Explanation Retrieval in Semantic Networks
Understanding Spreading Activation based Recommendations

Vanessa N. Michalke and Kerstin Hartig

Daimler Center for Automotive IT Innovations (DCAITI), TU Berlin, Ernst-Reuter-Platz 7, Berlin, Germany

Keywords: Explanation Retrieval, Spreading Activation, Pattern Recognition, Information Retrieval.

Abstract: Spreading Activation is a well-known semantic search technique to determine the relevance of nodes in a semantic network. When used for decision support, meaningful explanations of semantic search results are crucial for the user's acceptance and trust. Usually, explanations are generated based on the original network. Indeed, the data accumulated during the spreading activation process contains semantically extremely valuable information. Therefore, our approach exploits the so-called spread graph, a specific data structure that comprises the spreading progress data. In this paper, we present a three-step explanation retrieval method based on spread graphs. We show how to retrieve the most relevant parts of a network by minimization and extraction techniques and formulate meaningful explanations. The evaluation of the approach is then performed with a prototypical decision support system for automotive safety analyses.

1 INTRODUCTION

Recommender, advisory, and expert systems utilize available domain information or knowledge, often in order to help guide decision makers in decision-making processes (Kaklauskas, 2015). Semantic or associative networks may be used for representing such complex knowledge. Spreading activation algorithms support searching those nets semantically, providing information about relevance of specific nodes with respect to the specified search goals. Spreading activation algorithms base upon the wavelike distribution of activation values from nodes to neighbor nodes throughout the searched network. Consequently, a node's level of activation reveals its level of relevance in the current semantic search.

Semantic search results, such as the ones retrieved by spreading activation, can be used for providing recommendations and advices in decision support systems in various domains, e.g., medical diagnostics (Alvarez et al., 2011), or automotive safety analyses (Hartig and Karbe, 2016). In many applications, plausibility and understandability of the search results become important for users, i.e., the decision-makers, because they need to understand why a certain recommendation or advice was provided. Therefore, explanations of the results are crucial for the user's acceptance and trust in decision support systems. Current approaches usually identify relevant paths and environments by exploiting the original network. This neglects the valuable ancillary information accumulated during the search process. In this paper, we present a graph-based method for automatically generating explanations for semantic search results retrieved by applying spreading activation techniques. Our approach attempts to identify the nodes and edges that contributed most to the search result we try to explain. Therefore, we retrieve the most relevant extract not from the underlying semantic network but from a specific graph structure representing the spreading activation progress during search, i.e., the so-called spread graph.

The remainder of this paper is structured as follows. In Section 2, we examine related work in the areas of spreading activation, recommender systems and explaining information retrieval results. In Section 3, we present our three-step method for the automated generation of explanations. In Section 4, the fulfillment of explanation goals and the goodness of the generated explanations will be assessed. Finally, in Section 5, we summarize our results and present multiple possibilities to continue research.

2 RELATED WORK

In this section, we introduce the basics of searching semantic networks by spreading activation and its ap-

Michalke, V. and Hartig, K.
Explanation Retrieval in Semantic Networks - Understanding Spreading Activation based Recommendations.
DOI: 10.5220/0006050502910298
In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016) - Volume 1: KDIR*, pages 291-298
ISBN: 978-989-758-203-5

plication in recommender systems. We then outline recent research in the area of explanation generation as a meaningful extension for such systems.

2.1 Spreading Activation

One possibility for identifying and ranking relevant regions in semantic networks is spreading activation. In a comprehensive survey, Crestani provides an overview of its various applications for associative information retrieval (Crestani, 1997). Technically, starting from at least one start node, activation is passed to connected nodes. Thereby, the network is flooded wave-like, in so-called pulses, until a termination condition is satisfied. The spreading process is controlled by specific activation functions, allowing the calculation of different activation values, such as input and output activation as well as the activation level of each node. In each pulse, the activation value a node received is referred to as its input activation whereas the activation value distributed by a node is referred to as its output activation. Input activation combines activation values potentially transported by several edges. Therefore, the term transported activation of an edge corresponds to its actual activation value passed in a given pulse. The activation level of a node represents its current relevance level in the search. Thus, after applying spreading activation, relevant regions of a network can be identified, enabling application-specific interpretation.

