VExSearch: Improving Visualizations Results for Web-based Information Exploration and Refinement

Mohammed Najah Mahdi, Abdul Rahim Ahmad, Roslan Ismail, Adamu Muhammad Buhari

Abstract— **As the web grows, finding information from large data repositories has become increasingly difficult not only due to inadequate number of results that are relevant, but also due to poor sorting of relevant results from those irrelevant. The present search engines (SEs) use the query and response lookup process that does not provide precise results. Thus, researchers have gone beyond the paradigm to explore a new class of method to seek information, which is called exploratory research that is open-ended and its faceted search can improve the overall search process. Besides, many studies have begun tapping into enhancement of web search results relevancy. The web reflects vast heterogeneity, varying structure, and massive in volumes. Therefore, it is rather difficult to seek accurate outcomes as desired. As such, visualisation and interactive graphics have been proposed as methods to manage massive amounts of results and to project essential features for the web pages. Additionally, search engine controls reconstruction and reformulation of queries. As such, a search engine is presented in this study by developing it on the cloud computing platform environment. The search engine is based on the idea of improving visual exploratory search (VExSearch) while exploring information in the web. This particular notion reflects the process of seeking and combing through the vast information by using the coordinated visualisation method, apart from minimising the effort spent in seeking information per query. The VExSearch was evaluated for its capability and performance and later compared with IMDb SE and CloudMining SE. The comparative results showed that the VExSearch was 66% more accurate than the other SEs. VExSearch also seemed to provide the most relevant results among all the three SEs, aside from attaining an average improvement of 20% in terms of recall.**

Index Terms— **Exploratory Search, Faceted Search, Information Visualization, Information Retrieval, Visual Information Seeking.**

I. Introduction

As a popular tool on the WWW, search engines (SEs) are designed to help users find useful information that could contain text, pictures or videos[1]. The main challenge for SEs include the ever-increasing information made available on the growing web, The SE refers to an instrument that aids one to seek relevant information from the web. A query is entered into a web SE, and then, the web SE will display relevant results based on the entered query[2]. Query refers to question(s) that seek answers. As the web grows, web users seem to face increasing difficulty in seeking and identifying relevant information[3], primarily due to two reasons: (a) inadequate results that are relevant[4], as well as (b) poor segregation between relevant and irrelevant results[5]. The way a web SE sorts its results is embedded in user interface system. The user interface manages and displays query of item information and the required information is given to the user. It is also very easy to make changes to the user interface for implementation that could be beneficial for users. Changes to the user interface of a web SE are the simplest to implement and are potentially useful to users.

To date, people live in a world with effectively infinite information, where users carry devices (e.g. mobile phones) that allow them to access a large body of human knowledge. As such, finding facts is more significant than actually knowing the facts. The gate-keeper to this vast trove of data is the SE[3, 6].

Current-generation SEs are considerably useful, but they tend to work effectively only for easy queries such as finding homepages or popular/known topics[7]. In general, when users require well-defined information and possess good knowledge of target web pages, they can formulate an effective query, and the SE can always return top results. In such cases, a query with a few keywords is often sufficient. Unfortunately, queries with short keywords are not always effective when users do not have any particular target pages in mind or when they are not familiar with the topic to be searched, as is often the case in exploratory search (ES) and informational search[8]. Technically, queries are ineffective because of multiple reasons[9, 10].

The proposed SE prototype is a new ES engine that is designed to overcome the limitations of existing methods. Features of information visualisation are used to improve the comprehension of large amounts of data or information through graphical representations. Such system is suitable for the cloud computing platform environment. The prototype is based on the idea of a visual exploratory search (VExSearch) engine for information exploration on the web. This concept aims to support an active means of finding and exploring information via coordinated visualisations and to reduce the average amount of search efforts needed per query. VExSearch is evaluated with regard to its capability and

Manuscript received June 29, 2018; revised April 16, 2019.

Mohammad Najah Mahdi is a Post-Doc of the Institute of Informatics and Computing in Energy, College of Computing and Informatics (CCI), Universiti Tenaga Nasional, E-mail: mehdi.najah@uniten.edu.my

Ahmad A.R is with the Systems and Network Department, College of Computing and Informatics (CCI), Universiti Tenaga Nasional. E-mail: abdrahim@uniten.edu.my

Ismail R is with the Software Engineering Department, College of College of Computing and Informatics, Institute of Informatics and Computing in Energy, Universiti Tenaga Nasional. roslan@uniten.edu.my

Adamu Buhari is a PhD candidate at Faculty of Engineering, Multimedia University. E-mail: adam_m.buhari@yahoo.com.

performance and is then compared with IMDb SE and CloudMining SE. The VExSearch approach attempts to naturally extend the current consensus around traditional search. The system is built with the idea that no loss should be incurred in moving from one paradigm to the next. That is, whatever could be performed in the old setting must still hold in the new setting. The system eventually fulfils the list of what the researcher believes would be the main natural component of ES.

