A New Form of Search Engine: Alive()

Walter B. Gress V

Drexel University

Friday, June 7 2013

This concept of operations details a “Divide and Defend” Firewall. Recently there has been a surge of malware attacks, those attacks typically being orchestrated through botnets. Modern enterprise and personal firewalls are not designed by default to handle massive number of attacks at once like botnets can incur. D&D is designed with that very thought in mind.

# introduction

Alive() is a prototype search engine that is more than just a search engine. It is intelligent, human-friendly, conversational, and efficient.

# feature list and ITERATIVE PROPERTIES OF THE D&D FIREWALL

Alive() uses semantic networks in addition to standard search engine algorithms so the meaning of the search is taken into account, not just raw parsing,Alive() understands the

Query, it doesn’t just function off of mechanics and mathematics.

Alive() uses natural languge processing so it can do searching using **rephrased** versions of your keywords generating more, accurate, hits.

Often times when trying to search, ninety percent of the search process is determining what sort of query you should use. This shouldn’t be.

Layers. Multiple nested searches to narrow out what you are lookiing for. Example: First query is “Storms”, a normal results page comes up. Clicking on one of the results asks you if you want to keep that link or if you want to look further and search a nested layer of that first search. Then you get results, you can choose a link to go to or you can go further nested. This continues ad infinitum.

Queries are in the form of questions. The search engine is like talking to a computer like it is human. NLP is very important.

Use Clusters back end described below.

Alive() uses real-time search and

Alive() uses suggestions tailored by crossing previous searches and present search

# CLUSTER ALGORITHM

1. Divides searches into subject matter. (Like Google has “Images”, “News”, “Maps”
2. To do this the data mining algorithm has to be able to classify pages and sift through subject matter.
3. The user is not “searching” but asking the search engine questions to which the search engine answers via the pages it returns.

Properties of Cluster Algoirthm:

Common tags between two pages cause them to move together or “cluster”

Distance is a function of similar content.

Clusters are formed when individual pages have common terms. Entire clusters can be attracted to other clusters.

Large topics consist of major topics, while smaller clusters are subtopics.

Cluster space is defined using antonyms, where for example “hot” is on the opposite end of “Cold”. Antonym tags cause pages and clusters to repel.

**Example Clusters**

1.

<home>

Hi My Name is Ted and I’m building a search engine.

</home>

2.

<body>

Where is Joel? He’s late for his dentist appointment.

</body>

3.

<body>

I’m running a search on Joel in the search engine

</body>

Since the majority of movement is towards the origin, sending newly mined data to a pre-defined location representing the outskirts of the cluster space seems to make the most sense. The value for the outskirts boundary should be calculated empirically.

“It” Cluster

“Change” Cluster

Health Cluster

This data space can be useful for a different, more language based search engine.

Data Filter

Send to Cluster Space

Mine Data

Cluster Space

Search Engine Front End