Computing and Informatics, Vol. 33, 2014, 877–906

DISCOVERING RELATIONS BY ENTITY SEARCH IN LIGHTWEIGHT SEMANTIC TEXT GRAPHS

Michal Laclavík, Štefan Dlugolinský, Marek Ciglan

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: {michal.laclavik, stefan.dlugolinsky, marek.ciglan}@savba.sk

Abstract. Entity search is becoming a popular alternative for full text search. Recently Google released its entity search based on confirmed, human-generated data such as Wikipedia. In spite of these developments, the task of entity discovery, search, or relation search in unstructured text remains a major challenge in the fields of information retrieval and information extraction. This paper tries to address that challenge, focusing specifically on entity relation discovery. This is achieved by processing unstructured text using simple information extraction methods, building lightweight semantic graphs and reusing them for entity relation discovery by applying algorithms from graph theory. An important part is also user interaction with semantic graphs, which can significantly improve information extraction results and entity relation search. Entity relations can be discovered by various text mining methods, but the advantage of the presented method lies in the similarity between the lightweight semantics extracted from a text and the information networks available as structured data. Both graph structures have similar properties and similar relation discovery algorithms can be applied. In addition, we can benefit from the integration of such graph data. We provide both a relevance and performance evaluations of the approach and showcase it in several use case applications.

Keywords: Text graphs, information networks, entity search, semantic search, entity relation discovery

Mathematics Subject Classification 2010: 05C85, 05C82, 90B40

1 INTRODUCTION

Entity search is becoming a popular alternative for full text search. Google released its entity search [43] based on confirmed, human-generated data such as Wikipedia and Freebase[1], and Facebook is experimenting with graph search over its user generated context. New types of question answering systems such as IBM Watson, based on structured and unstructured data [14], are being developed, but the task of entity discovery, search, or relation search in unstructured text still remains a major challenge in information retrieval and information extraction fields. This paper attempts to address this challenge, focusing specifically on entity relation discovery.

Entity relation discovery is a topic well covered in literature dealing with structured data where the entities under investigation are already identified. Regarding the data in the form of graphs or networks (semantic or information networks), there are approaches for relation discovery based on the network theory and graph algorithms [7]. This includes methods for ranking relationships in ontologies [1]. When the data is unstructured, text mining methods [3, 45, 12, 36] such as pattern detection or clustering are usually used for relation discovery [44]. Well known Snowball method [3] can identify relation based on example patterns in unstructured texts, but can not discover "undefined" relations, which are resolved in StatSnowball [45].

To the best of our knowledge, none of the approaches is suitable for both structured and unstructured data sources. If we want to extract relations from text, databases or ontologies, we have to apply different relation extraction methods to discover relations among entities. The main contribution of this work is to make a step forward to apply the same principle to search and discover entity relations on both structured and unstructured data.

1.1 Motivation and Background

Graphs or networks often appear as a natural form of data representation in many applications such as social networks, call networks, World Wide Web, Wikipedia, LinkedData[2], or emails.

The analysis of email communication allows the extraction of social networks with links to people, organizations, locations, topics or time information. Social networks included in email archives are becoming increasingly valuable assets in organizations, enterprises, and communities, though to date they have been little explored. Unstructured text is still the most common medium for information sharing and communication. While it is available on the web, in emails, or within new social media like Facebook, Twitter or LinkedIn, it is also present in enterprises analytical data like document repositories or even database text fields. All of these

[1] Graph database with 46 million topics, `http://www.freebase.com/`

[2] `http://linkeddata.org/`

web, media communication, and organizational resources preserve a large part of their knowledge in unstructured textual form. In addition, such data is connected with graph/network data through web links, communication links, transactions, or social links and tags (lightweight semantics) in social media and is shared among many users and resources. It has been proved on Web 2.0 (or social web) that the lightweight semantics (tags) and social networks (graph data) give additional value to knowledge sharing, reuse, recommendation and analytics. The text can be transformed to trees or graph/network structures [22] which have a similar property as the information networks discussed in the paper. This paper will further examine those properties. Searching, analyzing, accessing, and visualizing information and knowledge hidden in such network structures are becoming increasingly important tasks in the area of data analytics [2], but different algorithms must be used for unstructured data processing such as text. In this work we try to create network structures from unstructured text data similar to those of structured data. Email communication is unique in this respect because it connects social networks (communication) with information networks, which can be extracted from text. We believe that email communication and its links to other organizational as well as public resources (e.g. LinkedData) can be a valuable source of information and knowledge for knowledge management, business intelligence, better enterprise, and personal email search. The future of email [10] is in interconnecting email with other resources, services (like social networks or collaboration tools), or data and entities which are present in email. This was also the main motivation and drive for our work, but we have discovered that the approach could be applied to any unstructured text data, and not only to email communication. As the size of the real graph data grows, there must be an adequate development in the field of graph data management. Fast graph traversing is the most important feature when querying large graphs. The challenge is to make the graph querying scalable, since graph traversing has to deal with random access pattern to the nodes [29]. Thus, we take performance seriously and evaluate the performance of our approach on large graph networks.

1.2 Text Graphs

The idea of building or extracting graphs from text is not new and it is used to accomplish many tasks in Natural Language Processing (NLP) [30] related to tasks in syntax like Part of Speech (POS) tagging, semantics like word sense disambiguation or applications like topic identification, summarization or machine translation. These topics were also the focus of a series of TextGraphs workshops[3]. In [15], text graphs are used to create signed social network from text discussions. In [32], graph walk algorithms are used to discover related words. An example is discovering synonyms based on the construction of sentence words graphs [31], where text graphs and random walk algorithms are also used to achieve named entity extraction, mes-

[3] `http://www.textgraphs.org/`

sage foldering or person name disambiguation. Text graphs are also introduced as a way to enhance concept maps [35]. In our work we try to use text graphs for entity relation discovery.

1.3 Overview of the Contribution

The main contribution of this paper is in searching and discovering entity relations in unstructured text using lightweight semantic graph data structures extracted from the text. The extracted graph structures have similar properties to other information or social networks, which opens up the possibility for integration of structured and unstructured data. Another advantage compared to other relation discovery approaches comes from user interaction. When the users search for relations, they can interact with the underlying graph data by deleting or merging entities and thus immediately improve search results, specifically discovered relations. We evaluate relation discovery and showcase possible improvements coming from user interaction. We also evaluate the approach by providing several use cases in which the approach seems to be relevant.

2 EXTRACTING LIGHTWEIGHT SEMANTIC TEXT GRAPH

In this section we discuss information extraction techniques focusing on Named Entity Recognition, explain how entities are recognized, and also explain how semantic trees and graphs are constructed. We discuss how information extraction can be improved by user interaction and evaluate our approach. Additionally, we also examine and discuss network properties of the extracted semantic networks.

2.1 Rule-Based Named Entity Recognition

Information Extraction (IE) techniques [12] focus on several information extraction tasks, where Named Entity Recognition is the prominent IE task. In [22], we described in detail the state-of-the-art in information extraction and advantages of pattern-based information extraction. We will not address it in this paper. We assume that information extraction techniques are in place and provide us with useful Named Entity (NE) recognition. The output is based on key-value pairs representing NEs. The work presented in this paper helps in relation discovery among entities and thus it solves mainly the IE task of relation extraction. Since we do not distinguish between NE or NE properties and all entities are treated as key-value pairs, the results of relation discovery are always related entities to one or multiple entities. This means that discovered related entities can show relations, entity aliases, and entity properties. For the Information Extraction we use Ontea [20] IE techniques [22], but any other IE tool that provides key-value pairs with position in text can be used. Ontea is based on regular expressions and gazetteers. Applied patterns and gazetteers extract key-value pairs (key: object type; value: object

value represented by the string) from a text as seen on the left side of Figure 1. If there is textual data present in binary form (e.g. PDF attachment) it is, if possible, converted to text before the information extraction process. Ontea is able to detect document segments such as message replies inside emails. The extracted key-value pairs are then used to build the tree [22] (left bottom side of Figure 1) and the network of entities [22] as a graph structure (right site of Figure 1). The Ontea IE tool is able to connect other extraction/annotation tools like GATE[4] [13], Stanford CoreNLP[5], or WM Wikifier[6]. In most experiments presented in this paper we have used pure gazetteers and regular expressions, along with some extra rules to support better user interaction. In the example below (Figure 1), the WM Wikifier was used and has annotated front desk text as Receptionist[7], detected Executive Director[8] and also recognized text north tower as List of tenants in One World Trade Center[9].