2.2 Recommendations

Today, web-based recommender systems are well-established, e.g., providing useful suggestions for products or videos (Jain et al., 2015). One system that recommends items by using spreading activation is WebSCSA (Web Search by Constrained Spreading Activation), as described in (Crestani and Lee, 2000). WebSCSA identifies relevant web pages based on marked pages during a user's web search. Besides web-based applications, recommendations in advisory systems support the management of complex domain-specific knowledge in enterprises. Such advisory systems may support decision-makers in diverse domains, e.g., medical diagnostics (Alvarez et al., 2011), or automotive safety analyses (Hartig and Karbe, 2016). The importance of high-quality decisions makes explanations especially valuable. Therefore, such systems are potential candidates for the explanation retrieval method presented in this paper.

2.3 Explanations

The user needs to understand the reasons for a specific recommendation to assess its quality (Sinha and Swearingen, 2002). Explanations can support this understanding. Reconstructive explanation generation, as described in (Wick and Thompson, 1992), is considered an active problem-solving process to explain any results received by a problem solving component. Based on the problem solving computations, i.e., the so-called line-of-reasoning, it generates a so-called line-of-explanation. For this purpose, the prototype REX (Reconstructive Explanation System) uses the shortest path identified by the A* algorithm (Wick and Thompson, 1992). The method, presented in this paper follows the general concepts of reconstructive explanation generation, but focuses on spreading activation as one particular problem solving technique.

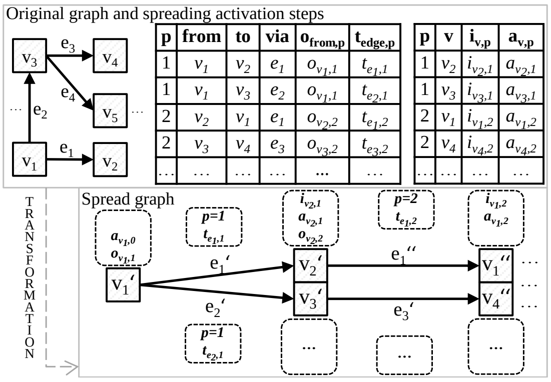
In (Forcher et al., 2010), explanations are also created by using the shortest path. However, other criteria than the path length must be considered when searching for the best explanation. (Aleman-Meza et al., 2005) rank the relationships in semantic networks by using configurable semantic and statistical criteria like subsumption or popularity. Similar criteria are used for the assessment of relevant paths, identified by a modified bidirectional breadth-first search (Viswanathan and Krishnamurthi, 2012) and a semantic ant colony optimization algorithm (Viswanathan and Krishnamurthi, 2015). When explaining spreading activation results, an assessment by the presented criteria is generally applicable. However, we assume that not limiting explanation retrieval to paths can be of greater value for meaningful explanations.

In contrast to path-based approaches, subgraph-based approaches consider potentially relevant elements across the entire network. Many subgraph-based approaches extract connection subgraphs from the original graph and utilize it for their explanation deduction (Faloutsos et al., 2004; Forcher et al., 2011; Forcher et al., 2012). Connection subgraphs aim at describing the relationships between nodes after particular relevance analyses, e.g., including the application of goodness criteria, keyword assignment, or neighborhood expansion. However, these approaches utilize the original graph for relevance assessment. Since semantic search by spreading activation is a means of relevance assessment of mutual nodes, we suppose the answer to the question why a node is highly relevant to another to ground on the spreading activation process itself. Therefore, we propose to utilize the valuable spreading activation data, i.e., our spread graph, a specific graph-based structure representing logged spreading activation process data.

3 METHOD FOR EXPLAINING RECOMMENDATIONS

In this paper, we present a method for explaining spreading activation based recommendations, which takes advantage of available additional information from the spreading activation process itself. The method aims at explaining the reason why a recommendation, i.e., a result node in the searched semantic network, is considered to be of relevance for the given start nodes of the spreading activation process. Therefore, we utilize the data produced during spreading activation and represent it in a dedicated spread graph structure. A *spread graph* is a directed graph illustrating the spreading activation process on the original graph. Nodes in the spread graph represent the state of their corresponding original graph's nodes in a certain spreading pulse. Each edge represents one activation value distribution step via the corresponding original graph's edge. The source(s) of a spread graph are the start node(s) of the corresponding spreading process.

