Natural Language Processing

SOCIAL CONTEXT AS MACHINE-PROCESSABLE KNOWLEDGE

Alexander Troussov, John Judge, Mikhail Alexandrov, Eugene Levner

***Abstract**: In this paper, we show how to represent to our formal reasoning and to model social context as knowledge using network models to aggregate heterogeneous information. We show how social context can be efficiently used for well understood tasks in natural language processing (such as context-dependent automated, large scale semantic annotation, term disambiguation, search of similar documents), as well as for novel applications such as social recommender systems which aim to alleviate information overload for social media users by presenting the most attractive and relevant content. We present the algorithms and the architecture of a hybrid recommender system in the activity centric environment Nepomuk-Simple (EU 6th Framework Project NEPOMUK): recommendations are computed on the fly by network flow methods performing in the unified multidimensional network of concepts from the personal information management ontology augmented with concepts extracted from the documents pertaining to the activity in question.*

***Keywords**: multidimensional networks, graph-based methods, network flow methods, data mining, natural language processing, recommender systems.*

***ACM Classification Keywords**: H.3.4 [Information Storage and Retrieval]: Systems and Software – information networks; H.3.5 [Information Storage and Retrieval]: Online Information Services – data sharing.*

Introduction

We live in an increasingly interconnected world of socio-technological systems, in which technological infrastructures composed of many layers are interoperating within a social context that drives their everyday use and development. Nowadays, most digital content is generated within public systems like Facebook, Delicious, Twitter, blogs and wiki systems, and also enterprise environments such as Microsoft SharePoint, and IBM Lotus Connections. These applications have transformed the Web from a mere document collection into a highly interconnected social space where documents are actively exchanged, filtered, organized, discussed and edited collaboratively.

The emergence of the Social Web opens up unforeseen opportunities for observing social behavior by tracing social interaction on the Web. In these socio-technological systems “everything is deeply intertwingled” using the term coined by the pioneer of the information technologies Ted Nelson [Nelson, 1974]: people are connected to other people and to “non-human agents” such as documents, datasets, analytic tools, tags and concepts. These networks become increasingly multidimensional [Contractor, 2008] providing rich context for network mining and understanding the role of particular nodes representing both people and digital content.

In this paper we show how to represent to our formal reasoning and to model social context as knowledge using network models to aggregate heterogeneous information. We show how social context could be efficiently used for well understood tasks of natural language processing (such as context-dependent automated, large scale semantic annotation, term disambiguation, search of similar documents ([Troussov et al., 2009a], [Judge et al., 2008]), as well as for novel applications such as social recommender systems which aim to alleviate information overload for social media users by presenting the most attractive and relevant content.

The log-files of social services and the links between human and non-human agents can be interpreted as triples: who did what on a social service, what node/entity is connected to another, etc. If aggregated, these triples can be treated as a single multidimensional network, which we will refer to as the social context.

The social context can be considered as knowledge in the same way as the semantic networks which are formed from concepts represented in ontologies. From the point of view of the traditional dichotomy between codification and collaboration approaches to knowledge management, the social context could be considered as bottom-up created social knowledge. As knowledge, the social context is a weaker type of knowledge when contrasted with ontologies and taxonomies in that it lacks proper conceptualisation, the links are usually typed and cannot be readily used for inferencing. Correspondingly, the potential of this knowledge can be fully revealed only by robust methods which are tolerant to errors and incompleteness of knowledge which is endemic in any user created, user centric knowledge system. Therefore instead of relying on the traditional logical methods of working with ontological semantic networks, we rely on graph-based methods which can be interpreted as methods of soft clustering and fuzzy inferencing. Graph-based methods provide clear intuition and elegant mathematics to mine networks. The applications described in this paper are based on the use of spreading activation methods [Troussov et al., 2009], which are more generic diffusion-based methods when compared to link analysis in Google's PageRank [Langville and Meyer, 2006], and in the FolkRank algorithms used for tag recommendation in Folksonomies [Jaschke et al., 2007].

As a processable knowledge for understanding the documents embedded into a socio-technological system, this social context has advantages over traditional ontologies. The social context is up-to-date knowledge about a subject area or community of users of a system which changes rapidly to reflect interests and developments in the area. It is populated with nodes representing the current state of the information and how it relates to processed texts and to the realities of a particular socio-technological system (people, projects, social groups).

