context in a logsonomy seems to hold a similar semantic information to the one we found in folksonomy data.

Our next systematic step is to check whether the most closely related tags are shared across the measures of relatedness. We consider the 10,000 most popular tags in AOL, and for each of them we compute the ten most related tags according to each of the relatedness measures. Table 3 reports the average number of shared tags for the relatedness measure we investigate. The results are again very close to our folksonomy analysis – in general, there is a rather small overlap between the lists of the 10 most related tags (between 0.71 and 2.28) – with an exception of almost six shared tags in average between cooccurrence and FolkRank. This supports our assumption that the computation of FolkRank on a logsonomy also tends to be dominated by the tag-tag cooccurrence network.

To better investigate this point, for each of the 10,000 most frequent tags in the AOL log, we computed the average rank (according to global frequency) of its ten most closely related tags, according to each of the relatedness measures under study. The results are shown in Figure 2, along with the folksonomy results

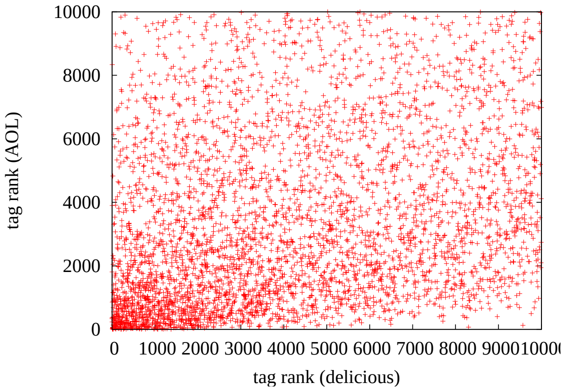
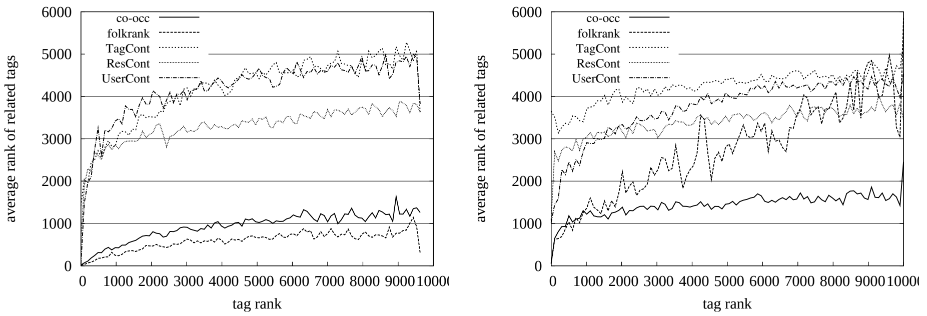


Fig. 1. Correlation between the tag ranks in the AOL logsonomy and the delicious folksonomy.

for comparison. One can see that co-occurrence, resource and user context relatedness show almost identical behaviour compared to folksonomies: Especially the co-occurrence relatedness does have the same strong bias towards high-frequency (i. e., low-rank) tags, independently of the frequency of the original tag. This effect is not so strong for the context measures; however, the distribution for tag context relatedness shows a significant difference: For very popular tags (roughly the top 2000 tags), its most related tags are comparatively rarely used ones (with a tag rank around ∼3500). We hypothesize that this could reflect the topical diversity of both datasets: Because the folksonomy is dominated by technophile topics, its most popular tags are probable to fall into that category. This implies especially that "sibling" tags of a "technical" tag are probably also used frequently. If the topical diversity among the popular tags in the logsonomy is higher, then the sibling tags are more probable to point towards less frequently used tags – which is what we observe here. Another remarkable difference is the behaviour of FolkRank; despite its high overlap with the co-occurrence relatedness reported in Table 3, their profiles differ significantly. The strongly peaked plot of folkrank suggests that there might exist some "outliers" – i. e., very infrequent tags – besides the overlap with the co-occurrence relatedness.

The last question we asked ourselves in this first step is to which extent a given measure returns the same tags when applied to a logsonomy and a folksonomy. To this end we restricted the examination to the overlapping 4451 tags (i. e., for each tag in the overlap, we computed its ten most closely related tags by all measures, whereby the most related tags had to be in the overlap again). Interestingly, we did not find a significant overlap between any two measures (the overlap values ranged from 0.5 to 2.3). For this reason we skip the inclusion of the complete overlap table, as it does not provide additional information. We take this as an indicator that – despite the similarities reported above – the actual semantics contained in folksonomies and logsonomies differ.



(a) folksonomy (b) logsonomy

Fig. 2. Average rank of the related tags as a function of the rank of the original tag.