Figure 1 depicts the transformation from the original graph to the spread graph as well as two tables illustrating the performed spreading activation steps. In the left table, each row describes one activation distribution from one node to another in a pulse p including output $o_{from,p}$ and transported activation $t_{edge,p}$. The right table shows the input activation $i_{v,p}$ and activation values $a_{v,p}$ for each node after each pulse. The spread graph combines both the elements from the original graph and the unrolled spreading activation steps. Therefore, the spread graph may contain more than one corresponding element to one original element, e.g., v'_1 and v''_1 correspond to v_1.



p	from	to	via	$o_{from,p}$	$t_{edge,p}$
1	v_1	v_2	e_1	$o_{v_1,1}$	$t_{e_1,1}$
1	v_1	v_3	e_2	$o_{v_1,1}$	$t_{e_2,1}$
2	v_2	v_1	e_1	$o_{v_2,2}$	$t_{e_1,2}$
2	v_3	v_4	e_3	$o_{v_3,2}$	$t_{e_3,2}$
...	...	...	...	...	...

p	v	$i_{v,p}$	$a_{v,p}$
1	v_2	$i_{v_2,1}$	$a_{v_2,1}$
1	v_3	$i_{v_3,1}$	$a_{v_3,1}$
2	v_1	$i_{v_1,2}$	$a_{v_1,2}$
2	v_4	$i_{v_4,2}$	$a_{v_4,2}$
...	...	...	...

Figure 1: Transformation to spread graph.

Knowledge-based recommender systems often utilize ontologies for knowledge representation, formulated in different RDF-based target languages, from plain RDF to the more expressive Web Ontology Language. However, they all base on an underlying RDF Graph, which can be visualized as directed graph, where the contained statements are represented by nodes and directed edges (RDF, 2014). The approach presented in this paper is applicable to any target structure based on RDF. Usually, a recommendation retrieved by spreading activation is characterized by the highest activation value. However, since the presented approach is capable of explaining every spreading activation result, the term result node specifies the node an explanation is required for.

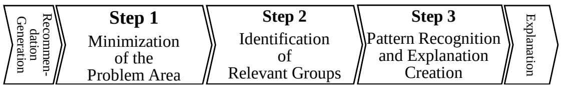

Recommendation Generation
Step 1
Minimization of the Problem Area
Step 2
Identification of Relevant Groups
Step 3
Pattern Recognition and Explanation Creation
Explanation

Figure 2: Method overview.

Figure 2 illustrates our three-step method for the explanation of a given recommendation. Initially, we apply a minimization technique to remove all nodes from the spread graph that are strictly irrelevant for the explanation. On base of the resulting minimized spread graph the most relevant subgraphs, so-called groups, are identified in a second step by using complex relevance detection techniques. In a third step, group-based explanations are created using pattern recognition. In the following sections, we provide a detailed description of each step together with a continuous example.

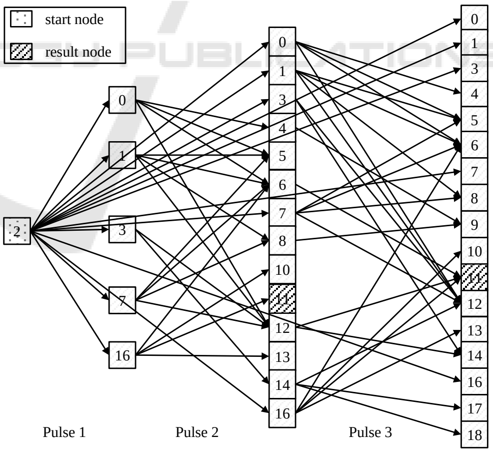


Figure 3: Example spread graph.

Example: In Figure 3, an example spread graph is depicted representing the states of each node at each pulse in a three-pulse spreading activation process. For example, node 3 receives activation from node 2 in the first pulse. The objective of this work is to explain the detected high relevance of result node 11 for start node 2. The corresponding original graph and the remaining spreading data are not depicted since they are not required for understanding the example.