The current trend in corpus linguistics is for bigger and bigger corpora in order to draw more general analyses. In order to provide the type of text analysis needed to drive the development of the social web, we need to look beyond the corpora and documents themselves and draw upon the individual context within which the documents exist. Instead considering how documents in the system relate to each other and also entities (people, tasks, ideas....) outside the scope of the traditional corpus but which have relevance when it comes to analysing the data in the text itself. In addition to word-level, paragraph-level and corpus-level text processing, text analytics on the socio-technological level yields a wealth of interesting and useful data and will play increasingly important role in future advances in this area.

We outline the use of spreading activation methods to navigate multidimensional social networks and to rank the nodes representing the actors ([Kinsella et al., 2008], [Troussov et al., 2008a], and [Troussov et al., 2008b]).

Finally we describe the algorithms and the architecture of the hybrid recommender system (partially covered in [Troussov et al., 2008b],[Nepomuk project, 2008], and [Troussov et al., 2009b]) in the activity centric environment Nepomuk-Simple (EU 6th Framework Project NEPOMUK): recommendations are computed on the fly by graph-based methods performing in the unified multidimensional network of concepts from the personal information management ontology augmented with concepts extracted from the documents pertaining to the activity in question.

Representing Social Context as a Knowledge and its Modeling by Multidimensional Networks

Nowadays, most digital content is generated within public systems like Facebook, Delicious, Twitter, blogs and wiki systems, and also enterprise environments such as Microsoft SharePoint, and IBM Lotus Connections. These applications have transformed the Web from a mere document collection into a highly interconnected social space where documents are actively exchanged, filtered, organized, discussed and edited collaboratively. To corroborate our interpretation of the social context, as constituted by the realities of socio-technological systems, could be interpreted as knowledge, and to illuminate specific of this knowledge, let us consider a contrived network depicted on the Fig. 1.

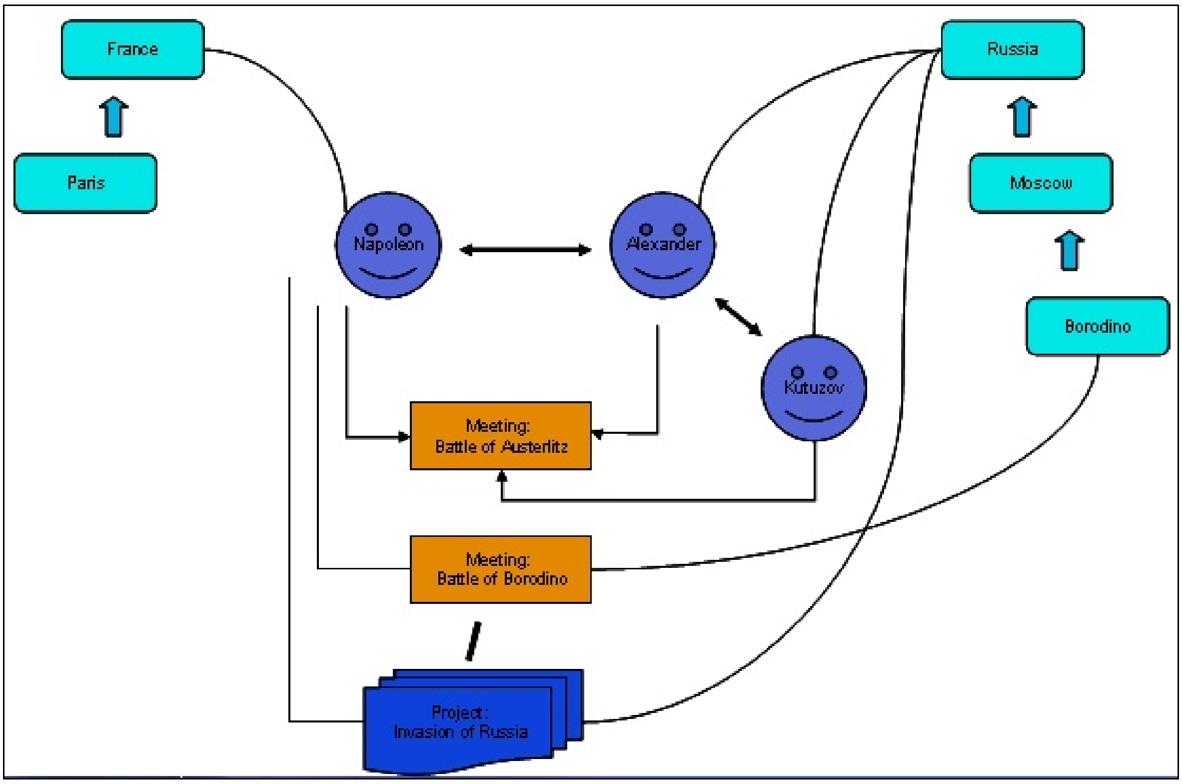

Fig. 1. A contrived example of a network which could be interpreted as a part of an ontology, or as a snapshot from a social site (log-files of socio-technological systems keep track about who did what; triples could be aggregated into a network)

From the point of view of the traditional dichotomy between codification and collaboration approaches to knowledge management, the social context could be considered as bottom-up created social knowledge.

