Creative Queries for Explorative Search

Rama Shravya
Faculty of Informatics
Otto Von Guericke Universitat
Magdeburg, Germany, 39106
Email:shravya.rama@gmail.com

Ikram Ul Haq
Faculty of Informatics
Otto Von Guericke Universitat
Magdeburg, Germany, 39106
Email:ikramatsofwer@gmail.com

***Abstract*—This article discusses our development of a new interactive search engine interface which generates more explorative and creative results, exceptional from typical accurate ad hoc search results. We will discuss in detail about the implementation of efficient algorithms on input queries to generate creative queries. This paper also discusses about different proposed interfaces and their evaluation techniques which resulted in the current interactive interface.**

I. Motivation

By using traditional Search Engines like Google, Yahoo and Bing, will always get you the most accurate and point to point results. These Search engines try to know what users want by utilizing the factors like cookies and geographical location. For example when you search for a keyword Manam which is a tollywood movie title from a system which is either acquainted with this language or location, search engine will give you the list of results of which all of them on the first page would be related to the movie itself. But when the same keyword when used on a different system which is no way related to this domain, search results would have two different fields in which one is related to the movie and the other to a Thai restaurant in Munche.

Traditional search engine suits really best for all other users who know what they want and probably who doesn't want to know the twisted solutions of the same problem. But for the users who would like to view results in a broader prospect, traditional search engine will narrow down its scope. Hence we need a search engine which is explorative in getting results and which can give a user different aspects of problem solving techniques rather than getting accustomed to the accurate search methodology.

II. Introduction

Exploratory Search is for those users who are unfamiliar with the domain of their goal and even unsure about the ways to achieve their goals. Explorative Search's goal is to get a broader results view of the search query, which is achieved by manipulating the original search term using various algorithms and then regenerating new query terms. The results which are based on these new query terms are unexpected but relevant to the content of original search term. Hence a user is exposed to wider concept of the search term rather than getting narrowed to the accurate results.

Febraury 18, 2016

A. Related Work

In regard with this concept, there are many papers published. The following are two of them which are ought to be implemented in Exploratory search.

1) Creative Search Using Pataphysics by Fania Raczinski, Hongji Yang, Andrew Hugill
2) The syzygy surfer: (Ab)using the semantic web to inspire creativity by James Hendler and Andrew Hugill

These papers suggest that, in contrast to the traditional search engine's semantic web technologies, in exploratory search engine, relationships between items are exploited in a total random fashion so as to generate new interesting and unusual relations. Syzygy, Clinamen and Anomaly are the different techniques used to collect different random but connected items on web. The new collected items are then fed to traditional search engine to generate explorative search results.

III. ALGORITHMS & APPROACH

Syzygy: This methodology uses WordNet dictionary using NLTK python library to find suitable results. WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Syzygy of a term is formed by intersecting the union of hyponyms, holonyms and hypernyms of the term with the original vocabulary of same term.

For a search term t
syzygy( ) = ∈ union( ) ∈
union( ) = hypo( ) łhyper( ) łholo( )
hypo( ) = hyponyms( )
hyper( ) = hypernyms( )
holo( ) = holonyms( )
syno( ) = synonyms( )
for syno( )

For e.g. Let t = live
syno(live) = populate, inhabit, be, domicile
hypo(live) = NULL
hyper(live) = be
hypo(live) = domicile, reside, camp, tent, nest

-¿Syzygy(live) = be, domicile

Clinamen: The clinamen function uses the Damerau-Levenshtein algorithm which measures the distance between two strings.
For a search term t
clinamen (t) ={   0 ¡ dameraulevenshtein (t,   ) 2 }, for
For e.g. Clinamen of LIVE= LOVE, LIES, SIZE, RIVER

Anomaly: Anomaly function simply makes use of WordNets antonyms. We first get all synonyms for query term and then find antonyms for synonyms. Words common to direct antonyms and synonym's antonyms are chosen to be anomaly output.

For a search term t
antinomy(   ) =   anto(   ) and
anto(   ) =   antonyms(   )
syno(   ) =   synonyms(   )
for   syno(   )

For e.g. antonym(live) = dead, recorded
synonym(live) = alive, animate, breathing
antonym(alive) = dead
antonym(animate) = dead, inactive, inhibit..
antonym(breathing) = breathless, dead
anomaly(live) = dead

1) Subsubsection Heading Here: Subsubsection text here.

IV. Implementation

V. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.