3.1 Minimization of the Problem Area

Since a spread graph represents the entire spreading activation process, it may contain unnecessary information for the explanation of concrete result nodes. Nodes and edges in the spread graph that do not contribute directly or indirectly to the activation of a result node, we consider to be *strictly irrelevant*. This applies to all nodes and edges that are not part of any path between start and result nodes. We refer to them as *dead-end elements* and propose their removal in order to minimize the problem area. As a result, we retrieve a minimized spread graph containing only potentially relevant nodes and edges. Figure 4 illustrates various dead-end elements, depicted as dashed lines.

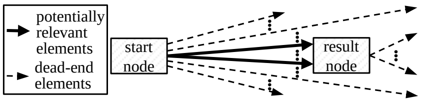


Figure 4: Minimization by backtracking.

Technically, the minimization follows the idea of impact analysis. Starting at the result nodes, all paths are tracked back in the edges' opposite direction until any start node is reached. All elements that are not part of the detected tracks are dead-end elements and can be ignored safely. The paths of the backtracking form the resulting minimized spread graph. During minimization, the activation values of the spread graph are transferred to the corresponding elements in the minimized spread graph.

Example: Figure 5 shows the result of minimization by dead-end removal for the spread graph in Figure 3. The minimized spread graph only contains potentially relevant information for the explanation of the result node 11. For example, node 8 (located between pulse two and three) is not part of any path that starts at start node 2 and ends at any result node 11. That means, node 8 did not contribute anything to the relevance of node 11 and can be removed.

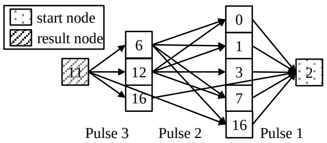


Figure 5: Example minimized spread graph.

3.2 Identification of Relevant Groups

Although a minimized spread graph is usually smaller than its corresponding spread graph, it still tends to be too large in order to create a meaningful and concise textual explanation. Therefore, the goal of this step is the identification of the most relevant subgraphs, so-called groups, of a minimized spread graph. Their detection strongly relies on the analysis of previously collected activation values assigned to the (minimized) spread graph. This data allows conclusions about the relevance of nodes and edges at different pulses. To identify such groups, we present two substeps: selecting the most relevant nodes and determining the most relevant paths and groups around them.

3.2.1 The Most Relevant Nodes

Since each node in the minimized spread graph carries a certain relevance, considering all of them in an explanation poses a very complex task. In order to reduce complexity, we introduce the additional parameter $n > 0$ to define the expected number of most relevant nodes. The precision n can be configured by a user or, alternatively, can be automatically determined based on specific properties of the given minimized spread graph, e.g., the number of represented pulses.

Potential *candidates* for most relevant nodes are all nodes in a minimized spread graph that are neither start nodes, result nodes, or semantically meaningless nodes such as class or property nodes in RDF-based semantic networks. Additionally, an application-specific filtering of the candidate nodes set is possible. The *relevance* of a candidate node is indicated by its relevance value r. The calculation of r is based on its input activation because it composes all transported activation values to the candidate in the given pulse and, therefore, denotes its relevance. Furthermore, two properties of the surrounding local network structure, i.e., connectivity and existence of class nodes, are used for the calculation. In the following, both are introduced in detail.

First, class nodes only represent the type of an instance node, for example *Ella* is of type canary. A name of an instance can be used together with its type to raise the comprehensibility of an explanation. However, types do not add further meaningful information to explanations. For that reason, the transported activation from these nodes to the candidates is subtracted from the input activation of the candidates, which leads to a reduced relevance value. Additionally, the relevance of the candidates that activate class nodes is slightly reduced to take into account their special relationship to class nodes.

Second, the connectivity of a candidate influences its relevance. A strong distribution of activation is an indicator for a very broad semantic meaning (Crestani and Lee, 2000) and, therefore, indicates less relevance for an explanation. On the other side, highly connected nodes, which receive activation over many edges, tend to get high input activation values. Nevertheless, candidates with lower input activation values might be more relevant for an explanation, e.g., when the activating edges carried more semantic meaning. In Figure 6(a), node *x* receives activation by only two nodes, whereas in Figure 6(b), node *y* receives activation by many nodes. The overall input activation *i* of node *y* exceeds the input activation of node *x* by 1. However, node *x* should be considered to be more relevant for an explanation than node *y*. In order to compare highly connected nodes with other nodes, the ratio of in- and outgoing edges is considered during relevance value computation.