Graphs are one of the traditional means of knowledge representation having advantages of extensibility, and the ease of merging heterogeneous information. One can use the nodes of a graph for facts, concepts, people, organizations, etc and the arcs as binary relationships between them.

Applications of Network Flow based Methods for Mining and Using Network Models of Social Context

In Sections 3-5 we show that social context modeled as a multidimensional network can be used as an efficient machine processable knowledge representation for various tasks. The application of this method includes such traditional areas of knowledge usage as knowledge based text understanding, and the recently emerged area of recommender systems. We analyse several applications and show that computational methods used in these applications are based on the network flow process, “that focuses on the outcomes for nodes in a network where something is flowing from node to node across the edges” ([Borgatti and Everett, 2006]). Following [Troussov et al., 2009] we interpret this “something” as a relevancy measure; for instance, the initial seed input value which shows nodes of interest in the network. Propagating the relevancy measure through outgoing links allows us to compute the relevancy measure for other network nodes and dynamically rank these nodes according to the relevancy measures. The same paradigm could be used to address the centrality measurements in social network analysis. Centralisation of the network can be achieved when we assume that all the nodes are equally important, and iteratively recompute the relevancy measure based on the connections between nodes. In addition to “global” centralisation, “local” centralisation could be performed if the initial seed values represent the nodes of interest.

The applications constituting previous art are monolithic software applications. In this paper we present a novel computational paradigm which breaks these applications into “atomic” components, where the computational methods for propagation are separated as distinct “atomic” network flow engines. This approach provides a unified view of previous applications. From the software engineering perspective the advantages of such an approach includes easy software maintenance, reuse and optimisation of network flow engines, and the guide for new applications.

It appears that the desiderata list for properties of propagation depends on the network properties and the task of propagation itself. Therefore in 3.1 we introduce the formal description of major types of propagation and their use when embedded in larger applications. We also provide a formal description of the “objects” used in these engines – nodes and fuzzy sets of nodes. In further sections we analyse the previous art to explain and justify our list. For each of the engines we indicate the efficient (near-linear with respect to the network size) implementation, however we do not assume that the described implementations are necessarily the best way to perform the task. Finally, we present the generic architecture of a software system which utilises the social context.

3.1. Network Flow Operations over Network Objects

A formal description of the network flow methods applications requires the introduction of some notation. We assume that the social context is represented by a multidimensional network modeled as a directed graph, which is a pair $G = (V,E)$ where

V – is the set of vertices v_i

E – is the set of edges e_j (although in oriented graphs edges are referred to as arcs).

We also introduce the following terms and notations describing the set of “atomic” operations from formal point of view (operands, etc) and the purpose.

Object (Network object) – is a node or a (fuzzy) set of nodes on the network (see [Chen, 1996]). Fuzzy sets are characterized by a membership function M which shows the degree of belonging of an element to the set.

M – the membership function for fuzzy sets which is a non-negative real-valued function.

Activation – the membership function when it is not interpreted in the fuzzy sets paradigm. We use the activation (the activation of nodes, or objects) as an abstract relevancy measure.

Cloud (cloud object) – is formally the same as the object, however we use the term cloud where we want to emphasize the fact that the membership function is non-negative real-valued function, not Boolean valued. As usual, we assume that a node *e* belongs to the fuzzy set *C*, or in mathematical notations $e \in C$ - if M(*e*)>0.

|…| - cardinality of sets. In case of clouds we define $|C| = \Sigma$ M(*e*) for all nodes *e* such that $e \in C$.

Query – an object used as a seed for local ranking (defined below)

Expansion – is a unary operation which transform a cloud into another cloud: Expansion: $C_1 \rightarrow C_2$. If C_1 and C_2 are crisp sets, we assume that C_1 is a proper subset of C_2: $C_1 \subset C_2$. If general, we assume that this operation does not change significantly the values of the membership functions on the nodes in C_1, and that $|C_1| \leq |C_2|$.

Smoothing – is formally the same as expansion, however the interpretation of this operation can not be done in the framework of fuzzy sets, instead, it roots in the operations with functions in calculus. We assume that smoothing makes the difference between the values of the function M() on neighbour nodes smaller.

