# Caching Spreading Activation Search

Ján SUCHAL*

*Slovak University of Technology*
*Faculty of Informatics and Information Technologies*
*Ilkovičova 3, 842 16 Bratislava, Slovakia*
`suchal@fiit.stuba.sk`

**Abstract.** Spreading activation search is an approach to determine similarity of graph vertices based on energy distribution. Computation of such energy distribution is defined by a simple, but time-expensive recurrent procedure. This paper introduces a caching strategy, which can speed up computation of energy distribution of vertices that are closely connected to already calculated vertices.

## 1 Introduction

In information retrieval, associations, relationships, and co-occurrence between items are often represented by graphs. Clustering related items in such graphs is usually done by searching similar vertices using various graph topology metrics and node ranking algorithms. Results of such clustering can be used in intelligent search engines, collaborative filtering, or recommendation systems.

Spreading activation search is a simple recurrent procedure used to measure similarity between graph vertices. However in large and dense graphs this procedure starts to be slow and inefficient. This paper presents a caching strategy that aims to speed up spreading activation algorithm by storing partial results that can be used in subsequent calculations.

The rest of this article is organized as follows. In Section 2 contextual network graph theory is explained on an example of fulltext search. Section 3 introduces spreading activation search procedure. Section 4 presents caching strategy concept and Section 5 discusses fields of applicability of this caching approach.

---

* Supervisor: prof. Ing. Pavol Návrat, PhD., Faculty of Informatics and Information Technologies STU in Bratislava

*IIT.SRC 2007, Bratislava, April 18, 2007, pp. 1–8.*

## 2 Contextual Network Graphs

Contextual network graphs [5] are used to represent unstructured data. In domain of fulltext search contextual network graphs usually contain two types of vertices — *terms* and *documents*. When a term occurs in a document representing vertices are connected by an edge.[1]

Table 1 contains a collection of short documents. Table 2 contains terms, that appear at least in two documents of this collection. Figure 1 shows complete contextual network graph of this document collection.

| Vertex | Document body |
|---|---|
| 1 | Glacial ice often appears blue. |
| 2 | Glaciers are made up of fallen snow. |
| 3 | Firn is an intermediate state between snow and glacial ice. |
| 4 | Ice shelves occur when ice sheets extend over the sea. |
| 5 | Glaciers and ice sheets calve icebergs into the sea. |
| 6 | Firn is half as dense as sea water. |
| 7 | Icebergs are chunks of glacial ice under water. |

**Tab. 1.** Example document collection.

| Vertex | Term |
|---|---|
| $a$ | glacial ice |
| $b$ | ice |
| $c$ | glacier |
| $d$ | snow |
| $e$ | firn |
| $f$ | ice sheet |
| $g$ | sea |
| $h$ | water |
| $i$ | iceberg |
| $j$ | sheet |

**Tab. 2.** Terms appearing at least in two documents from Table 1.

[1] In general, graph edges can be directed, weighted, or even typed. Such graphs are used to represent complex domain ontologies.

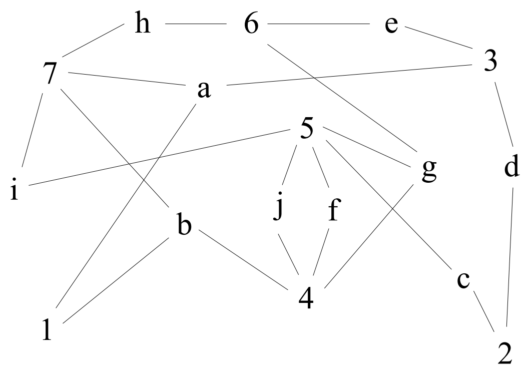


**Fig. 1.** Contextual network graph of example document collection [5].

## 3 Spreading Activation Search

Spreading activation search [5] is based on a simple recursive procedure that distributes energy on vertices through graph edges. Similarity search in graph is done by activating selected vertex with starting energy $E$ and distributing this energy through edges on neighboring vertices. This energy distribution is then recursively done on all neighboring vertices until convergence. In addition, a sum of all energy activations (overall activation) for each vertex is stored. Similarity of starting vertex to other graph vertices is then expressed by overall activation energy distribution $A$.