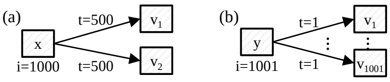


Figure 6: Highly connected nodes.

The relevance of each node is calculated by defined relevance functions based on its input activation with respect to the beforementioned properties. Then, the *n* nodes with the highest relevance values from the candidate set represent the *most relevant nodes*.

Example: The input activation and the calculated relevance values for all candidates in the minimized spread graph are shown in Figure 7. Start node 2 and result node 11 are by definition no candidates as well as the class nodes 16. Nodes 12 and 3 are the resulting relevant nodes due to their highest relevance values. Note, that node 6 has a higher input activation than node 3. However, regarding the relevance value, node 3 outperforms node 6 because of the connectivity.

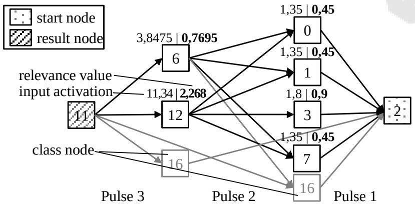


Figure 7: Weighted example minimized spread graph.

3.2.2 The Most Relevant Paths and Groups

The detected most relevant nodes do not form an explanation yet due to missing information about their relationship to the start and result node(s). This relationship can be clarified through the paths leading from the result node via a relevant node to a start node in the minimized spread graph. The problem of finding such paths can be transformed into the two problems of finding a set of paths between the result and relevant node and between the relevant and the start node(s), as shown in Figure 8. We define an additional parameter $m > 0$ for each subproblem, which restricts the number of paths to identify in order to reduce the explanation length. This value is either user-specified or based on specific properties.

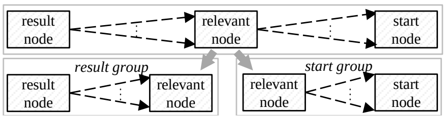


Figure 8: Dividing the path-finding problem.

For the explanation, we search for the most expressive and, therefore, the most relevant paths. Since the transported activation reflects the relevance of an edge, the relevance of a path can be determined by the sum of transported activation values along its edges. Thus, the most relevant paths are the longest paths regarding their overall transported activation. Generally, finding a longest path is NP-hard (Uehara and Uno, 2004). However, directed acyclic graphs, such as (minimized) spread graphs, allow an efficient solution of the problem. In (Sedgewick and Wayne, 2011), the authors propose the determination of longest paths by using a shortest path algorithm with negated edge weights. We apply this approach to a Bellman-Ford algorithm. In contrast to original graphs, spread graphs support the efficient longest path search. This is an advantage of the approach presented in this paper.

Each identified set of most relevant paths for one relevant node can be represented as a subgraph of the minimized spread graph. We call such subgraphs groups, whereas the group from a result node to a relevant node is called result group and the group from a relevant node to start node(s) is called start group (see Figure 8). Distinguishing between these two groups is a benefit, because it allows customized explanations for their diverging meaning. While the start group explains the origin of a relevant node, i.e., the recommendation request, the result group explains its effect, i.e., recommendation.

Example: Based on the minimized spread graph, the start and result groups for the relevant nodes 12 and 3 are illustrated in Figure 9. For this example, the maximum number of paths per group (parameter *m*) is restricted to 2. Therefore, the start group of node 12 does not contain all potentially relevant paths.

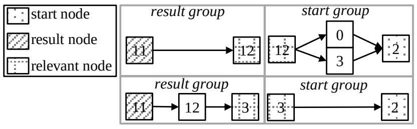


Figure 9: Example groups for the relevant nodes.

3.3 Pattern Recognition and Explanation Creation

The benefit of group-based explanations is that we do not depend on explaining paths as one-dimensional statements but can take advantage of cross-path information within the graph. Pattern analysis allows to linguistically subsume elements in an explanation that structurally belong together in the underlying group. This prevents confusing explanations and supports conciseness, compactness, and comprehensibility.