Local ranking - is formally the same as expansion. The purpose of this operation is to get the value of the activation which shows the proximity, or relevance, of objects to a query.

Shrinking – is a unary operation which transforms a cloud into another cloud: *Shrinking*: $C_2 \rightarrow C_1$. If C_1 and C_2 are crisp sets, we assume that C_1 is a proper subset of C_2, i.e. $C_1 \subset C_2$. If general, we assume that this operation does not change significantly the values of the membership functions on the nodes in C_2, and that $| C_1| \leq | C_2|$. Shrinking is a kind of inverse operation to expansion, although we do not necessarily assume that for any pair of such operations $C_1 \equiv$ *Shrinking*(*Expansion*(C_1) for each object C_1.

3.2. Network Flow Operations over Network Objects

Spreading activation (SA) (see [Troussov et al., 2009]) is one of the network flow methods to implement the operations described above. This is a wide class of algorithms which iteratively propagate the activation (relevancy measure) from the initial seed to other nodes.

3.3. Network Flow Methods for Natural Text Processing

Spreading activation algorithms were used for knowledge based natural language (text) processing ([Judge et al., 2008], [Troussov et al., 2009], [Troussov et al., 2008a], [Troussov et al., 2009b], and [Judge et al., 2007]). In this approach the text is modeled as a cloud of concepts (in a formal definition introduced in Section 3.1) in a semantic network (such as network of concepts from ontologies) and graph-based operations were used for mining of text models. The rationale and intended goals of graph-based methods described in these papers could be recounted as follows.

We assume that the source text is coherent and cohesive as opposed to random list of words. Therefore if some concept are relevant to the text, as indicated by the big value of M, the "neighbour" concepts are also somehow relevant to the text, since the neighbourhood of nodes is defined by links which represent semantic relations between concepts such as synonymy, "is-a", "part-of" etc. We also assume that the keywords (subject terms, subject headings, descriptors), defined in information retrieval as terms that capture the essence of the topic of a document, should have a special position within the clustering structure of the text models (for instance, they hardly exist outside of strong clusters induced by the terms mentioned in the document). Term disambiguation, and other tasks of natural language understanding, are usually perceived as inference: "mentioning of *car* in a sentence increases our awareness that the term *Jaguar* mentioned in the same text refers to a *car*" [Troussov et al., 2009]. However, inferencing from one term is not quite reliable, while inferencing based on mentions of various terms is more reliable from a probabilistic point of view, which was confirmed by our numerical simulation showing sharp phase transition from uncertainty to certainty with the increase of number of the nodes in the initial seed.

Finding the key terms is done by spreading the activation from concepts mentioned in the text to other concepts in a semantic network. From this point of view the purpose of these operations could be classified as local ranking. The proper ranking should be achieved by *Smoothing* to account for inferencing. The key concepts of a text are not necessarily mentioned in the text, so the operation is one of *Expansion*. So as an end to end solution the knowledge based semantic processing of a text is transforming the seed (concepts mentioned in a text) to a larger set of concepts and providing *Local ranking*.

3.4. Navigating Networks of People and Associated Objects

Social spaces such as blogs, wikis and online social networking sites are enabling the formation of online communities where people are linked to each other through direct profile connections and also through the content items that they are creating, sharing, tagging, etc. The Semantic Web provides a platform for gathering diverse information from heterogeneous sources and aggregating such linked data into multidimensional network of nodes representing people, organisations, projects etc. Spreading activation methods were used in [Kinsella et al., 2008] to augment objects from social spaces, by highlighting related objects, recommending tags, and suggesting relevant sources of knowledge.

Recommendations are performed using a network flow engine which, in light of the current paper, provides the operation of *Local ranking* defined in Section 3.1 in the ego-centric network defined by the *Query*.

3.5. Hybrid Recommender System in the Activity Centric Environment Nepomuk-Simple

The concept of navigation in the ego-centric networks [Kinsella et al., 2008] by queries which are single nodes of the network, was extended to navigation by queries which are *Clouds* in [Troussov et al., 2008b].

This paper presents the architecture of the hybrid recommender system in the activity centric environment Nepomuk-Simple (EU 6th Framework Project NEPOMUK).

“Real” desktops usually have piles of things on them where the users (consciously or unconsciously) group together items which are related to each other or to a task. The so called Pile UI, used in the Nepomuk-Simple imitates this type of data and metadata organisation which helps to avoid premature categorisation and reduces the retention of useless documents.