II. Research Background

The literature pertaining to information exploration and navigation runs across multiple research groups. This section elaborates several essential concepts within the scope of VExSEarch.

A. Finding and Exploring Information

Many studies have shown that ES activities are common information-seeking behaviour exhibited by users. Based on a study that determined the trend of personally-motivated search in the web, emails, and files, Teevan, et al. [11] discovered that exploratory type of search was the highest at 61%. In another empirical study, White, et al. [12] found that 23% of searchers were almost entirely exploratory, while 17% did not exert such trait. Wilson [13] analysed a particular interface that displayed both searching and browsing activities, in which approximately 50% of the searches showed alternatives to keyword search. The present commercial SEs appear to use a process known as the query and response. The user issues a query, and receives, as a response, a set of potentially relevant documents. The process has been formalised by Bates [14] in the lookup-based model.

Another ES approach that is novel is the faceted search (FS), which helps users to explore information from the massive databank. In precise, FS permits its users to seek information through several discrete functions known as 'facets'[15, 16]. Generally, information overload can be avoided by minimising the amount of facets or facet values, and/or by preferring those with higher coverage in the set returned. The vocabulary can also emerge as an issue if the design of the facets is at odds with the user's expected classification[17].

B. Advantages of Faceted Search

Faceted search provides a more dynamic means to browse and search for resources than traditional "advanced search form", where all the available search fields are provided at once. In the traditional manner, users have to set up the search criteria at the beginning of the search. However, users may not be completely clear about the keywords in all the dimensions when they initiate the search, and thus, the traditional search is unsuitable for typical searches. Moreover, enterprises attempt to provide additional valuable information to the existing structured template[18]. Consequently, structured properties may increase to an extremely large number, which leads to a challenging search task and a loss of search focus. In addition, users may want to select a combination of values that does not even exist in the document data set, and thus, an ideal solution is a navigational system provided by FS that can guide users to their areas of interest.

C. Information Visualization for Search and Similarity Search

Faceted search allows users to explore or navigate within the document collection. However, most mainstream search systems only feature a fixed mode of interaction. For example, search results are most often depicted as a list of text with minimal interactions, such as sorting or paging. To obtain new understanding of data, allowing for multiple interaction modes is necessary. According to White and Roth [19], ES system should increase user responsibility and control. This feature should include letting the user select how the data is visualized depending on the task of interest. Therefore, this paper will go beyond traditional FS and look into the way information visualization could be employed to make the user experience more of the exploratory aspect of the search. First, the query terms will be revisited, and then, cover several examples of visualizations that could be applied to search results or to facets.

In a typical search scenario, the user inputs a set of query terms and obtains a set of matching documents. Usually the query terms remain in the search box. To reformulate the query, the user has to click in the input search box and manually add or remove query terms. A different approach consists of allowing users to more directly interact with query terms. These terms are usually depicted in the form of tags with actions, such as toggling, removing, or clearing. The user is able to easily manipulate the query quicker, thereby obtaining narrower or broader search results[4].

Another visual play on the query consists of providing relevant suggestions. Query suggestions are the product of extensive research in information retrieval (IR) on query expansion[20, 21]. The idea behind query suggestion is to offer user additional keywords for the user to consider, which could guide the search towards relevant documents. In its most simple usage, the suggested query terms simply act as shortcuts to previously typed queries[22]. However, suggestions may help the user discover a set of query terms that lead to new documents of interest. Query suggestion has most commonly been implemented within large commercial SE by using substantial search logs[23]. For example, if a user were to type the query "the hobbit", the input by other users may have led to suggest "movie". The user may have been unaware of the movie, "The Hobbit". If the user had been searching for "Tolkien" instead, then the SE might more simply, have suggested the query term "book". The suggestions are most often depicted in the form of a list. However, other SEs have tried more appealing visualizations with quite a success because a SE can respond to a user query with an overwhelming number of related web pages.

Quintura is a visual find engine that uses tag clouds and a rich interface to dynamically display and refine results. The interface includes a query box at the top of the page, a tag cloud on the left side of the screen, and a rich results display on the right hand side of the page. (See Fig 1).