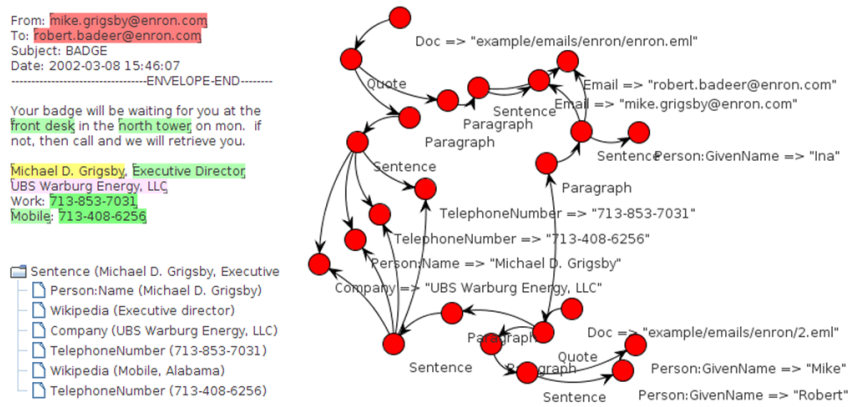


Figure 1. User interface of the IE tool Ontea [22], with highlighted extracted objects (top left) and tree structure (bottom left), which is used to build social network graphs (right)

2.2 Lightweight Semantic Trees and Graphs

We can see a graph built from two emails in Figure 1 on the right side. Such a graph can be built from any text collection in which the document is represented by a document node, its paragraph and sentences nodes, and the documents are

[4] http://gate.ac.uk/
[5] http://nlp.stanford.edu/software/corenlp.shtml#About
[6] http://www.nzdl.org/wikification/
[7] http://en.wikipedia.org/wiki/Receptionist
[8] http://en.wikipedia.org/wiki/Executive_Director
[9] http://en.wikipedia.org/wiki/List_of_tenants_in_One_World_Trade_Center

interconnected by the nodes representing entities which are present in multiple documents. Note the two telephone numbers, company name, and person name nodes connected to two different sentence nodes in Figure 1. Even though these entities have been found in both emails, they are presented only once in the graph. In this respect, they are unique. For both nodes and edges, we know also the numeric value of node or edge occurrence in the collection. This can be used as edge or node weight. So far we have not used it in the relation discovery algorithm, but we plan to do so in the future. On the right side of Figure 1 you can see the directed graph as it was extracted from text (emails) based on the tree structure, which can be seen on bottom left of Figure 1. However, when applying graph algorithms (e.g. spreading activation) for relation discovery, we converted the graph to the undirected form. In some cases we had to set some edge types as directed; this was true for LinkedIn and Event graph use cases described in Section 4.2. Thus we defined the graph as directed, but in most cases we considered that if there is an edge in one direction then the edge in the other direction also exists.

The extracted information can be encoded in different types of mathematical graphs, e.g. hypergraphs and related databases [16], or in labelled graphs such as RDF[10], but since we just identifying co-occurrence and Named Entities, we decide to create simple associativity graphs as described below.

Our Semantic Text Graph[11] can be defined as follows: $G = (V, E)$, where V is a set of vertices and E is a set of edges.

$$V = \{v_i = (\text{key}_i, \text{value}_i) \mid 1 \leq i \leq n\}$$

Each vertex v is composed of a key, value pair, where *key* represents the entity type and value is the text string representing entity (in most cases value is a string extracted from a text document).

$$\forall v_i, \forall v_j; v_i \in V; v_j \in V; \text{key}_i = \text{key}_j \wedge \text{value}_i = \text{value}_j \Rightarrow v_i = v_j$$

A unique node is represented by a unique key, value pair. If the same key, value pair is detected multiple times in a document or in multiple documents, they are merged into a single node and connected by edges to the documents, sentences, or paragraphs where they were discovered.

$$e_{ij} \in E;\ e_{ij} = \{(v_i, v_j); v_i \in V; v_j \in V\}$$

$$e_{ij} \in E \Rightarrow e_{ji} \in E$$

A generated graph contains edges, which connect vertices/entities with other vertices representing sentences, paragraphs, or documents where these entities were

[10] Resource Description Framework, http://www.w3.org/RDF/

[11] We call our graphs "Semantic Graphs" or "Lightweight Semantic Graphs" because they have two important semantic properties: Named Entities extracted from text with defined types and unlabelled relations with other entities/concepts.

discovered. Edges are type-less with no defined properties, although in the future we would like to use edge weights and edge labels (types of relations), or an edge timestamp for better relation discovery. In our current algorithms the edges are used only to retrieve vertices neighbors. Text graphs are generated as directed graphs, but in most cases we work with undirected graphs.

2.3 User Interaction with Semantics

Since we are building semantic network for relation discovery, this approach can be further enriched by user interaction with network, which can help to improve IE extraction and underlying semantic graph. We have described our early experiments in [26]. To summarize, we found out that by user operations such as deleting, merging, annotating or changing entity type user did a few operations, which had big impact on data quality and immediate impact on search results quality. User involvement is perceived as quite time consuming. However, if the users see the immediate impact in the form of better search results and better entities identification in text, they may be willing to do it. We would like to further extend the approach to an automatic creation of training datasets for IE machine learning approach.

2.4 Evaluation of Extracted Semantics

We have evaluated the rule-based information extraction approach in [22]. The focus was on extracting personal names, postal addresses, and telephone numbers from emails. We have evaluated NER on small set of English and Spanish emails. This evaluation was completed two years ago, and we did not provide a new evaluation on current datasets presented in the paper. Good information extraction results have a significant impact on relation discovery. In this paper we do not focus on this topic but rather on the relation discovery itself. However, we believe similar results as presented in [22] (success rate between 85 %–95 %) can be achieved in the discussed datasets with the help of user interaction.

In [22] we have also proved that a skilled information extraction expert can customize extraction rules for the application within a couple of hours with satisfactory results. In the future we would like to apply the machine learning approach for information extraction, in which the knowledge engineering approach will be combined with machine learning. The developer will define patterns in a similar way as now, but if several patterns match the same string in the text (e.g. a person or a location), the user will need to resolve which one is valid and when. Machine learning can help set up the probabilities of relevance match based on the training data coming from user interaction. The training set will cover positive examples (user annotations or changes of entity type) as well as negative examples (deletions).

2.5 Network Properties of Semantic Graphs

In this section we discuss the network properties of semantic graphs extracted from text data. They have similar properties to web graph, social networks, and information networks such as Wikipedia. The similarity of properties is important because it opens new possibilities for adapting similar algorithms and tools for information networks or for combination and integration of graph data from both structured and unstructured data sources, thus applying the same approach to relation discovery and semantic search. We illustrate details mostly through the graphs extracted from emails (Enron corpus), but we discuss other sources like semantic graphs extracted from web data (BBC, LinkedData, DSK), from a single monolithic document (Gorila), and also from event graph (agent simulation) and LinkedData (ACM publications) which can have different properties than information networks. The analysis of real-world networks has shown that they usually have several common properties, such as power law degree distribution, small-world property, and high clustering coefficient.

By *Small world networks*, we understand graphs in which any two random nodes can be connected by a relatively short path. These networks appear in many applications, but they are also typical for web graph, Wikipedia, or social networks. We compute several measurable network properties, described in detail in [27], namely:

Degree distribution: The degree of a node in a network is defined as the number of connections it has to other nodes, while the degree distribution is the probability distribution of these degrees over the whole network. In most cases, small world networks follow the power law degree distribution [40]. When this degree distribution is shown on the log scale, it can be interpolated by a linear function. It also forms a so-called long tail.

Complementary cumulative distribution function (CCDF): In CCDF we sum all degrees of nodes with the degree higher than a given degree (x). The advantage is that a long tail is transformed into a curve/line that can be more easily interpolated by a linear function.

Clustering coefficient: This is a property of graphs or networks which describes how much the nodes tend to interconnect to each other. There are three different measures for this coefficient: local, global, and the average clustering coefficient. The local clustering coefficient is a measure that expresses what proportion of the nodes neighbors are also direct neighbors. It is done by measuring the number of existing edges between the neighbors of a node. The clustering coefficient is equal to 1 when the node neighbors form a clique.

Assortativity coefficient: Degree assortativity coefficient (AC) denotes a tendency of nodes to be connected with other nodes of a similar degree. It is defined as the Pearson correlation coefficient of the degrees of pairs of nodes connected by an edge in the network [34]. According to [34], social networks

tend to have a positive assortativity coefficient so the networks are assortative, while networks such as internet, biological networks or other information networks tend to have negative assortative coefficient and we refer to them as being disassortative.

2.6 Text Graphs Properties

In this section we examine and discuss properties of several information networks.