Metadata describing user data is stored in the Nepomuk Personal Information Management Ontology (PIMO). Proper recommendations, such as recommendations for additional items to add to the pile, apparently should be based on the textual content of the items in the pile. Although methods of natural language processing for information retrieval could be useful, the most important type of textual processing are those which allow us to relate concepts in PIMO to the processed texts. Since any given PIMO will change over time, this type of natural language processing cannot be performed as pre-processing of all textual context related to the user. Hybrid recommendation needs on-the-fly textual processing with the ability to aggregate the current instantiation of PIMO with the results of textual processing.

Modeling this ontology as a multidimensional network allows the augmentation of the ontology with new information, such as the “semantic” content of the textual information in user documents. Recommendations in Nepomuk-Simple are computed on the fly by graph-based methods performing in the unified multidimensional network of concepts from PIMO augmented with concepts extracted from the documents pertaining to the activity in question. In this paper, we classify Nepomuk-Simple recommendations into two major types. The first type of recommendations is the recommendation of additional items to the pile, when the user is working on an activity. The second type of recommendation arises, for instance, when the user is browsing the Web; Nepomuk-Simple can recommend that the current resource might be relevant to one or more activities performed by the user. In both cases there is a need to operate with *Clouds* (fuzzy sets of PIMO nodes): *Clouds* describe topicality of documents in terms of PIMO as we described in Section 3.3, the pile itself is a *Cloud*.

Similarity of Sets of Nodes and Search for Similar Sets

Discussion on the distinction between similarity and proximity of any given network nodes is outside of the scope of this paper. In this Section we present an empirical approach to the computation of similarity based on a network flow process. The similarity of network nodes, or more generally the similarity of two network objects (like clouds which are fuzzy sets of network nodes) could be described in terms of their ability to affect various parts of the network (like in viral marketing applications [Kempe et al., 2003], [Kempe et al., 2005]). In other words, the similarity of two sets A_0 and B_0 should be defined as the similarity of two fuzzy sets A=*Expanding*(A_0) and B=*Expanding*(B_0), where the operation *Expanding* is done by network flow methods compatible with the targeted applications. For instance, if the target application is in the area of "viral marketing", than we expect that the *Expanding* is done by network flow methods which model "viral marketing".

In section 4.1 we provide additional arguments to justify our approach to similarity and introduce the similarity of two fuzzy sets on a network. In Section 4.2 we describe efficient and scalable implementation of search for similar sets.

Operations with network objects introduced in Section 3.1 could be classified in terms of number of arguments or operands that the operation takes. In logic, mathematics, and computer science, this number is called arity, in linguistics arity is sometimes called valency. All operations introduced in Section 3.1 are unary. Computation of similarity of two nodes or two objects defined in Section 4.1. is a binary operation, search for objects similar to object of interest in a collection of *n* objects is (*n*+1)-ary operation.

4.1 Similarity of Two Sets of Nodes

Traditional measures for set similarity (such as Jaccard similarity coefficient) describe how much in common two crisp sets have in terms of the "exact" match. The similarity value is a number in the range 0 to 1, 0 – no common elements, 1 – the sets are equal:

$$\text{Similarity (Set1, Set2)} = \frac{\text{The number of elements in the common}}{\text{The number of elements in the union of two sets}}$$

or, using mathematical notations:

$$S(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

The links between nodes must be taken into account when comparing the sets of nodes. Instead of the degree of exact match, we need to use a "fuzzy" matching technique. To illustrate this "fuzzy" matching, let us consider a geometrical example of four sets of nodes on a two dimensional grid shown on the Fig. 3 with shapes shown on the Fig 2.