Table 4. WordNet coverage of logsonomy tags.

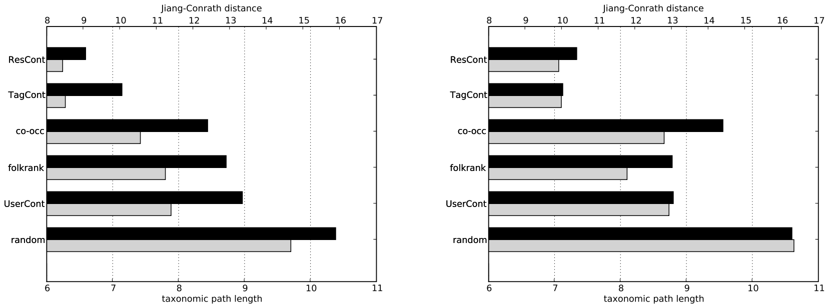
# top-frequency tags	100	500	1,000	5,000	10,000
fraction in WordNet (AOL split query logsonomy)	95 %	96 %	94 %	88 %	81 %
fraction in WordNet (del.icio.us folksonomy)	82 %	80 %	79 %	69 %	61 %

6 Semantic Analysis

Following the qualitative analysis of the previous section, we now go one step further to a more formally grounded characterization of the measures under consideration. In our previous work [2], we introduced the notion of *Semantic Grounding*: The basic idea hereby is to ground the relations between the original and the related tags by looking up the tags in an external structured dictionary of word meanings. Within these structured knowledge representations, there exist often well-defined metrics of semantic similarity; based on these, one can infer which type of *semantic* relation holds between the original and the related tags.

We follow this approach and use WordNet [13], a semantic lexicon of the English Language. The core structure we exploit hereby is its built-in taxonomy of words, grouped into *synsets*, which represent distinct concepts. Each synset consists of one or more words, and is connected via the *is-a* relation to other synsets. The resulting directed acyclic graph connects *hyponyms* (more specific synsets) to *hypernyms* (more general synsets).

Based on this semantic graph structure, several metrics of semantic similarity have been proposed. The most intuitive one is simply counting the number of nodes one has to traverse from one synset to another one. We adopted this *taxonomic shortest-path length* for our experiments. In addition, we use a measure of semantic distance introduced by Jiang and Conrath [5] which combines the taxonomic path length with an information-theoretic similarity measure by Resnik [14]. The choice of this measure was guided by a work of Budanitsky and Hirst [6], who showed by means of a user study that the Jiang-Conrath distance comes most closely to what humans perceive as semantically related. We use the implementation of those measures available in the `WordNet::Similarity` library [15].



(a) folksonomy

(b) logsonomy

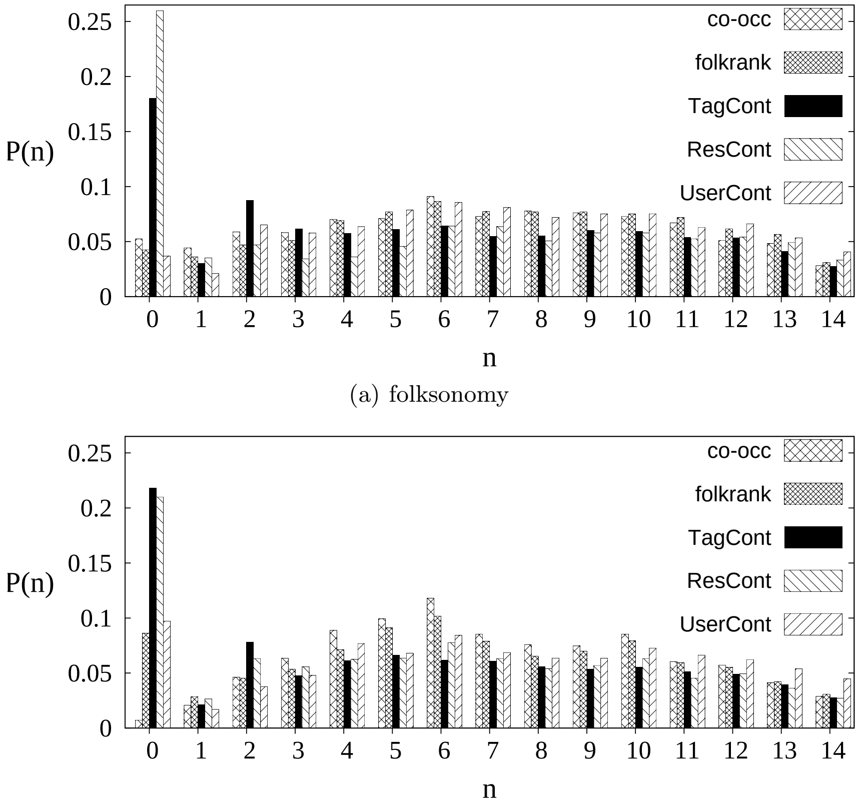
Fig. 3. Average semantic distance, measured in WordNet, from the original tag to the most closely related one. The distance is reported for each of the measures of tag similarity discussed in the main text (labels on the left). Grey bars (bottom) show the taxonomic path length in WordNet. Black bars (top) show the Jiang-Conrath measure of semantic distance.