A drawback to query suggestion is that the method may induce the users into the most conventional pathways and consequently reduce exploration. This problem is usually referred to as "query drifting"[24]. In this respect, what most people would be presented with is a much narrower set of the entire web. The results that Google retrieves may just as well be coming from its cache[25]. One way to address this issue

is to provide greater feedback between the query and the retrieved results. This feature would then lead to another matter for discussion, which is the tight coupling between query terms and search results in the form of dynamic queries[26].

Fig. 1. Quintura Represents Suggested Query Terms

D. *Visualization on the Facets*

FS provides a seamless integration between browsing and searching. The user searches for keywords, obtains results, and potentially continues browsing the corpus through the different facet values. Facets also provide interesting summaries of search results with respect to facet classification. More precisely, the facets can reveal patterns of distribution and occurrence at an aggregate level. However, for those patterns to emerge, the data must be represented appropriately. This section covers how different visualizations can be applied to the facets themselves.

Much of the success of FS is due to the use of query previews. Query previews [27, 28] give the user a hint of what to expect before he selects a link or issues a query. In a standard FS system, the query preview is a simple numerical count. Some systems have attempted to represent this count more graphically. The Relation Browser (RB) is an early example. In RB, a bar indicates the relative frequency of the facet terms.

The darker portion of the bar shows the count if the facet term is selected within the current search space, while the lighter and longer portion of the bar shows the overall count of the facet term within the entire collection. RB also has the ability to switch views between search results and facets [29]. Facets are presented in a cloud view similar to a tag cloud. RB also features dynamic queries with an excellent response to feedback. However, the system is client-based, which limits its scalability. As we will see, VExSearch shares many features with RB, but can scale to thousands of users and millions of documents.

Another system worth mentioning is FacetLens[30]. The facets on FacetLens use most of the real estate of the interface. The facet values are ordered by frequency and depicted as large circles. These circles depict the actual search results of interest. According to the authors, the interface helps users identify and compare trends. Furthermore, it offers pivot operations, which allow the user to navigate the dataset using relationships between items.

Visualizing metrics, such as frequency, within facets could be interesting in the discovery experience. The correct visualization can shift the focus from searching to more exploratory tasks, such as data analysis. However, facets originate in metadata of many different types. For example, dates can be represented textually or more graphically, as a timeline. Locations can be better served by points or regions on a map, rather than by a list of coordinates. Therefore, a broader set of visualizations than those limited to depicting frequencies is possible. This section now will go to review some of the visualizations possible with respect to the "type" of facet at play.

E. *Quantifying Data with Bubbles*

Another visualization example comes from the ManyEyes project [32]. In the project, users upload data to ManyEyes, choose visualization, and then share it with others for discussion. Fig 2 presents a depiction of the human world population by Language Speakers. This visualization numerous other possible visualizations can be employed. Although not necessarily immediately applicable to FS, the classic books from Tufte [33], Hays [34] and provide numerous interesting methods of visualizing information. However, the real challenge consists of building a system which can integrate many of these visualizations.

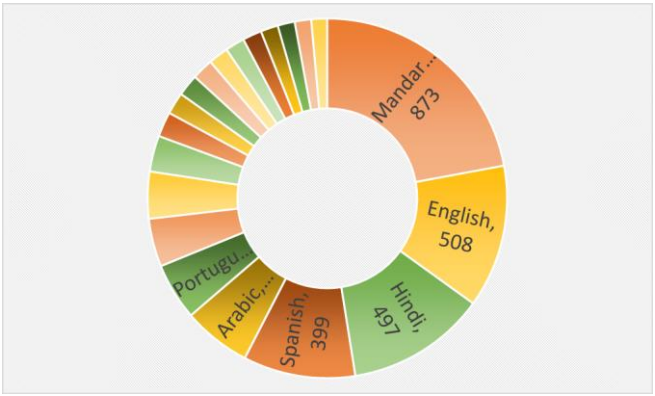

Fig. 2. Visualization Demographic of Language Speakers

III. System Implementation

The implementation of Web-based system is aimed to support visual exploration of large information collections utilizing VExSearch. The prototype performs item-based search and it can be scaled to handle very large datasets of tens of millions or even billions of items.

A. *System Architecture*

The architecture of the system is designed to allow pluggable search. It is composed of software modules that can be used independently such as widgets that performs a particular function[35]. The desired user interaction will first be described with an example and next, the pluggable search architecture of VExSearch will be outlined. The different modules that comprise VExSearch will also be described, and the process of building the instances will be presented. The architecture of the visual information exploration system is divided into two parts: client-side and server-side (see Fig 3).