In Figure 2 we provide the probability degree distribution for eight datasets. The node degree is on the x axis and the number of nodes with such a degree is on the y axis. The first dataset is the degree distribution of DBPedia. The other seven datasets were used in our experiments. We did not experiment with DBPedia graph, which has similar properties as some of the graphs with which we have experimented, but we do provide it as a typical representative of information networks. For the Enron, Gorila, and DSK graphs we see clear power-law degree distributions. LinkedIn can be also considered a power-law. For BBC, we notice a strange curve and also something like two independent datasets. We did not investigate this further, but it could be caused by processing BBC news as well as BBC country profiles. For the country profiles, we have extracted and identified more entities so that the topology of trees/graphs extracted from the country profiles is a bit different from the topology of graphs extracted from the news pages. When applying CCDF on BBC (Figure 3 left) we can see that the power-law distribution is quite valid, especially if nodes with too high or too low degree of distribution are not considered. Another dataset is the events dataset obtained from a log file of a multi-agent simulation. The degree distribution of the event dataset is not power-law, as we can also see on the right side of Figure 3. In addition we experimented with ACM LinkedData dataset[12]. It has similar properties to text graphs but it seems that the ACM dataset does not have the power-law degree distribution especially when considering its CCDF (right side of Figure 3). This is caused by the strange structure of the long tail seen in Figure 2, where we have quite a high number of hub nodes with various high degrees but always only one node for a wide range of the highest degrees. We experimented by deleting the nodes with a degree above 1 000 neighbors, and such a truncated network indeed behaved as a power-law one.

In Table 1 we provided the properties of graphs used in our experiments. We can see that all of the graphs have a high clustering coefficient of about 30 %. The average shortest path was computed for just a sample of random nodes, and it was between 5.5 to 7.5 hops. Events graph has very different properties concerning the average shortest path and the degree distribution. For the ACM dataset, we can see that most of its properties are similar to our text graphs. We have also computed the assortativity coefficient. Information networks should be disassortative (with a negative assortativity coefficient) [34]. Our hypothesis was that our

[12] `http://datahub.io/dataset/rkb-explorer-acm`

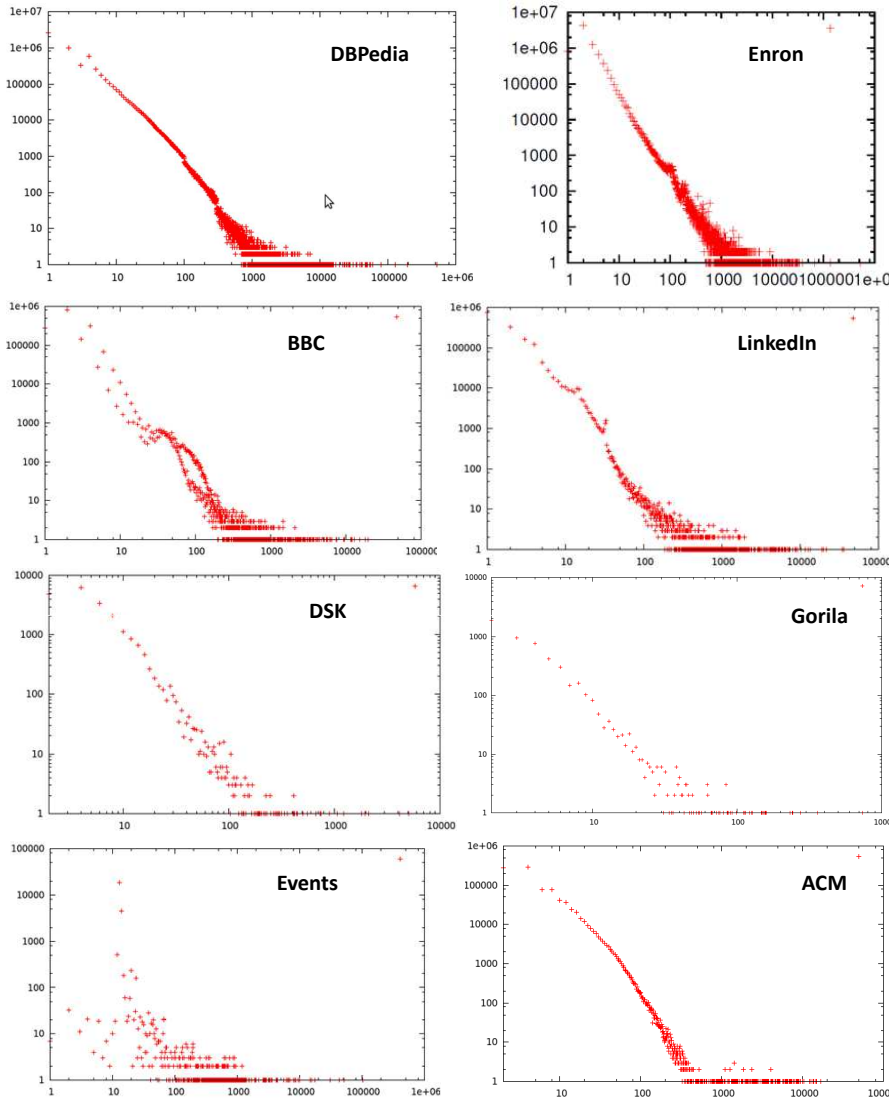


Figure 2. Log scale degree distribution on various graphs

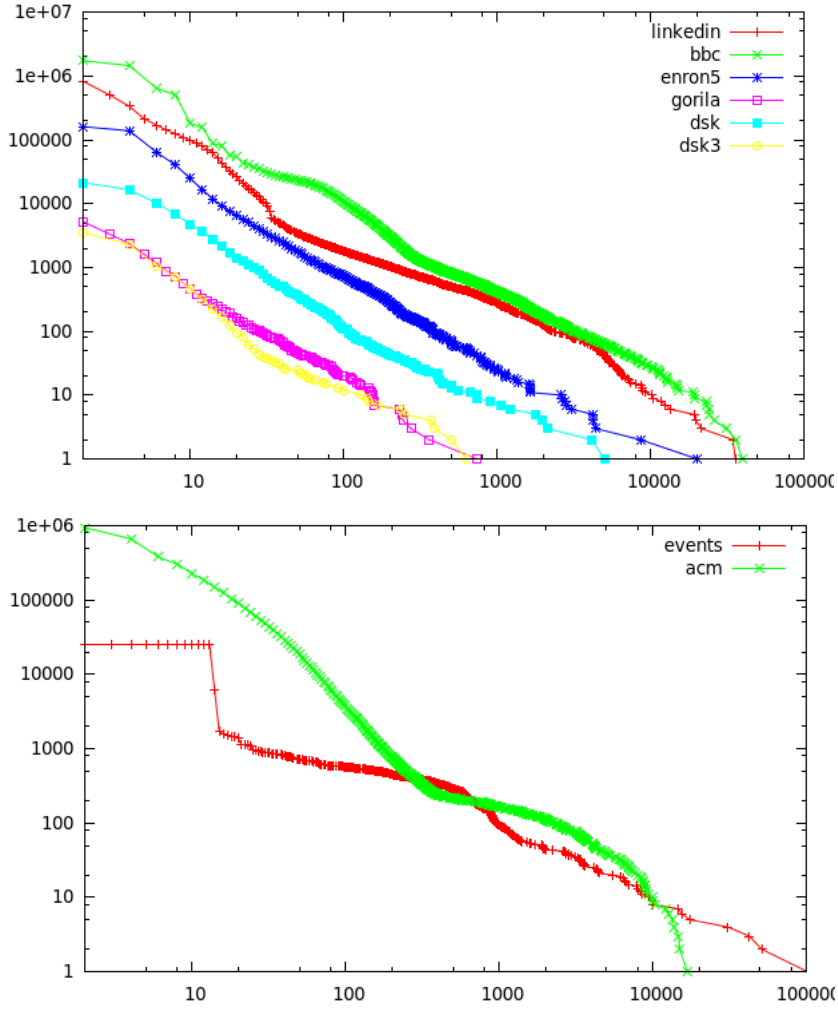


Figure 3. CCDF applied on degree distribution for text graphs (top) and non-text graphs (bottom) from experiments

relation search algorithm will work better for disassortative networks since we had problems with relation discovery in LinkedIn and Gorila networks which have positive assortativity coefficients. On the other hand, we have noted that the DSK network is assortative with quite a high assortativity coefficient and our relation discovery worked well there. It would then seem that assortativity does not have influence on the results, but more experiments are needed to confirm it. To conclude, in this section we examined the network properties of our generated text graphs. We could see that the extracted text graphs were not random graphs but

Graph/Experiment Name	Nodes	Edges	Average clustering coefficient	Assortativity coefficient	Average shortest path
Enron Full	8 269 278	20 383 709	0.29	−0.02	6.58
Enron5 (5 mailboxes)	160 387	630 330	0.30	−0.04	6.64
LinkedIn	1 564 698	6 094 634	0.36	0.13	6.48
BBC	1 725 900	6 839 358	0.34	−0.05	7.55
DSK	21 518	98 952	0.31	0.39	5.79
DSK3 (3 Wikipages)	2 857	8 754	0.36	−0.14	5.46
Gorila	5 959	23 724	0.31	0.03	6.25
Events (simulation)	25 478	539 328	0.38	−0.25	2.47
ACM Linked Data	941 322	2 198 001	0.34	−0.06	7.30

Table 1. Properties of graphs used in experimentation

had properties similar to other information networks known from various studies and exploited in applications. Since the properties of the networks were similar, we believe that the described relation discovery algorithm can be applied on both types of networks – those generated from unstructured texts, as well as the natively structured information networks. Moreover, it should also be applicable on the networks created by the integration of both kinds of data sources – structured and unstructured.