A natural prerequisite for the semantic grounding described above is that a significant fraction of the most popular tags in the logsonomy (in other words, the most popular search query terms) is present in WordNet. Despite some limiting factors (different languages, misspellings, queries for names of persons or things, . . .), Table 4 shows a relatively high overlap – 81 % of the 10,000 most popular search terms are in fact proper English words. This is significantly more than in our previous work with a snapshot of the del.icio.us folksonomy, which is probably due to the more idiosyncratic nature of folksonomy tags. The higher overlap puts the following grounding process on an even more solid basis.

Following the pattern proposed in [2], we carry out a first assessment of our measures of relatedness by measuring – in WordNet – the average semantic distance between a tag and the corresponding most closely related tag according to each of the relatedness measures under consideration. For each tag of our logsonomy, we find its most closely related tag using one of our measures; if we can map this pair to WordNet (i.e., if both tags are present), we measure the semantic distance between the two synsets containg these two tags. If any of the two tags occurs in more than one synset, we use the pair of synsets which minimizes the path length.

Figure 3 reports the average semantic distance between the original tag and the most related one, computed in WordNet by using both the taxonomic path length and the Jiang-Conrath distance. Overall, the diagrams are quite similar in respect to structure and scale. In both cases, the random relatedness (where we associated a given tag with a randomly chosen one) constitutes the worst case scenario.

Similar to our prior results for folksonomies (i. e., those shown in Figure 3a), for the logsonomy the tag and resource context relatedness measures yield the semantically most closely related tags. However, in the logsonomy case, the context resource relatedness could not repeat the superior performance it showed for the folksonomy. We attribute this to the way how the logsonomy is built: When users



(a) folksonomy

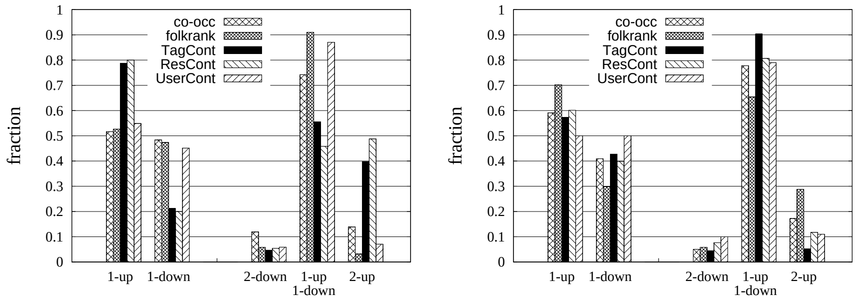
(b) logsonomy

Fig. 4. Probability distribution for the lengths of the shortest path leading from the original tag to the most closely related one. Path lengths are computed using the subsumption hierarchy in WordNet.

tag *implicitly* a certain URL by clicking on it, they are probably not as aware of the actual content of this page as a user who *explicitly* tags this URL in a social bookmarking system.

Another remarkable difference compared to the folksonomy data is that the co-occurrence relatedness yields tags whose meanings are comparatively distant from the one of the original tag. A further examination (see section 5) revealed that co-occurrence often “reconstructs” compound expressions; e. g., the most related tag to *power* according to co-occurrence relatedness is *point*. This is a natural consequence of splitting queries and consequently splitting compound expressions as we did; so our results confirm the intuitive assumption that the semantics of isolated parts of a compound expression usually are semantically complementary.

Figure 3b shows – as in the folksonomy case – that the analysis of the semantic measures for the logsonomy data yields basically the same results with the path length as with the Jiang-Conrath measure. Therefore, we will stick to the simpler-to-understand path length in the sequel.



(a) folksonomy

(b) logsonomy

Fig. 5. Edge composition of the shortest paths of length 1 (left) and 2 (right). An "up" edge leads to a hypernym, while a "down" edge leads to a hyponym.