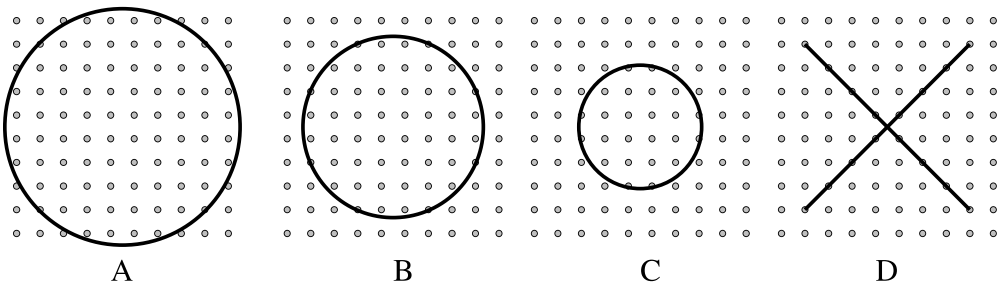

Fig. 2. Four sets of nodes on two dimensional grid

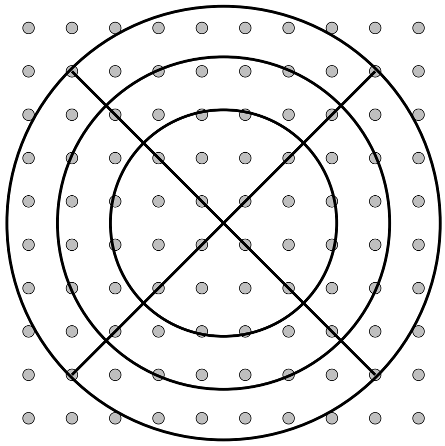

Fig. 3. Four sets of nodes on two dimensional grid with centres of symmetry placed at the origin of the grid

Which two sets of nodes are most similar? If our matching strategy is to look for an exact match, then the pair A and D would be most similar because they have the most nodes in common. However, intuitively, A and B are closest. How do we make a computation based on this intuition which will show us that A and B are very similar? Our approach for "fuzzy" matching is to expand all the sets by making their boundaries less well defined and more "fluffy" and as the measure of similarity between original pair we choose the exact match (i.e. overlap) of their expanded variants.

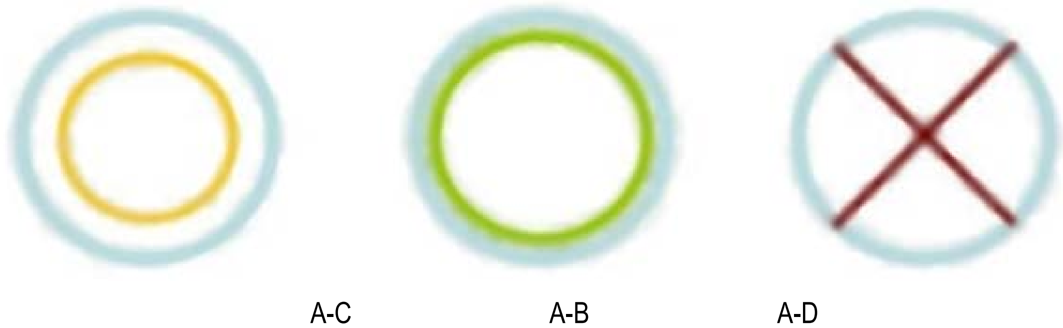

Fig. 4. To provide "fuzzy" comparison of original sets, we perform their "fuzzyfication" first by the operation defined as Expanding in Section 3.1

From the Fig. 4 one can see that expanded sets A and C still do not have common area, and hence the measure of their similarity is zero. Expanded sets A and B became practically indistinguishable, while sets A and D have common areas only in the four corners.

4.2 Retrieval of Similar Sets in a Collection

Operations introduced in Section 3.1 are unary. Computing the similarity of two sets of nodes defined in Section 4.1. is a binary operation. Search for sets similar to the set of interest in a collection of *n* objects is (*n*+1)-ary operation. In the Nepomuk-Simple environment such an operation is used to find activities (piles) similar to user's current activity, or to provide the recommendation that the currently viewed web resource could be useful for particular activities of the user.

A scalable and linear wrt to *n* implementation of this operation could be based on the algorithm suggested in [Troussov et al., 2009b] and the processing scheme on the Fig. 5.