3 DISCOVERING ENTITY RELATIONS

We have described both how semantic text graphs are created and the properties they have. It is important to know the graph properties in order to apply search/discovery algorithm on such graphs. We believe much more work can be done on selecting appropriate graph algorithms for entity relation discovery on such graphs, since many algorithms applicable for small world networks can be used. However, in this paper we describe our experiments in applying spreading activation algorithm and its modifications on the semantic text graphs.

3.1 Spreading Activation Algorithm

The Spreading Activation method is a common approach for information retrieval [37] in semantic networks [7]. Spreading Activation algorithm can be implemented in many different ways, but the main idea is as follows: search is initiated from set of initial nodes with activation value, which is then propagated over connected edges to neighbouring nodes with several iterations. Nodes with top propagated value are the most relevant.

In our approach, we used spreading activation on the graph of a multidimensional social network in a similar way as IBM Galaxy [17], in which the concept of multidimensional social network for text processing was introduced. Spreading activation is

also used on the Slovak website Foaf.sk [39] for discovering relations between people and enterprises in the Slovak business register, in recommendation systems [42], and also in relation discovery in Wikipedia [4]. Spreading activation was also used for semantic desktop search [38]. Additionally, spreading activation was used on big semantic networks [28] (LinkedData) in the LarKC project, specifically focusing on scalable inference. In general, spreading activation identifies a smaller part of a semantic network for further logic based inference. In [8] the complexity of spreading activation is discussed. In small word networks one can clearly reach all the nodes in a graph within a few spreading activation iterations corresponding to the average shortest path. Thus, as in any graph algorithm, reasonable heuristics reducing the search space is needed. As we mentioned in the Introduction, random node access is the key problem for fast graph traversing [29], which is also used in the spreading activation algorithm. Simple Graph Database SGDB [5] was developed to be optimized for spreading activation. SGDB stores information about nodes and edges in an optimized form of key-value pairs.

In our previous implementation [23] we used an in-memory graph with the JUNG graph library, but we could not even load the full Enron Graph Corpus. Currently, we use SGDB on a single machine and achieve satisfactory results (Section 3.3 discusses the performance evaluation) on the whole Enron Graph Corpus, the biggest network we have experimented with. To the best of our knowledge, SGDB [5] is the best graph engine for real-time graph querying [6]. In our future work we would like to go further, creating a scalable graph querying solution on a shared-nothing architecture cluster.

When performing spreading activation we traverse only a part of the whole network, but this part grows quite fast with the depth of search since we deal with small world networks which have short paths between any randomly chosen nodes. After a few levels of activation, the spreading activation algorithm can reach the whole graph if the decay factor is not set properly. Therefore, we still need to optimize the spreading activation (or other relation discovery algorithm) even when a fast traversing infrastructure like SGDB is used. Most of the algorithms use modifications of Breadth First search and thus the depth of search needs to be optimized for each query. We have discovered experimentally that we cannot set up a common level of depth for different node relations discovery in information networks (such as the text graphs described in Section 2.5) to achieve both satisfactory relevant result and satisfactory performance because the graph topology is different in each case. One common factor that needs to be dealt with is the high-degree nodes.

In our algorithm, activation is started from a set of nodes ($S = \{v_1, v_2 \dots v_k\}$). The activation value is a constant ($n = 10\,000$) determined experimentally. The n value was set up to return results in reasonable time on tested hardware. For faster hardware it can be set up higher. The number does not have to be changed when dealing with smaller or larger datasets across all use cases. It is also a maximum number of visited nodes. Visited nodes are stored in the set V, which contains the starting nodes at the beginning ($V = S$). Starting nodes are put into the queue

$P = (v_1, v_2 \dots v_k)$. R is a set of nodes with assigned relevance, which is computed as n/k:

$$R = \{(v_1, n/k), (v_2, n/k) \dots (v_k, n/k)\}.$$

1. Because we traverse the graph using the Breadth First method, when the queue is defined as $P = (p_1, p_2 \dots p_l)$, we first take out the first node for processing $p = p_1$; $P = (p_2 \dots p_l)$. Then the queue is processed for each p until $P \neq \emptyset \wedge n > 0$. For each p, all of its neighbors are defined as a set N_p:

$$N_p = b : (p, b) \in E.$$

2. For each $b_i \in N_p$ we compute new relevance value of node $q_b = q_p / |N_p|$. We know the value q_b of node p because $(p, q_p) \in R$. We process the neighbours of p only if $q_b > threshold \wedge n > |N_p|$, otherwise the next node from P is processed.
3. Each b_i is added into queue $P = (p_2 \dots p_l, b_i)$, but only if it does not already belong to the set of visited nodes V. After processing, b_i is added to V.
4. If $(b_i, q) \in R$ then it is replaced by $(b_i, q + q_b) \in R$, otherwise $(b_i, q_b) \in R$.
5. When all b_i are processed, n is decreased by the neighbor count of node p:

$$n = n - |N_p| .$$

6. Then we process the next node $p \in P$ from the queue going back to the first step.

When the algorithm finishes, the set R contains the list of nodes relevant to the set of starting nodes (S) with assigned relevancy values (q_i) including the starting nodes.

$$R = \{(r_1, q_1), (r_2, q_2) \dots (r_n, q_n)\}.$$

In our algorithm we also define *OR* and *AND* operations over the starting nodes. *OR* operation is done exactly as we described, starting from multiple nodes. When using *AND* operation, we independently run algorithm for each starting node. For example, if running *AND* for two nodes, we get the following results sets:

$$\begin{aligned} R_1 &= \{(r_1, q_1), (r_2, q_2) \dots (r_n, q_n)\} \\ R_2 &= \{(s_1, g_1), (s_2, g_2) \dots (s_n, g_n)\}. \end{aligned}$$

In final result set for *AND* operation we include only those nodes which appeared in both sets, and the relevance value is computed by multiplying relevancies: $q = q_i g_j$

$$(r_i, q_i g_j) \in R \Leftrightarrow (r_i, q_i) \in R_1 \wedge (s_j, g_j) \in R_2 \wedge r_i = s_j.$$

As mentioned before, we use Breadth First traversing, which is limited to visit only n nodes. The algorithm skips the nodes with the higher degree (i.e. higher number of neighbor nodes) than the number of the remaining nodes to be visited. When a node

is skipped, we process the next node in the queue. We have experimentally set n (the maximum number of nodes to be visited) to 10 000 nodes to have a reasonable search time and satisfactory relevant results. The same number n is also used as the initial activation value, which is divided by the number of neighbor nodes in the next step. If we have more than one node as activation node, we also divide this initial activation by the number of starting nodes.

The algorithm finishes in reasonable times (around one second – based on the setting for the number of visited nodes) and still returns satisfactory relation results, but it can also fail especially if we want to compute the relations for the nodes with high degree. For example, if we would search for relations to the town of *Hudson* or to the state of *Texas*, such entities have too many connections in the Enron Graph Corpus. It does not make sense to infer entities related to *Texas*, but it can make sense to infer entities related to a concrete person as well as *Texas* at the same time. In our current approach, *Texas* would just be ignored. In our future work, we plan to improve our dealing with the high-degree nodes in the sense that we would include them in the relevant results if they were activated from low-degree nodes, but we would not let the high-degree nodes fire and pass their activation value to other nodes (otherwise the whole graph might get included in the results).

3.2 Graph Based Semantic Search – gSemSearch

In this section we describe our user interface called gSemSearch, which calls spread of activation search algorithm described in the previous section. The text of this section is based on [24].

To sum things up, the gSemSearch functionality and its user interface allow relation discovery, through which a user can perform a full-text search (e.g., *Gr***by* surname in Figure 4), select starting nodes (e.g., two variations of person names of *Michael Gr***by* and *UBS* company in Figure 4 on the left), and search for the related nodes. A list of nodes with mixed type is returned. It can be restricted to one node type by clicking on the selected type (e.g. *TelephoneNumber* in Figure 4 on the left). This will return nodes of the desired type as seen in Figure 4 (related phone numbers). Our prototype suggests that the starting nodes and the return results are related but does not suggest the type of relation. The type of the relation can be gauged by the user by clicking on the *Msg* links next to the result nodes in the list. This will highlight starting nodes in the most relevant email message in yellow and the selected node in red (note that same objects can be present in multiple messages), which can also be seen in Figure 4.