A more fine-grained analysis of the nature of related tags can be obtained by analyzing the shortest paths that lead from a given tag to its most closely related tag (according to our measures under consideration). Figure 4 displays the normalized distribution $P(n)$ of shortest-path lengths n (number of edges) connecting a tag to its closest related tag in WordNet. The most obvious analogy to [2] is the strong peak of both tag and resource context relatedness at a path length of 0. Paths of length 0 reveal a synonym relation in WordNet, which means that the two query words appear in the same synset in WordNet. Interestingly, the co-occurrence measure shows very low probability of finding synonyms, but is highest when it comes to longer path lengths (for example $n = 6$). This again is in line with our assumption, that the co-occurrence relatedness finds compound expressions which appear to have longer path lengths within WordNet.

For both the logsonomy and the folksonomy and for all measures under study, a path length of 1 occurs very infrequently. This indicates that none of the measures frequently returns direct hypernyms or direct hyponyms. The contrast between high and low probabilities for the path lengths of 0 and 1 can be attributed to the fact, that a path length of 1 leads to either a hypernym or a hyponym in the WordNet hierarchy, never to a sibling. None of the applied measures reveal such a hierarchical relation.

Next, we focus on the shortest path lengths of $n = 1$ and $n = 2$ in the two datasets, e. g., the potential hypernym/hyponym and sibling relations. For $n = 2$ (right-hand side of the subfigures in Figure 5), all measures show – both for folksonomies and logsonomies – a prevalent peak for siblings (1-up/1-down, corresponding to a hypernym edge (up) and a hyponym edge (down)). This observation especially holds for the tag, resource and user context measures. Surprisingly, in the logsonomy, this also holds (though with a lower probability) for the co-occurrence relatedness – in contrast to the folksonomy case. Considering the process of search, some users probably tend to describe their information need with "sibling" query terms like *microwave oven* or *black white*. When interpreting these results, one also has to keep in mind that the absolute number of 1-up-1-down pairs is much larger for tag and resource context relatedness (424 / 367) compared to the other three

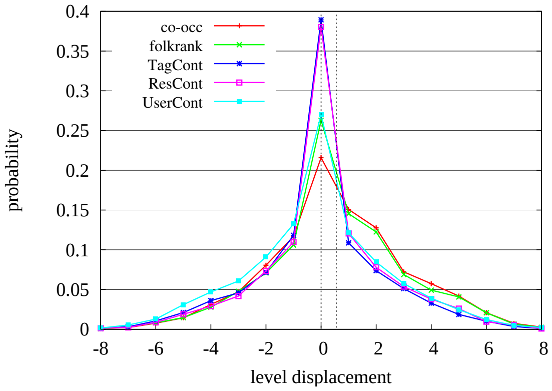


Fig. 6. Probability distribution of the level displacement Δl in the WordNet hierarchy.

measures (279 / 257 / 200 for co-occurrence, FolkRank and user context, respectively). For paths with $n = 1$, we observe – except of the user context relatedness – a slight preference towards hypernym edges (e. g., one level up in the WordNet taxonomy). This finding is the strongest for the FolkRank, but also the other three measures show a slight tendency to reveal hypernyms rather than hyponyms. Especially for the co-occurrence relatedness this behaviour is rather different from our results on folksonomies; again we think that one can see this as another indicator that co-occurrence relatedness restores compound terms in the logsonomy case.

When we generalize the analysis of Figure 5 to paths of arbitrary length, however, the slight tendency towards hypernym edges for the context relatedness measure vanishes. Figure 6 displays the *hierarchical displacement* Δl, i. e., the difference in hierarchical depth between the synset where the path starts and the synset where the path ends. Δl is the difference between the number of edges towards a hypernym (up) and the number of edges towards a hyponym (down). We do not include the folksonomy data for comparison here because it is nearly identical (see [2]). In both cases, we observe a strong peak at $\Delta l = 0$ for all context relatedness measures, which means that the measures do not imply a systematic bias towards more general or more specific terms. The average value of Δl for all the contextual measures is $\overline{\Delta l} \simeq 0$ (dotted line at $\Delta l = 0$). The probability distributions for both co-occurrence and folkrank relatedness are less symmetric and have both an average of $\overline{\Delta l} \simeq 0.55$ (right-hand dotted line). This means that for these measures – as we have already observed – the related tags lie preferentially higher in the WordNet hierarchy.

7 Conclusions and Outlook

In this paper, we investigated emergent semantics in search engine logs by means of term relatedness measures that have been shown to reveal semantic information inherent in the folksonomy graph [2]. We built our analysis on a folksonomy-like representation of the logdata, namely on *logsonomies*, because prior work showed promising structural similarities between log- and folksonomies [1]. This approach allowed us to directly compare log- and folksonomies in respect to the kinds of semantics captured by the different relatedness measures. In order to provide a semantic grounding of the measures under study, we used WordNet and well-established measures of semantic relatedness.