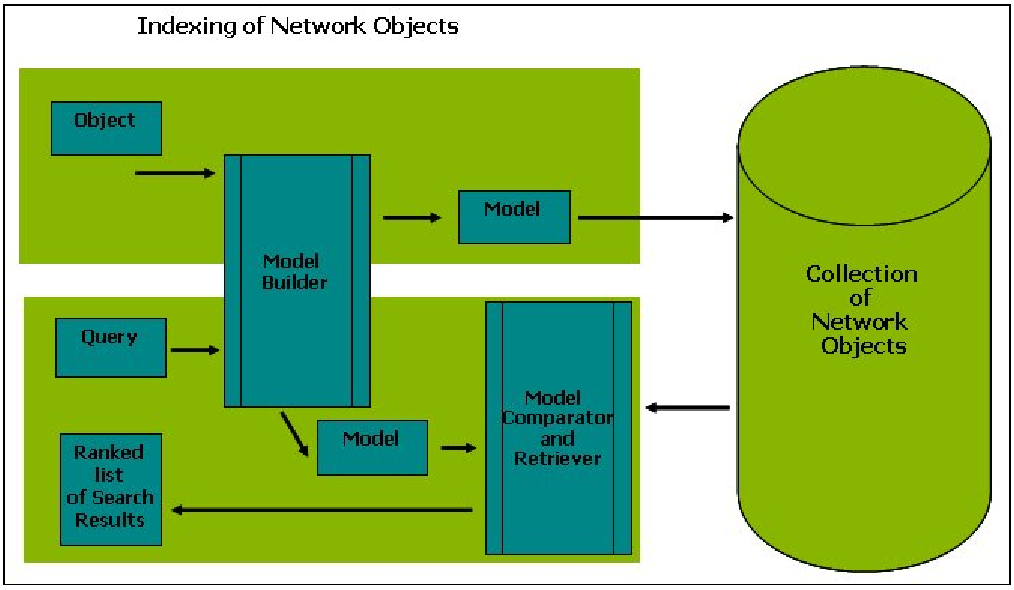

Fig. 5. Indexing and retrieval process to find similar fuzzy sets of nodes
(or network objects defined in Section 3.1)

Network-flow based Computational Systems for Mining and Use of the Social Context

In this Section we present a novel software framework for mining and use of network models of social context based on a set of atomic software engines implementing network flow operations described in Section 3.1. Arguments (operands) of these operations are network objects (fuzzy sets of network nodes).

This framework generalises the design of systems constituting the previous art without introducing new components which could potentially hamper performance and scalability. We show that efficient and scalable implementations for each of the atomic software engines already exist (although as part of monolithic software applications). For instance, the paper [Judge et al., 2007] described the system which performs atomic operations on networks with several hundred thousand nodes in 200msc on an ordinary PC. The paper [Troussov et al., 2008b] describes the large scale multifunctional application where various recommendations are done using hybrid methods including natural text processing. Therefore we conclude that the framework described in this Section could be successfully used to mine and exploit the social context.

Major steps involved in building the software application based on principles described in this paper are:

1. Modeling the social context (such as instantiations of socio-technological systems) using multidimensional networks.

2. Tasks are modeled as a *Cloud* - fuzzy set of nodes which perform the role of *Query*

3. Task dependent models of social context enhancement (the network is enhanced on the fly by new objects and new links between nodes, and augmented by new task dependent objects)

4. Local ranking using *Query* as the initial seed provides the ranked list of network objects relevant to the *Query*

The previous sections provide examples of these steps. For instance, in Nepomuk-Simple the underlying network is enhanced on the fly by concepts extracted from the textual content of pile items.

Conclusions and Future Work

We have revised the previous art in use of network models of weak knowledge, and we described the algorithms and the architecture of the hybrid recommender system in the activity centric environment Nepomuk-Simple (EU 6th Framework Project NEPOMUK):

The applications constituting previous art were monolithic software applications. In this paper we present a novel computational paradigm which breaks these applications into "atomic" components, where the computational methods for propagation are separated as distinct "atomic" network flow engines and described efficient scalable implementations of such operations with the performance on the subsecond level for networks with several hundred thousand nodes. This approach provides a unified view of previous applications. From the software engineering perspective the advantages of such an approach include easy software maintenance, reuse and optimization of network flow engines, and the guide for new applications.

Future work requires refining the set of atomic operations and selection of network flow methods for each of such operations. Evaluation of the results for each operation as well of the applications build from these operations is the next stage.

Bibliography

[Nelson, 1974] Ted Nelson (1974): Computer Lib/Dream Machines. Self-published, 1974, sixth printing (May 1978), ISBN 0-89347-002-3, page DM45

[Contractor, 2008] Contractor, N. (2008). "The Emergence of Multidimensional Networks." Retrieved February 13, 2010, from http://www.hctd.net/newsletters/fall2007/Noshir Contractor.pdf

[Troussov et al., 2009a] Troussov, A, Judge, J., Sogrin, M., Akrout, A., Davis, B., Handschuh, S. "A Linguistic Light Approach to Multilingualism in Lexical Layers for Ontologies", SLT, vol 12, Polish Phonetics Association, ed. G. Demanko, K. Jassem, S. Szpakowicz

[Judge et al., 2008] Judge, J., Nakayama, A. , Sogrin, M., and Troussov, A. "Method and System for Finding a Focus of a Document". Patent Application US 20080/263038 Kind Code: A1. Filing Date: 02/26/2008

[Troussov et al., 2009] Troussov, A., Levner. E., Bogdan, C., Judge, J., and Botvich, D. "Spreading Activation Methods." In Shawkat A., Xiang, Y. (eds). Dynamic and Advanced Data Mining for Progressing Technological Development, IGI Global, USA, 2009.