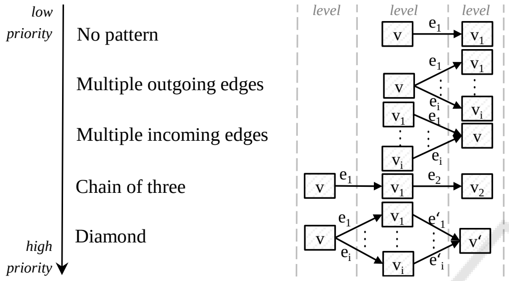
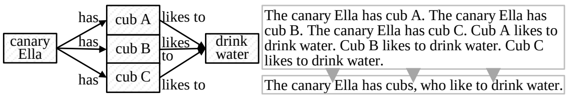


Figure 10: Patterns.

We identified the patterns in Figure 10, which are prioritized regarding their expressiveness: from low priority for no specific pattern to highly prioritized diamonds. If no special pattern can be identified, two nodes and their connecting edge can be interpreted as a simple statement of subject, predicate, and object, as common for semantic networks. In case one node has at least two outgoing edges to other nodes, the pattern of multiple outgoing edges applies. Similarly, there is a pattern for multiple incoming edges. The chain of three pattern applies whenever there is a sequence of three nodes connected by unidirectional edges. The diamond pattern describes two or more equally long downward paths that split in the first node and rejoin in the last node. However, the level of depth must be exactly 3 and the nodes of a level and the edges between the same levels have to be of the same type. A less restrictive concept for diamond structures was introduced by (Furnas and Zacks, 1994) in the area of multi-tree analysis. Since all other patterns are included in a diamond, it contains most structural information. This makes diamond structures especially expressive and valuable for explanations. After the proposed pattern recognition, the actual *sentence building process* is straightforward. In this work, English language is used. However, language-specific application is possible. Sentence building follows two basic principles. First, as many nodes and edges as possible, that are located on the same depth level in the pattern, should be subsumed, e.g., utilizing connection words such as *and* or using types. Second, the different levels of a pattern need to be concatenated to form a sentence, e.g., utilizing connection words such as *which*. The generated textual explanation supports the user to comprehend the reasons for a given spreading activation based recommendation.

Example: Figure 11 contains a sentence building example for the described diamond pattern, which allows to build only one short and more expressive sentence instead of six sentences.



The canary Ella has cub A. The canary Ella has cub B. The canary Ella has cub C. Cub A likes to drink water. Cub B likes to drink water. Cub C likes to drink water.

The canary Ella has cubs, who like to drink water.

Figure 11: Sentence building example based on diamonds.

4 EVALUATION

Evaluating explanations is very challenging since the most interesting evaluation criterion refers to usefulness. Usefulness can only be assessed subjectively, mostly depending on the experience and preferences of the observer (Klahold, 2009). In this paper, two different evaluation approaches are utilized to show the explanations' usefulness. The first approach argues on the fulfillment of selected explanation goals as proposed by (Tintarev and Masthoff, 2011). The second approach evaluates the goodness of explanations by estimating precision, recall, and F-measure. These measures are common for assessing the goodness of recommendations and are applicable to explanations as well (Klahold, 2009).

4.1 Fulfillment of the Explanation Goals

In (Tintarev and Masthoff, 2011), the authors propose an explanation goal dependent evaluation of explanations with goals like transparency, effectiveness, and efficiency. Furthermore, they provide guidelines for the application-specific selection of goals and their evaluation. In this work, we focus on examining the goals transparency as well as the support of efficient and effective decisions for users.

Transparency: The proposed explanation generation process follows a strict and reproducible process, utilizing different intermediate graph structures. Each intermediate graph is based on its predecessor and their mutual relationships are defined by transformation rules. Therefore, an explanation is traceable backwards through the intermediate graph structures to the spread graph depicting the spreading activation process. The result node as well as its corresponding nodes can be identified in each single process step.

This supports the user's understanding of how the system works. We therefore consider the proposed explanation generation process to be transparent.

Effectiveness and Efficiency: Usually, original graphs contain much information and spread graphs tend to be even larger. For example, an original graph examined in this paper with more than 26 000 nodes and 138 000 edges leads to a spread graph of 108 000 nodes and 136 000 edges after 6 pulses. Manual traceability of this large amount of data is infeasible. In contrast, the minimized spread graph only contains 90 nodes and 190 edges, the groups even consist of only 11 nodes and 10 edges. That means, the user retrieves an immensely reduced data set, which supports an effective and efficient decision making process.