Resuming the research questions stated in the introduction of this paper, we can summarize our contributions as follows:

Emergent semantics in Logsonomies: The presence of inherent semantics in query log data has been reported before. Our contribution is to show that – despite some differences – the formalization of log data into logsonomies retains the semantic information and facilitates the application of established folksonomy analysis techniques to capture the semantics. However, the process of logsonomy construction seems to play an important role: In our case (i. e., when tags are created by splitting the original queries), the co-occurrence relatedness tends to restore compound expressions contained in the original queries.

Comparison of semantics in logsonomies and folksonomies: Our presented approach allows for a direct comparison of the semantics emerging from *explicit* tagging in social bookmarking systems and *implicit* tagging by clicking on search engine results. The results demonstrate that the *type* of inherent semantic information is similar in both cases, but the actual *instances* seem to vary. In other words: Both structures allow mining for synonym and sibling terms, but the *actual* synonyms and siblings retrieved *for a given term* differ.

Characteristics of relatedness measures: Interestingly, applying the resource context relatedness to logsonomies is much less precise in discovering semantically close terms, compared to a folksonomy. We attribute this mainly to incomplete user knowledge about the content of a result page they click on, leading e. g., to "erroneous" clicks. The behaviour of the tag context measure is more similar to the folksonomy case, which recommends it as a candidate for synonym and "sibling" term identification. Additionally, the semantics of the co-occurrence relatedness is strongly influenced by the process of constructing the logsonomy.

In general, we think that our work can help to model the semantic implications of user interactions with search engines. Ultimately, a deeper understanding of this will facilitate the improvement of search engines (e. g., via query expansion) on the one hand, and the harvesting of ontologies for Semantic Web applications on the other hand. We are currently working on voting approaches for combining several measures of relatedness in order to separate even more clearly e. g., synonym and sibling terms. Another promising research direction is to further characterize and understand the structural differences between logsonomies and folksonomies which are responsible for the different behaviour of some relatedness measures.

Acknowledgement. This research was partially funded by the European Commission in the project "Tagora – Emergent Semiotics in Social Online Communities" and by DFG in the project "Info 2.0 – Informationelle Selbstbestimmung im Web 2.0".

References

1. Krause, B., Jäschke, R., Hotho, A., Stumme, G.: Logsonomy - social information retrieval with logdata. In: HT '08: Proceedings of the Nineteenth ACM Conference on Hypertext and Hypermedia, New York, NY, USA, ACM (2008) 157–166
2. Cattuto, C., Benz, D., Hotho, A., Stumme, G.: Semantic grounding of tag relatedness in social bookmarking systems. The Semantic Web - ISWC 2008 (2008) 615–631
3. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Information retrieval in folksonomies: Search and ranking. In Sure, Y., Domingue, J., eds.: The Semantic Web: Research and Applications. Volume 4011 of LNAI., Heidelberg, Springer (2006) 411–426
4. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. In: WWW'98, Brisbane, Australia (1998) 161–172
5. Jiang, J.J., Conrath, D.W.: Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. In: Proceedings of the International Conference on Research in Computational Linguistics (ROCLING), Taiwan (1997)
6. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics **32**(1) (2006) 13–47
7. Baeza-Yates, R., Tiberi, A.: Extracting semantic relations from query logs. In: KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2007) 76–85
8. Zhang, D., Dong, Y.: A novel web usage mining approach for search engines. Computer Networks **39**(3) (june 2002) 303–310
9. Francisco, A.P., Baeza-Yates, R.A., Oliveira, A.L.: Clique analysis of query log graphs. In Amir, A., Turpin, A., Moffat, A., eds.: SPIRE. Volume 5280 of Lecture Notes in Computer Science., Springer (2008) 188–199
10. Pass, G., Chowdhury, A., Torgeson, C.: A picture of search. In: Proc. 1st Intl. Conf. on Scalable Information Systems, ACM Press New York, NY, USA (2006)
11. Salton, G.: Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1989)
12. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems **30**(1-7) (April 1998) 107–117
13. Fellbaum, C., ed.: WordNet: an electronic lexical database. MIT Press (1998)
14. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: Proceedings of the XI International Joint Conferences on Artificial. (1995) 448–453
15. Pedersen, T., Patwardhan, S., Michelizzi, J.: Wordnet::similarity - measuring the relatedness of concepts (2004) http://citeseer.ist.psu.edu/665035.html.