From a software perspective, the functionalities of VExSearch modules must be decoupled from each other as much as possible. The VExSearch opted to make a web

application, which requires different modules for its various exploratory search tasks.

Web-based architecture of the implemented system, each module is software that can be used completely independently. For example, Sphinx module is used for FS, whereas similarity search module is used as an item based SE which implements Bayesian Sets. It performs item-based search, and it can be distributed to scale into very large datasets of tens of millions or even billions of items.

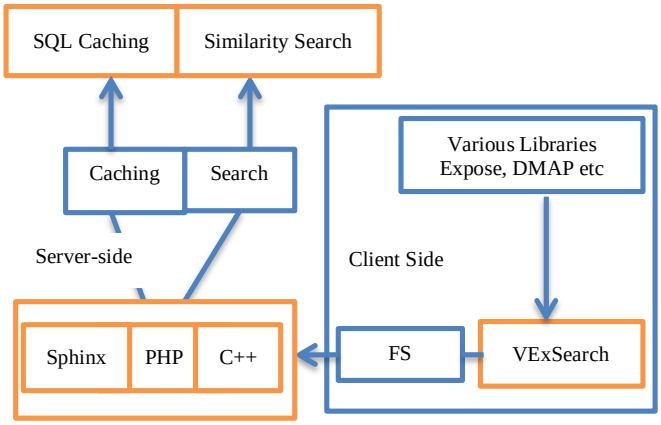

Fig. 3. Web-based architecture of the implemented system

Fig 4 shows how each module is built on top of existing technologies and how these technologies interact with one another. The figure depicts the main application stack of the VExSearch architecture. VExSearch is built on top of the Python web architecture[36]. VExSearch uses various libraries such as Expose, Dimap and WebError to perform many of its functions.

The actual facet computation, is the Sphinx retrieval engine[37]. Sphinx is an open-source and full-text search server written in C++, which powers many websites, such as Craigslist[38], Living Social[39], MetaCafe[40], and Groupon[39]. Sphinx was selected because of its speed and scalability, which consequently makes VExSearch as scalable as Sphinx. By contrast, systems such as Flamenco[41, 42] are entirely database driven and thus difficult to scale beyond hundreds of thousands of documents[41, 43]. Despite its speed and scalability, Sphinx lacks features found in other SE such as Lucene[44] or Solr [45]. For example, Sphinx does not have a storage engine and does not support aggressive caching. Facet computation requires various back-end setups to function properly and is not to be necessarily user-friendly.

B. *Visualisation on VExSearch*

It has been reckoned that the web is comprised of various structures that can be found in massive volumes, along with its heterogeneity nature. Therefore, it is indeed difficult to retrieve search results that best match one's query. As such, the approaches of visualisation and interactive graphics provide displays in order to better manage the massive amount of results, apart from offering some features of the web pages via VExSearch. Besides, as the reconstruction and reformulation of queries are normally handled by SEs, the results may be poorly irrelevant or unnecessarily redundant. Therefore, in order to improve the relevancy of search results, users are also incorporated in reformulating queries through visualisation in VExSearch.

The results that are in line with user query would appear in the hit list offered by the SE. However, the format of textual list and the vast amount of files make finding the desired results a difficult task. VExSearch, which is designed to enhance web search, adheres to the notions depicted below:
a) Visualisation must present the display for VExSearch in a format that can ease comprehension due to its massive hit list. b) The visualisation must embed several web document functions, such as size of page(s), similarity of the view page with the rest of the document, and the last update information, in order to offer the users detailed and cutting-edge VExSearch results. c) The visualisation method must weigh in the tasks and user types. This is because; most users seem to be the casual type without adequate search skills. Hence, VExSearch should consider search learning and training when users perform the search process.

C. *User Interaction*

The instances built with VExSearch all share the same kind of user interaction regardless of the type of data being explored. The features of search, facets, visualization, and query by example are all included. To build the interface, a conventional FS interface is adopted and extended with exploratory capabilities. Rather than the interface, the overall sought user interaction will be focused on. The details of the interface and its customization will be discussed in the next sections.

D. *Bayesian Sets in VExSearch*

VExSearch uses Bayesian Sets[46] search algorithm which is a model-based concept of a cluster that ranks each item by using a score that represents the marginal probability of belonging to a cluster containing the query items. The approach allows for multiple item-based queries which reduce the work involved in setting up a similarity search based solution to feature engineering. The search algorithm is referred to as item-based as opposed to content-based[47].