Spreading activation search can be formulated as follows.

> If vertex $n$ is activated by energy $E$, energy $E$ is added to overall activation $A_n = A_n + E$. Subsequently all vertices directly connected to vertex $n$ are activated by energy $\frac{E}{\rho(n)}$, where $\rho(n)$ is the degree of vertex $n$.
>
> For computability reasons vertex activation energy $E$ must also satisfy $E > T$, where $T$ is a given small energy threshold value.

## 4 Caching Normalized Activation Distributions

Table 3 shows an example distribution of calculated overall activation. Normalized activation distribution table is calculated by dividing vertex activation values by starting energy.

This normalized activation distribution is cached and can be easily used to skip the need for full recursive procedure. Every time a request for activation of a vertex is made, activation distribution cache is checked for this vertex. If cache already contains precomputed distribution table needed for this vertex, only a simple multiplication of ratios by activation energy is needed to compute overall vertex activations. All

| Vertex | Overall activation | Ratio to starting energy |
| --- | --- | --- |
| $a$ | 120 | 1.2 |
| $b$ | 60 | 0.6 |
| $c$ | 30 | 0.3 |

**Tab. 3.** Overall activation and normalized activation ratio for starting energy $E = 100$ applied on vertex $a$.

successive recursive calls normally needed to calculate such distribution for this vertex are omitted.[2]

## 5 Fields of Applicability

Spreading activation search applications include recommendation systems, fulltext search engines [1], trust propagation algorithms [6], etc.

Proposed normalized activation distribution cache is mostly useful in environments where spreading activation computation requests are more frequent than graph changes that result into cache resets, and thus making a fully recurrent procedure necessary. Furthermore, in most applications of spreading activation search, graph structure does not change so radically and therefore only minor computation errors are introduced when slightly older cache data is used for activation computation of new vertices.

## 6 Conclusions and Future Work

Proposed caching strategy based on storing normalized activation distributions speeds up computation of activation distributions for vertices connected with already calculated and cached vertices. Due to recursive nature of spreading activation algorithm overall activation distributions can be calculated by superimposing partial cached results.

Furthermore, this caching strategy can be combined with constrained version of spreading activation search which is based on enforcing additional constraints for energy propagation, such as maximum distance from starting node, node type enforcement, maximum degree of outgoing edges, and customized activation threshold functions [2–4].

Proposed normalized activation distribution cache is mostly useful in environments with infrequent and minor graph topology changes that usually result into cache resets or in environments where minor errors caused by using an older version of cached data are negligible.

[2] Total amount of skipped recursive calls depends on graph topology, graph density, and degrees of vertices.

Effectiveness evaluation of proposed caching strategy, such as cache hit ratio depending on graph density and graph topology is a subject to further research.

*Acknowledgement:* This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/3102/06 and the Slovak Research and Development Agency under the contract No. APVV-0391-06.

## References

[1] Aswath, D., Ahmed, S.T., D'cunha, J., Davulcu, H.: Boosting Item Keyword Search with Spreading Activation. In: *WI '05: Proceedings of the The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, Washington, DC, USA, IEEE Computer Society, 2005, pp. 704–707.

[2] Crestani, F.: Application of Spreading Activation Techniques in Information Retrieval. *Artif. Intell. Rev.*, 1997, vol. 11, no. 6, pp. 453–482.

[3] Crestani, F., Lee, P.L.: WebSCSA: Web Search by Constrained Spreading Activation. In: *ADL '99: Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries*, IEEE Computer Society, 1999, p. 163.

[4] Crestani, F., Lee, P.L.: Searching the Web by constrained spreading activation. *Inf. Process. Manage.*, 2000, vol. 36, no. 4, pp. 585–605.

[5] Maciej Ceglowski, A.C., Cuadrado, J.: Semantic Search of Unstructured Data using Contextual Network Graphs, 2003.

[6] Ziegler, C.N., Lausen, G.: Spreading Activation Models for Trust Propagation. In: *EEE '04: Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, Washington, DC, USA, IEEE Computer Society, 2004, pp. 83–97.