These arguments show that the generated explanations are adequate to their purpose. The explanations are transparent and support effective and efficient decisions, which contributes to their usefulness.

4.2 Goodness of the Explanations

To evaluate the explanations' goodness, we apply the proposed explanation generation method to results from the HARvESTer recommender system, which supports safety experts when performing automotive safety analyses (Hartig and Karbe, 2016). There, semantic networks represent real data of completed automotive safety analyses by an automobile manufacturer. For each node and edge of such networks, the information whether or not it is expected to be part of an explanation for a given recommendation is assigned[1]. Then, an explanation is generated for the recommendation using the proposed explanation method. Both results can then be compared regarding their precision, recall, and calculated F-measure. For the evaluation, we examine three different requests, e.g., inquiring malfunctions based on user-given functions, with 4 to 6 pulses. They are performed on four semantic networks of different sizes listed in Table 1.

Table 1: Sizes of examined networks.

Networks	A	B	C	D
# nodes	1800	775	1218	26432
# edges	8779	3426	5319	138773

The *precision* for each request and network is shown in Figure 12 (left), grouped by nodes (light gray) and edges (dark gray). The precision results range from 33% to 100%, where 100% means that the generated explanation contains no false positive element, i.e., unexpected node or edge. The lower precision rates, e.g., for network B, are caused by two main reasons. First, multiple nodes and edges are often of similar importance. Second, the size of the explanation is restricted by the presented parameters n and m. Thus, it is possible that only some of several nodes and edges of similar importance are chosen for an explanation while other expected nodes and edges are missing. This explains that receiving 100% is not always possible. The *recall* for each request and network is shown in Figure 12 (right). Results range from 60% to 100%, where 100% means that all expected nodes or edges are part of a generated explanation, e.g., for request 1 on network C. Lower recall results are mainly caused by similar important nodes and edges, resulting in similar good explanations.

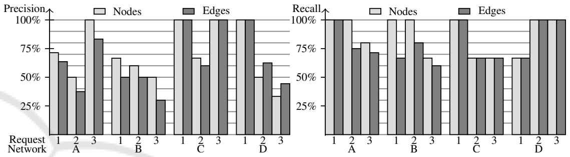


Figure 12: Precision and recall for nodes and edges.

The quality of an explanation benefits more from a high rate of expected elements in the explanation than it loses by additional nodes and edges. Therefore, recall can be considered to be more important than precision, i.e., represented by a F_2*-measure*. Its results range between 62% and 100% with an average of 84% for nodes and 78% for edges (excluding outliers). The evaluation shows, especially underlined by the high average F-measure, that the content of the generated explanation generally corresponds to the expectations. Besides the explanation generation method, the goodness of an explanation strongly depends on the different user preferences, the expressiveness of the underlying knowledge base and the configuration of the spreading activation algorithm.

5 CONCLUSION AND OUTLOOK

In this paper, we presented a method to retrieve explanations for semantic search results obtained by spreading activation techniques. The novelty of our approach is the excessive exploitation of the data accumulated during the search process instead of utilizing the original network. Therefore, our method is performed on a specific graph representing the spreading process, the so-called spread graph. We presented our explanation retrieval method in three steps. Firstly, we introduced a minimization tech-

[1] Remark: The assignment of the expected values are determined by one advanced user. Certainly, the evaluations' reliability can benefit from an assignment by a safety expert group. For this work, an expert group is not available. This is planned for extensive future case studies.

nique for the spread graph in order to remove strictly irrelevant parts. Secondly, we presented the retrieval of the most relevant subgraphs, so-called groups, from the minimized spread graph. Lastly, we described pattern recognition techniques to facilitate meaningful and concise explanation verbalization in natural language. The evaluation of the approach showed promising results. For the examined semantic networks, we were able to highlight both the fulfillment of the explanation goals and the goodness of the generated explanations. However, we see much potential for future research. Since the identification of relevant parts in the spread graph is based on very complex relations, our approach can be refined, e.g., by an additional consideration of more complex neighborhood influences. A more extensive pattern analysis can improve the conciseness of generated explanations. We furthermore plan on extended case studies in real-world application environments to see the benefit of the provided explanations, e.g., the increasing trust in recommendations. Especially, an explanation goodness comparison with existing approaches can emphasize the benefit of our approach.