The search algorithm can also be improved by allowing the users to delete the wrongly extracted objects or to connect various aliases of the same object (e.g. the same company or personal names spelled differently) as seen in Figure 5. Such user feedback enables the search algorithm to learn and return better results in the future. For example, if we merge the three selected person name aliases seen in Figure 5, there will be better results (e.g. phone number, address or organization)

Figure 4. gSemSearch user interface

returned for any of the aliases. In addition, the gSemSearch user interface supports actions like node merging and deleting or changing the node type.

Figure 5. Prototype GUI with the results of full-text search and several objects (aliases) selected for possible merge or delete. When these object aliases are merged, subsequent searches return better results for any of them.

We also use a unique approach for synonymy and polysemy of the explored entities (ambiguity and disambiguation). If an entity is represented by more than one node (multiple aliases, similar to the person in Figure 4 or Figure 5), we can use two methods to explore the entities related to such an entity. We can either select all of the aliases and search for the nodes related to this node cluster (Figure 4), or we can merge the aliases to a single node (Figure 5) and explore its relations as if it were a single node. Another problem arises when the same string represents two different entities. We do not provide automatic disambiguation during the extraction, so two different people with same name will be presented as one node in

the graph. However, if some extra auxiliary information is known about the nodes, for example an address or company related to the person, the person node can be selected along with these related nodes, and the search can be performed for other entities related to this multiple selection. This way the sub-graphs related to the other person represented by the same named entity will either not be explored at all, or will be explored/activated only partially.

We have also tested the gSemSearch relation discovery on other data types like graphs extracted from BBC news, LinkedIn job offers, and event graphs of agent-based simulations, so we see the possibility of exploring our relation discovery approach and user interface in other domains in which data can be represented by graph/network structures with properties of information networks. This is discussed in Section 4.

3.3 Relevance Evaluation

In this section we evaluate the relevance of the proposed algorithm on various datasets using the information retrieval measures of Precision and Recall.

We have evaluated our approach on the relations between telephone numbers and organizations/people [19] using small set of English emails. The relations were identified with a precision of 76.9 % and a recall of 58.8 %. In the scope of the Commius project, we also tested the algorithm on a set of 50 Spanish emails [21]. The Information Extraction part of the evaluation was briefly described in Section 2.4. Because Information Extraction part was quite poor for Spanish emails, we achieved only 60 % precision for Spanish emails.

3.4 Precision of the Entity Relation Discovery

In [21] we have evaluated the success (precision and recall) of the IE and the success rate of relation discovery (the spreading activation algorithm) with satisfactory results [21]. The discovered relations precision was 60 % for the Spanish email dataset and 77 % for the English one. Interestingly, most of the errors were introduced by imperfect information extraction. When ignoring the information extraction errors, the relation discovery precision was about 85 % [21]. The algorithm tested on small datasets had a reasonable performance (search time) with acceptable results, but when it was applied on larger datasets we discovered the performance problems described in Section 3.5. The algorithm was subsequently optimized for faster performance. It is quite a hard task to evaluate how well the algorithm for relation discovery works on larger datasets. We decided to at least evaluate the precision of the returned results on BBC dataset. Recall could not be computed since we would have to go through all of the data manually.

In Table 2 we provide a summary of the evaluation experiment on BBC dataset. We have evaluated four types of relations for several queries. First, we have tried to return the list of relevant people for a concrete person (politicians from different countries in this case). When selecting a concrete person such as *Barroso*, we have

	P@1	P@5	P@5 part.	P@5 alias		P@1	P@5	P@5 part.	P@5 alias
Barroso	1	1	1	1	Nob Prize	1	1	1	1
Sarcozy	1	1	1	1	IMF	1	1	1	0.8
Fico	1	0.8	1	0.8	NATO	0	0.6	1	0.6
Cameron	1	1	1	0.8	EU	0	0	1	0
Merkel	1	0.8	0.8	0.4	L. Treaty	1	0.8	0.8	0.8
Tusk	1	0.8	1	0.8	EC	1	0.6	0.8	0.4
Person ⇒ People	100 %	90 %	97 %	80 %	NE ⇒ People	67 %	67 %	93 %	60 %
Slovakia	1	1	1	1					
Czech	1	1	1	0.8	Hungary	1	0.33	0.67	
Hungary	1	0.2	0.6	0.2	Poland	1	0.2	0.4	
Poland	1	0.33	0.67	0.33	UK	1	0.4	0.6	
UK	0	0.6	0.8	0.6	Ukraine	1	0.6	1	
Country ⇒ People	80 %	63 %	81 %	59 %	Country ⇒ City	100 %	38 %	67 %	
					Total	86 %	67 %	86 %	67 %

Table 2. Evaluation on BBC dataset

also selected all possible aliases such as *Mr. Barroso* or *Jose Manuel Barroso*, and then evaluated the returned relevant person list. In the "P@1" column we provide the precision rate for the first returned item in the result list. The "P@5" column is the precision rate for the first five listed results. In the "P@5 alias" column we examined the first five results and if the aliases of the same person appeared (e.g. *Mr. Dzurinda* and *Mikulas Dzurinda* returned for *Mr. Fico*), we considered more than five results grouping aliases together. In the "P@5 partially" column we evaluated the precision of the first five results, but we also considered partially relevant results. For example, in many cases one of the returned names was that of a journalist writing about the country, person, or organization in the query, or people related to the queried entity because of some events mentioned in the processed news. The Person-to-People precision was quite high, but we cannot be sure about recall. In the Country-to-People cases we had a problem detecting people related to Hungary or Poland; therefore, we did not get very good results since human names were also identified based on gazetteers of first names. We have used first names in English, Spanish, Italian and Slovak. Good results were returned for type-less entities (NE) to people relations. We did not have such good numbers for precise relation between entities, but when we examined the partially relevant results the precision rate was 93 %.

Based on the provided evaluation, it is hard to draw any conclusions on precise relevance of the returned results. From all of our experiments we can conclude that the entity relation search method gives good results in many cases, but it relies on the quality of the extracted named entities. In the BBC dataset, the named entity identification strategy was rather simple, producing many false entities and

many typeless named entities; however, with a little user effort to clean the data interactively (as discussed in Section 2.3) the results can often be substantially improved. This held true for most of the datasets with which we experimented. More serious problems were found when experimenting with LinkedIn and Gorila datasets. When searching the LinkedIn datasets, we wanted to infer job offers based on entities such as locations or skills which had a high node degree, but the algorithm was not able to search deeply enough. This problem occurred because the activation was stopped after a certain number of visited nodes had been reached. We will have to investigate how to deal with entities/nodes with high degrees, but currently we usually ignore them or the activation is stopped in such nodes. Please see more details in the LinkedIn use case description in Section 4.2. In the Gorila dataset, the main problem was that many entities had many different "name" nodes for the same person due to the rich morphology (inflexions) of the Slovak language. That is why it was hard to select appropriate nodes in the initial search when inferring results for a concrete person. This problem can be solved by improving information extraction so that it groups various morphology forms of the same entity together.

To conclude, spreading activation is a valuable method for finding relations among entities in information networks, which is confirmed not only by our experiments but also by other relevant work [17, 4, 42] in the field. Semantic text graphs have similar properties to other networks where spreading activation was tested. The challenges for text graphs are better information extraction and scalability. We tried to achieve better information extraction by applying simple extraction methods and then allowing users to interact and improve the data while searching for relevant results. By user interaction, we not only directly improve the results as well as enable better extraction in the following rounds of search, but also create possible training datasets for machine learning. In the future we intend to investigate various machine learning approaches. Scalability or performance is another challenge when working with large networks, since graph traversing is of an exponential nature. We will discuss this in the next section.