[Langville and Meyer, 2006] Langville, A.N. and Meyer, C. "Google's PageRank and Beyond: The Science of Search Engine Rankings." Princeton University Press, 2006

[Jaschke et al., 2007] Jaschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., and Stumme, G. (2007) "Tag Recommendations in Folksonomies." Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases PKDD, Warsaw, Poland.

[Kinsella et al., 2008] Kinsella, S., Harth, A., Troussov, A., Sogrin, M., Judge, J., Hayes, C., & Breslin, J.G. (2008). "Navigating and Annotating Semantically-Enabled Networks of People and Associated Objects." Why Context Matters: Applications of Social Network Analysis (T. Friemel, ed.), VS Verlag, ISBN 3531163280, 79-96

[Troussov et al., 2008a] Troussov, A., Sogrin, A., Judge, J., Botvich, D. "Mining Socio-Semantic Networks Using Spreading Activation Technique". Proceedings of I-KNOW '08 and I-MEDIA '08, Graz, Austria, September 3-5, 2008

[Troussov et al., 2008b] Troussov, A., Judge, J., Sogrin, M., Bogdan, C., Lannero, P., Edlund, H., Sundblad, Y. "Navigation Networked Data using Polycentric Fuzzy Queries and the Pile UI Metaphor". Proceedings of the International SoNet Workshop), pp. 5-12, 2008

[Nepomuk project, 2008] Nepomuk PSEW Recommendation: Using the Recommendations View in PSEW. Retrieved June 30, 2011, from http://dev.nepomuk.semanticdesktop.org/wiki/UsingPsewRecommendations

[Troussov et al., 2009b] Troussov, A., Sogrin, Judge, J., Botvich, D. "System and Method for Ontology-based Personalized Semantic Search", Disclosed by IBM. Loaded into the IP.com Prior Art Database on 2009-04-21 UTC, IPCOM000181971D

[Borgatti and Everett, 2006] Borgatti, S. and Everett, M. (2006). A graph-theoretic perspective on centrality. Social Networks, 28(4):466–484.

[Chen, 1996] Chen, C.H., Ed. 1996 Fuzzy Logic and Neural Network Handbook. McGraw-Hill, Inc.

[Kempe et al., 2003] Kempe, D., Kleinberg, J., and Tardos, E. Maximizing the Spread of Influence through a Social Network. Proc. 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 2003.

[Kempe et al., 2005] Kempe, D., Kleinberg, J., and Tardos, E. Influential nodes in a diffusion model for social networks. In Proc. 32nd Intl. Colloq. on Automata, Languages and Programming, pages 1127–1138, 2005

[Judge et al., 2007]Judge, J., Sogrin, M., and Troussov, A. "Galaxy: IBM Ontological Network Miner". Proceedings of the 1st Conference on Social Semantic Web (CSSW), September 26-28, 2007, Leipzig, Germany.

Authors' Information

Alexander Troussov *–Ph.D., IBM Dublin Center for Advanced Studies Chief Scientist. Dublin Software Lab, Building 6, IBM Technology Campus, Damastown Ind. Est., Mulhuddart, Dublin 15, Ireland; e-mail: troussov@gmail.com*

Major Fields of Scientific Research: natural language processing, software technologies, network analysis

John Judge *– Ph.D., Centre for Next Generation Localisation Dublin City University Dublin 9, Ireland; e-mail: jjudge@computing.dcu.ie*

Major Fields of Scientific Research: computational linguistics, natural language processing, social network analysis, semantic web applications

Mikhail Alexandrov *– Professor, Academy of national economy and civil service under the President of Russia; Prosp. Vernadskogo 82, bld. 1, Moscow, 119571, Russia; fLexSem Research Group, Autonomous University of Barcelona, Bellaterra (Barcelona), 08193, Spain; e-mail: MAlexandrov@mail.ru*

Major Fields of Scientific Research: data mining, text mining, mathematical modeling

Eugene Levner *– Professor of Computer Science at Holon Institute of Technology and Bar-Ilan University, Israel. He is the full member of the International Academy of Information Sciences, a member of editorial boards of four international journals; 52, Golomb St, Holon 68102 Israel; e-mail: levner@Hit.ac.il*

Major Fields of Scientific Research: combinatorial optimization, operations research, design and analysis of computer algorithms, algorithm complexity and computability, scheduling theory, grid optimization, network analysis, and risk analysis