REFERENCES

Aleman-Meza, B., Halaschek-Weiner, C., Arpinar, I. B., Ramakrishnan, C., and Sheth, A. (2005). Ranking complex relationships on the semantic web. *IEEE Internet Computing*, 9(3):37–44.

Alvarez, J. M., Polo, L., Jimenez, W., Abella, P., and Labra, J. E. (2011). Application of the spreading activation technique for recommending concepts of well-known ontologies in medical systems. *Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine (BCB '11)*, page 626.

Crestani, F. (1997). Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482.

Crestani, F. and Lee, P. L. (2000). Searching the web by constrained spreading activation. *Information Processing and Management: an International Journal - Artificial Intelligence and Information Retrieval*, 36(4):585–605.

Faloutsos, C., McCurley, K. S., and Tomkins, A. (2004). Fast discovery of connection subgraphs. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '04)*, pages 118–127, New York. ACM.

Forcher, B., Agne, S., Dengel, A., Gillmann, M., and Roth-Berghofer, T. (2012). Towards Understandable Explanations for Document Analysis Systems. In *10th International Workshop on Document Analysis Systems(DAS) IAPR*, pages 6–10, Gold Cost.

Forcher, B., Roth-Berghofer, T., Sintek, M., and Dengel, A. (2010). Constructing Understandable Explanations for Semantic Search Results. In *17th International Conference, Knowledge Engineering and Management by the Masses (EKAW)*, pages 493–502, Lisbon. Springer.

Forcher, B., Roth-Berghofer, T., Sintek, M., and Dengel, A. (2011). Semantic Logging: Towards Explanation-Aware DAS. In *International Conference on Document Analysis and Recognition (ICDAR'11)*, pages 1140–1144, Beijing. IEEE.

Furnas, G. W. and Zacks, J. (1994). Multitrees: Enriching and Reusing Hierarchical Structure. In *Conference on Human Factors in Computing Systems (CHI'94)*, pages 330–336, Bosten, Massachusetts. ACM.

Hartig, K. and Karbe, T. (2016). Recommendation-based Decision Support for Hazard Analysis and Risk Assessment. In *8th International Conference on Information, Process, and Knowledge Management (eKNOW '16)*, pages 108–111.

Jain, S., Grover, A., Thakur, P. S., and Choudhary, S. K. (2015). Trends, Problems And Solutions of Recommender System. In *International Conference on Computing, Communication and Automation (ICCCA'15)*, pages 955–958, Noida. IEEE.

Kaklauskas, A. (2015). Intelligent Decision Support Systems. In *Biometric and Intelligent Decision Making Support*, volume 81, pages 195–220. Springer.

Klahold, A. (2009). *Empfehlungssysteme: Recommender Systems - Grundlagen, Konzepte und Lösungen [Recommender Systems - Fundamentals, concepts and solutions]*. Vieweg + Teubner, Wiesbaden.

RDF (2014). *RDF 1.1 Concepts and Abstract Syntax*. World Wide Web Consortium - W3C. Accessed: 15.06.2016.

Sedgewick, R. and Wayne, K. (2011). *Algorithms - Fourth Edition*. Addison-Wesley, Boston, Massachusetts.

Sinha, R. and Swearingen, K. (2002). The role of transparency in recommender systems. In *Extended Abstracts on Human Factors in Computing Systems (CHI EA, CHI)*, pages 830–831, New York. ACM.

Tintarev, N. and Masthoff, J. (2011). Designing and evaluating explanations for recommender systems. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 479–510. Springer, US.

Uehara, R. and Uno, Y. (2004). Efficient Algorithms for the Longest Path Problem. In *Algorithms and Computation - 15th International Symposium (ISAAC '04)*, pages 871–883, Hong Kong. Springer.

Viswanathan, V. and Krishnamurthi, I. (2012). Finding relevant semantic association paths through user-specific intermediate entities. *Human-centric Computing and Information Sciences*, 2(9).

Viswanathan, V. and Krishnamurthi, I. (2015). Finding relevant semantic association paths using semantic ant colony optimization algorithm. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 19(1):251–260.

Wick, M. R. and Thompson, W. B. (1992). Reconstructive expert system explanation. *Artificial Intelligence*, 54(1-2):33–70.