3.5 Performance Evaluation

In this section we summarize the performance evaluation of entity relation discovery in extracted networks. The performance problems were discussed in [23] and solved to some extent in [24], here we summarize our findings. Before examining the network properties of the extracted graphs (see Section 2.6), our hypothesis was that the performance (search time) should be stable even with large graphs because we always activate only a small portion of the graph. This was found to be valid only to an extent. One problem is that the created semantic graph has the properties of small world networks. For example, in a similar work performed on the Wikipedia graph [4], only two iterations of spreading activation could be performed before it would visit too many nodes. In [21] we have used 30 iterations, but in large graphs the impact on performance was too high. In the experiment presented in [23], we

have determined that the optimal number of iterations was four. This value seems to have little to no impact on the relevance of the returned results. The algorithm implemented in [23] seems to visit too many nodes even with four iterations; moreover, it visits the same nodes several times. When evaluating performance in [23], we worked only with small portion of Enron Corpus (from 3 000 up to 50 000 emails) and results were delivered sometimes even after 3 seconds. Later, we updated the algorithm [24] as described in Section 3.1. The current algorithm is able to deliver results within defined time (e.g. below 1 second, for the number of visited nodes set to 10 000). In [24] we provide a performance evaluation similar to the evaluation in [23] but on the full Enron Graph Corpus, while in [23], we tested the algorithm performance with 50 000 messages and less than 1 million nodes, but now the algorithm and infrastructure are capable of scaling up to 500 000 messages and 8 million nodes. The various selected types of entities evaluated in [24] represent a different topology of related sub-graphs explored during the graph traversal. For example, an email address is usually connected to many nodes directly, while a telephone number or address is connected to just a few sentence nodes. When searching for related nodes, different depths of graph traversing need to be explored for different object types. We achieved this by using the algorithm presented in the paper. As we mentioned earlier, the algorithm visited only n nodes while traversing the graph, where n was set experimentally to 10 000. Thus we see that the number of visited nodes is less than 10 000, and the number of unique visited nodes is even smaller. The search time is usually lower than 1 second, but it varies (from 171 ms to 1 195 ms in our experiment) based on the cached data of the underlying key-value store infrastructure.

4 USE CASES

In this section we briefly discuss the use cases in which our approach to entity relation discovery was tested. The properties of datasets associated with these use cases were already reported in Section 2.6. A deeper information on some of the use cases can be found in [27].

4.1 Email Communication

Our approach to entity relation discovery was started, developed and explored mainly in email communication. Since emails comprise both unstructured text (in message bodies) and a social network of communicating people (in message headers), we have tried to discover a common general method for entity relation search in email archives that would cover both the unstructured text and social networks.

Commius Project. The basis for our approach was the Email Social Network Search[13] [22] prototype which was developed in the scope of the Commius[14] project and extended after the end of this project. In Commius, we have tested the relevance of the approach in enterprise emails, described in [22]. In [22], you can also find a screenshot of the first prototype of relation discovery interface (Figure 4 in the paper).

Enron Email Corpus. Later when trying to analyze the Enron [11, 18] email corpus, whose size and properties are reported in Section 2.6, we had to deal with the slow performance of the algorithm on large graphs, causing us to modify the algorithms as described in this paper. Details are also available in [24]. Screenshots of the use case based on the Enron dataset are available in Section 3.2.

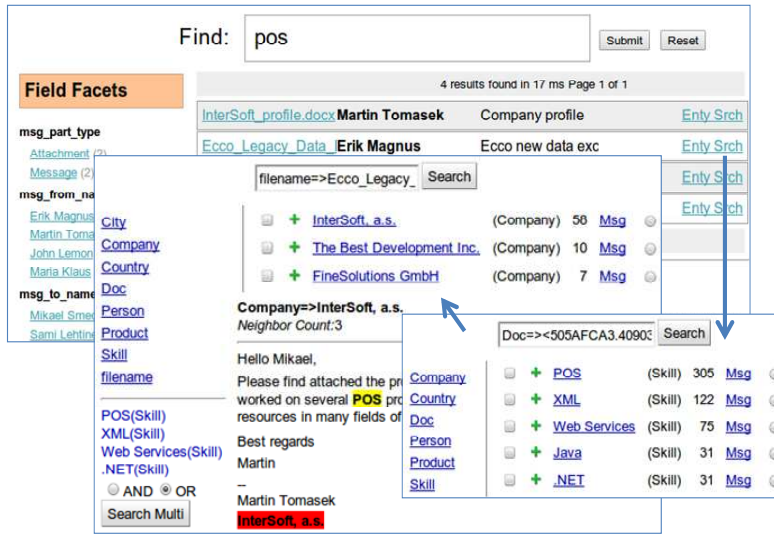


Figure 6. Prototype Enterprise search user interface

Email based Enterprise Search. We have tested the approach also for enterprise entity search based on email and document analysis, which can help small enterprises achieve their business tasks especially those fulfilling some information need. Proof of the concept implementation was provided within the VENIS[15] project in the InterSoft use case. This particular use case is related to the allocation of development resources to various providers and customers at the

[13] http://ikt.ui.sav.sk/esns/
[14] http://www.commius.eu/
[15] http://www.venis-project.eu/

same time. Customers usually ask large providers of software solutions to fulfill their complex projects. Since the providers often do not have all of the required development resources available, they need to find suitable subcontractors and involve them in the collaboration so as to complete the project for the customer.

In Figure 6 we see an early implementation of search functionality. There is a search field on the top of the screen which can be used to search for documents or emails using full-text search. Full-text search is integrated with an entity search, in which the related entities are displayed by clicking on *Enty Srch* link. The entity search and the recommendation have features as described in the paper and displayed in Figure 6 two front windows. The front-most one shows the skills detected in the development requirement email, while the one behind it shows the list of companies relevant for the skills. The idea is to return the relevant entities for one or more selected elements (i.e. context) and thus deliver the needed information for the business task represented by the email or document. More details can be found in [25] and a demo video is also available[16].

4.2 Entity Relation Discovery in Web Documents

In this section we discuss how the entity relation discovery approach can be applied to web documents. We will examine the following three use cases: BBC news, DSK and LinkedIn. While in the BBC and LinkedIn use cases we have crawled many pages from one website, in the DSK use case we have crawled web pages relevant to one topic returned by Google search.

BBC news. For the BBC use case, we crawled the BBC news portal. We crawled and processed about 19 000 news articles, in which we then identified about 100 000 entities such as people, organizations, countries, cities, or type-less named entities.

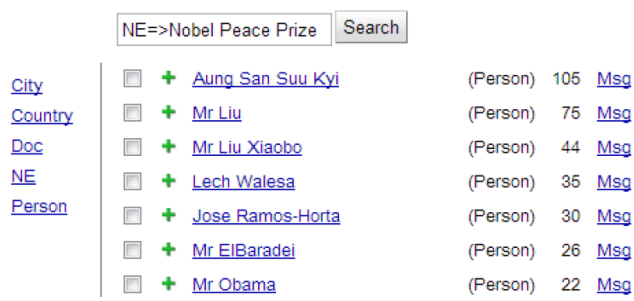


Figure 7. People relevant to the Nobel Peace Prize in the BBC Use case

[16] http://youtu.be/MSS3t__GLdk

In Figure 7, we have the list of relevant people returned for the type-less entity *Nobel Peace Prize*. As you can see, the return results are people somehow related to the Nobel Peace Prize. The top of the list clearly contains the prize winners. By the end of the list (not on the figure), though, other people start to appear such as members of the Nobel Peace Prize award committee or those involved in the award ceremony.

DSK use case. The DSK use case is related to the Dominique Strauss-Kahn case, involving a French politician and former director of the International Monetary Fund. For this case we crawled about 100 web pages relevant for the DSK case and tested the approach. The selected 100 pages were the first 100 Google results returned for the Dominique Strauss-Kahn query, and they included heterogeneous sources. Therefore, this use case demonstrates the universality of our approach with respect to heterogeneous web data. Here the relation discovery worked very well. The results and screen-shots are presented in [25].

LinkedIn Job Search. In this use case, we focused on crawling and searching LinkedIn job offers. We crawled more than 100 000 web pages, identified more than 70 000 job offers, and extracted more than 200 000 entities. The goal of the application was to match peoples CVs with the job offers. We used strategies based on full-text and faceted search as well as those for graph search described in this paper. Details of the application can be found in [9]. We discovered that the full-text search approach supported by facets based on the extracted entities was more successful than graph search. We believe this was due to the following reasons:

- Objects of interest were usually documents (CVs or job offers) and not the entities mentioned in the text.
- In the Skills category, extraction was not very successful.
- Locations were nodes with a high degree and they were usually starting points.
- The assortativity coefficient of the network was positive.

In order to improve the results, we modified the undirected graph to a directed one, where in most cases the edges were going in both directions. However, we have defined several types of restriction on the direction of the edges: $* \Rightarrow$ Money, Doc $\Rightarrow$ JobTitle, Doc $\Rightarrow$ DocTitle, City $\Rightarrow$ JobLocation, City $\Rightarrow$ Location. These restrictions are defined for vertex types (key in vertex graph definition) and help differentiate among the edge types although, formally, our algorithm works with typeless edges. We have defined these rules for directed edges in order to infer job properties such as salary, job title, and job location. In the case of undirected graph, these properties were not correctly inferred in most searches.

4.3 Gorilla

In the Gorilla Scandal[17] (Kauza Gorila in Slovak) use case we analyzed a leaked document from the Slovak Information Service. The document was one monolithic report. We have divided the report into smaller parts based on its paragraphs, and we have extracted human names, political parties, company names, dates, and amounts. Relation discovery works quite well in some cases. The biggest problem is the Slovak morphology in which many word forms for the same entity exist. In order to search properly we would need to group these forms. We can merge them manually, but an automatic or semi-automatic approach would be needed in order to improve the search. We noticed that the extracted network was assortative, similar to the LinkedIn network. We would like to further investigate whether a search algorithm works better on disassortative networks only. This will be part of our future work. More details and screenshots from this use case can be found in [27].

4.4 Entity Relation Discovery on Non-Text Graphs

The approach was also tested on graphs constructed from sources other than text, which is described below. We would like to show the possibility of future integration of structured and unstructured data in order to use the same approach for entity relation discovery.

Graph data form Agent-based Simulation. One of tested no-text graph was event data graph gathered from a multi-agent simulation of interaction between angry civilians and soldiers. It was performed as part of EUSAS[18] project which focused on creating a tool for the simulation of civilians and training military personnel for operations in urban environment. The method is used for the data analysis of simulation runs which explore interesting events in simulation in order to analyze Measures of Effectiveness (MoE) or to discover potential problems in the simulation model. For example, the number of injuries and fatalities are important MoEs, and therefore it would be very useful to be able to discover the underlying causes leading to them in order to prevent such situations. Some information on this use case and the application of graph analysis was also published in [41] and is also available at [27].

LinkedData Simplified Graphs. We have conducted several experiments on entity relation search also on LinkedData graphs. Concretely we have focused on the DBLP (Computer Science Bibliography) dataset as well as the ACM DL datasets. Experiments showed that we can return relevant entity relations; however, it became evident that the simplified graph structure with type-less

[17] `http://en.wikipedia.org/wiki/Gorilla_scandal`

[18] EUSAS – EDA project: European Urban Simulation for Asymmetric Scenarios (2010–2012) A-0938-RT-GC

edges is not sufficient to explore the rich relations described by LinkedData. In order to use our approach on LinkedData graphs, we would need to redefine graph structure to include, explore, and use labeled graphs (edges with types/properties). Since the DBLP dataset does not contain citations, we have focused more on ACM. The advantage is that through citations, one can get the related papers or authors for a paper or subject represented by the selected papers. More information on ACM graph simplification can be found in [33]. Through this experiment, we have demonstrated that our algorithm can also be used to some extent in finding relevant articles related to the selected authors, research fields, and articles. In the process of discovering relations, citation graph is exploited.

5 CONCLUSIONS

In this paper we have focused on entity relation discovery from unstructured text, where text was transformed into an information network with similar properties to other social or information networks. We have conducted experiments on several large networks and graphs extracted from diverse text resources as well as on structured data such as ACM publications. We have shown and evaluated an interactive method of relation discovery available in the gSemSearch prototype.

We believe the information networks, such as the graphs in our experiments, can help to interconnect unstructured and structured data such as text documents (web pages, emails, documents) with the structured data (hyper text networks, social networks, LinkedData). When structured and unstructured data possess similar properties of small world information networks, we can apply common algorithms for search and exploration of entities and their relations.

The proposed approach was evaluated on several use cases with quite large datasets in terms of quality and scalability performance and in terms of properties of generated semantic networks. Since the paper introduces novel approach of entity relation search using lightweight semantic text graphs, we did not compare it with existing relation extraction methods directly, because motivation and goal was to develop common interactive approach, which can handle both structured and unstructured data. To the best of our knowledge, none of existing methods goes in this direction. Nowadays, this is needed especially in enterprise content, where diverse structured and unstructured data (emails, documents, databases, social networks, or transactions) are available and they are not interconnected in any way to be searched and explored.

Acknowledgment

This work is supported by the projects VENIS FP7-284984, VEGA 2/0185/13 and CLAN APVV-0809-11.

REFERENCES

[1] Anyanwu, K.—Maduko, A.—Sheth, A.: SemRank: Ranking Complex Relationship Search Results on the Semantic Web. In Proceedings of the 14th International Conference on World Wide Web (WWW '05) 2005, ACM, New York, NY, USA, pp. 117–127, DOI 10.1145/1060745.1060766.

[2] Aggarwal, C. C. Ed.: Social Network Data Analytics. Springer 2011, 502 pp., ISBN 978-1-4419-8462-3.

[3] Agichtein, E.—Gravano, L.: Snowball: Extracting Relations from Large Plain-Text Collections. In Proceedings of the Fifth ACM Conference on Digital Libraries 2000, pp. 85–94.

[4] Ciglan, M.—Rivière, É.—Nørvåg, K.: Learning to Find Interesting Connections in Wikipedia. In Proceedings of the 2010 12th International Asia-Pacific Web Conference (APWEB '10), IEEE Computer Society, Washington, DC, USA 2010, pp. 243–249, DOI 10.1109/APWeb.2010.62.

[5] Ciglan, M.—Nørvåg, K.: SGDB – Simple Graph Database Optimized for Activation Spreading Computation. Proceedings of GDM '10 (in conjunction with DASFAA '10).

[6] Ciglan, M.—Averbuch, A.—Hluchý, L.: Benchmarking Traversal Operations over Graph Databases. Proceedings of GDM '12, IEEE ICDE Workshop 2012.

[7] Crestani, F.: Application of Spreading Activation Techniques in Information Retrieval. Artif. Intell. Rev. 11, No. 6 (December 1997), pp. 453–482, DOI 10.1023/A:1006569829653.

[8] Dix, A.—Katifori, A.—Lepouras, G.—Vassilakis, C.—Shabir, N.: Spreading Activation over Ontology-Based Resources: From Personal Context to web Scale Reasoning. International Journal of Semantic Computing, Vol. 4, 2010, No. 1, pp. 59–102.

[9] Dlugolinský, Š.—Šeleng, M.—Laclavík, M.—Hluchý, L.: Distributed Web-Scale Infrastructure for Crawling, Indexing and Search with Semantic Support. Computer Science Journal, Vol. 13, 2012, No. 4, pp. 5–19, http://dx.doi.org/10.7494/csci.2012.13.4.5.

[10] Fauscette, M.: The Future of Email Is Social. White Paper; IBM IDC report 2012, http://tinyurl.com/IBMRepFutureEmail2012.

[11] Chapanond, A.—Krishnamoorthy, M. S.—Yener, B.: Graph Theoretic and Spectral Analysis of Enron Email Data. Computational & Mathematical Organization Theory, Vol. 11, 2005, No. 3, pp. 265–281.

[12] Cunningham, H.: Information Extraction, Automatic. In: Keith Brown, (Editor in Chief), Encyclopedia of Language & Linguistics, Second Edition, Elsevier 2006, Vol. 5, pp. 665–677.

[13] Cunningham, H. et al.: Text Processing with GATE (Version 6). University of Sheffield, Department of Computer Science, 15 April 2011, ISBN 0956599311.

[14] Ferrucci, D. A.: IBM's Watson/DeepQA. ACM SIGARCH Computer Architecture News (ISCA '11), Vol. 39, June 2011, No. 3, DOI 10.1145/2024723.2019525, http://doi.acm.org/10.1145/2024723.2019525.

[15] Hassan, A.—Abu-Jbara, A.—Radev, D.: Extracting Signed Social Networks From Text. In Workshop Proceedings of TextGraphs-7 on Graph-Based Methods for Natural Language Processing, 2012, pp. 6–14.

[16] Iordanov, B.: HyperGraphDB: A Generalized Graph Database. In Web-Age Information Management, LNCS, Vol. 6185, 2010, pp. 25–36.

[17] Judge, J.—Sogrin, M.—Troussov, A.: Galaxy: IBM Ontological Network Miner. In Proceedings of the 1st Conference on Social Semantic Web 2007, Lecture Notes in Informatics (LNI), Vol. 113, 2007, pp. 157–160.

[18] Klimt, B.—Yang, Y.: Introducing the Enron Corpus. First Conference on Email and Anti-Spam (CEAS) 2004, http://www.ceas.cc/papers-2004/168.pdf, http://www.cs.cmu.edu/~enron/.

[19] Kvassay M.—Laclavík, M.—Dlugolinský, Š.: Reconstructing Social Networks from Emails. In J. Pokorný, V. Snášel, K. Richta (Eds.), DATESO 2010: Databases, Text, Specifications, Objects, Praha, MATFYZPRESS Publishing House 2010, pp. 50–59, ISBN 978-80-7378-116-3.

[20] Laclavík, M.—Šeleng, M.—Ciglan, M.—Hluchý, L.: Ontea: Platform for Pattern Based Automated Semantic Annotation. Computing and Informatics, Vol. 28, 2009, No. 4, pp. 555–579.

[21] Laclavík, M.—Kvassay, M.—Dlugolinský, Š.—Hluchý, L.: Use of Email Social Networks for Enterprise Benefit. In: IWCSN 2010, IEEE/WIC/ACM WI-IAT 2010, pp. 67–70, DOI 10.1109/WI-IAT.2010.126.

[22] Laclavík, M.—Dlugolinský, Š.—Šeleng, M.—Kvassay M.—Gatial, E.—Balogh, Z.—Hluchý, L.: Email Analysis and Information Extraction for Enterprise Benefit. In Computing and Informatics, Vol. 30, 2011, No. 1, pp. 57–87.

[23] Laclavík, M.—Dlugolinský, Š.—Kvassay, M.—Hluchý, L.: Email Social Network Extraction and Search. In NextMail 2011 Workshop, WI-IAT 2011, the 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, IEEE Computer Society 2011, pp. 373–376, ISBN 978-0-7695-4513-4.

[24] Laclavík, M.—Dlugolinský, Š.—Šeleng, M.—Ciglan, M.—Hluchý, L.: Emails as Graph: Relation Discovery in Email Archive. In Proceedings of the 21st International Conference Companion on World Wide Web (WWW '12 Companion), ACM, New York, NY, USA 2012, pp. 841–846, http://www2012.wwwconference.org/proceedings/companion/p841.pdf, DOI 10.1145/2187980.2188210.

[25] Laclavík, M.—Dlugolinský, Š.—Šeleng, M.—Ciglan, M.—Tomášek, M.—Kvassay, M.—Hluchý, L.: Lightweight Semantic Approach for Enterprise Search and Interoperability. In CEUR Workshop Proceedings: InteropVlab.IT 2012, CEUR 2012, Vol. 915, pp. 35–42. ISSN 1613-0073.

[26] Laclavík, M.: Improving Entity and Relation Discovery by User Interaction with Semantic Graphs. In 7th Workshop on Intelligent and Knowledge Oriented Technologies, Bratislava: Nakladateľstvo STU 2012, pp. 161–164, ISBN 978-80-227-3812-5.

[27] Laclavík, M.: Discovering Entity Relations in Semantic Text Graphs. Habilitation Thesis submitted for the Associate Professor degree, submitted on February 2013, defended October 2013, http://tinyurl.com/EntityRelationsTextGraphs.

[28] Grinberg, M.—Stefanov, H.—Stefanov, K.—Peikov, I.: D2.4.3 Spreading Activation Components V3. LarKC Project Deliverable, http://www.larkc.eu/wp-content/uploads/2008/01/LarKC-D2.4.3-Spreading-activation-components-v3.pdf.

[29] Lumsdaine, A.—Gregor, D.—Hendrickson, B.—Berry, J.: Challenges in Parallel Graph Processing. Parallel Processing Letters, Vol. 17, 2007, No. 1, pp. 5–20.

[30] Mihalcea, R.—Radev, D.: Graph-Based Natural Language Processing and Information Retrieval. Cambridge University Press 2011, ISBN: 9780521896139, 208 pp.

[31] Minkov, E.: Adaptive Graph Walk Based Similarity Measures in Entity-Relation Graphs. Doctoral dissertation, University of Texas at Austin 2008.

[32] Minkov, E.—Cohen, W. W.: Graph Based Similarity Measures for Synonym Extraction from Parsed Text. In Workshop Proceedings of TextGraphs-7 on Graph-Based Methods for Natural Language Processing, 2012, pp. 20–24.

[33] Mojžiš, J.—Laclavík, M.: Navigation in Simplified LinkData Graph (Navigácia v zjednodušenom LinkedData grafe). In 7th Workshop on Intelligent and Knowledge Oriented Technologies, Bratislava 2012, Nakladateľstvo STU 2012, pp. 15–18, ISBN 978-80-227-3812-5.

[34] Newman, M. E. J.: Mixing Patterns in Networks. Physical Review E, Vol. 67, 2003, No. 2, 026126, DOI 10.1103/PhysRevE.67.026126.

[35] Nuutila, E.—Törmä, S.: Text Graphs: Accurate Concept Mapping with Well-Defined Meaning. In Proceedings of the First International Conference on Concept Mapping 2004.

[36] Pantel, P.—Pennacchiotti, M.: Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ACL-44), Association for Computational Linguistics, Stroudsburg, PA, USA 2006, pp. 113–120, DOI 10.3115/1220175.1220190.

[37] Salton, G.—Buckley, C.: On the Use of Spreading Activation Methods in Automatic Information Retrieval. Technical Report, Cornell University, Ithaca, NY, USA 1988.

[38] Schumacher, K.—Sintek, M.—Sauermann, L.: Combining Fact and Document Retrieval with Spreading Activation for Semantic Desktop Search. In Proceedings of the 5th European Semantic Web Conference on the Semantic Web: Research and Applications (ESWC '08), Springer-Verlag, Berlin, Heidelberg 2008, pp. 569–583.

[39] Suchal, J.—Návrat, P.: Full Text Search Engine as Scalable k-Nearest Neighbor Recommendation System. Artificial Intelligence in Theory and Practice III IFIP Advances in Information and Communication Technology 2010, Vol. 331, 2010, pp. 165–173.

[40] Strogatz, S. H.: Exploring Complex Networks. Nature, Vol. 410, 2001, pp. 268–276, DOI 10.1038/35065725.

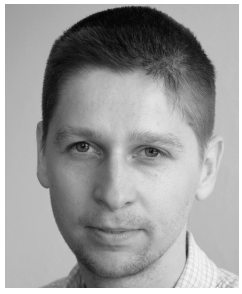
[41] Tavcar, A.—Gams, M.—Kvassay, M.—Laclavík, M.—Hluchý, L.—Schneider, B.—Bracker, H.: Graph-Based Analysis of Data from Human Behaviour Simulations. 10th IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI) 2012, pp. 421–426, DOI 10.1109/SAMI.2012.6209003.

[42] Troussov, A.—Parra, D.—Brusilovsky, P.: Spreading Activation Approach to Tag-Aware Recommenders: Modeling Similarity on Multidimensional Networks. In: D. Jannach, et al. (Eds.): Proceedings of Workshop on Recommender Systems and the Social Web at the 2009 ACM Conference on Recommender systems (RecSys '09), New York, NY 2009.

[43] Ulanoff, L.: Google Knowledge Graph Could Change Search Forever. `http://mashable.com/2012/02/13/google-knowledge-graph-change-search/`.

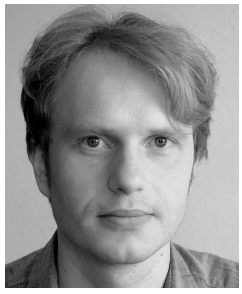
[44] Zhang, M.—Su, J.—Wang, D.—Zhou, G.—Tan C. L.: Discovering Relations Between Named Entities from a Large Raw Corpus Using Tree Similarity-Based Clustering. In Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP '05), Springer-Verlag, Berlin, Heidelberg 2005, pp. 378–389, DOI 10.1007/11562214 34.

[45] Zhu, J.—Nie, Z.—Liu, X.—Zhang, B.—Wen, J. R.: StatSnowball: A Statistical Approach to Extracting Entity Relationships. In Proceedings of the 18th International Conference on World Wide Web 2009, pp. 101–110.



Michal Laclavík is a researcher at Institute of Informatics, Slovak Academy of Sciences, and Data Scientist at Magnetic Media Online. In 1999 he received his M. Sc. degree in computer science and physics. He received his Ph. D. degree in applied informatics with focus on knowledge oriented technologies in 2006. He is the author and co-author of more than 100 publications with more than 200 citations, and participates in the Pellucid, K-Wf Grid and Commius European projects and several national projects. He has strong scientific and development expertise in email analysis, information extraction and information retrieval. He also gives lectures on information retrieval at Slovak University of Technology.

Štefan Dlugolinský is a researcher at the Institute of Informatics of Slovak Academy of Sciences. In 2009, he received his M. Eng. degree in information systems at the Faculty of Informatics and Information Technology, Slovak University of Technology. He is the author and co-author of several research papers and participates in European and national research projects. His research interests include email analysis and processing, information extraction and semantic annotation.



Marek Ciglan is a researcher at the Institute of Informatics of Slovak Academy of Sciences. He received his Ph. D. in 2008; he has spent two years as a Postdoctoral Fellow at NTNU, Trondheim, Norway. He has been working on several European and national research projects, has authored more than 30 research publications. His main research interests include semantic technologies and graph data management and processing. Concerning the graph data management, he is the author of a research prototype graph database; he is the author of one of the first graph benchmarking efforts and has published works on graph data mining.