In full text search, the query is made of keywords which are then matched against a back-of-the-book index. In item-based search, the query consists of whole items which are themselves composed of feature values. Here, the back-of-the-book index is replaced by a list of feature values with their corresponding items. The algorithm finds the set of items which best fits within the cluster defined by the query items. Bayesian Sets have been chosen for matching similarity to VExSearch.

As we have seen, the unique characteristics of Bayesian Sets fit quite well within architecture such as VExSearch. However, the design of an interface which supports FS as well as item based search still remains a challenge. There are many open questions about the design of the interface to support such functionality, but the overall sought behavior can be outlined.

The overall design goal was not to depart too much from conventional FS behaviors. As such, the interface should provide the ability to mix items with conventional faceted metadata selections. The results should be a set of similar items restricted to the facet selection(s) and/or full text search. The facet grouping function should be adapted to make use of the similarity search scores. For example, the terms in a tag cloud could be weighted by the similarity scores

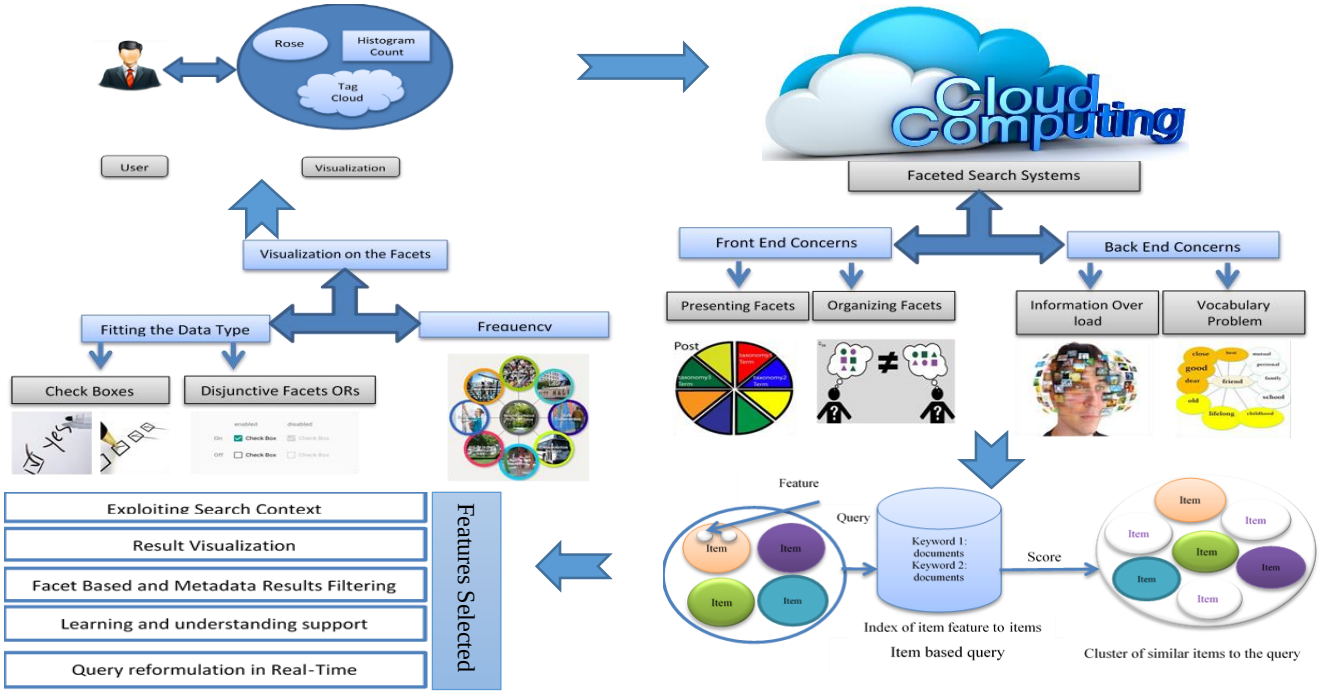

Fig. 4. VExSearch Web-based Architecture

in addition to its frequency. The interface should also provide feedback as to why the documents have matched. This is important in order to help users form a mental model about the underlying matching algorithm[48].

E. Building Search Engine

A VExSearch instance is a web application that overrides the default behavior provided by the architecture which uses the normal default search engine. To build an instance, first a Sphinx client (to extract the data of interest) is written. The Sphinx client is registered to the instance. For item-based search, a similarity search client (SimClient) is registered in a similar manner. The application is run like any other Python web application.

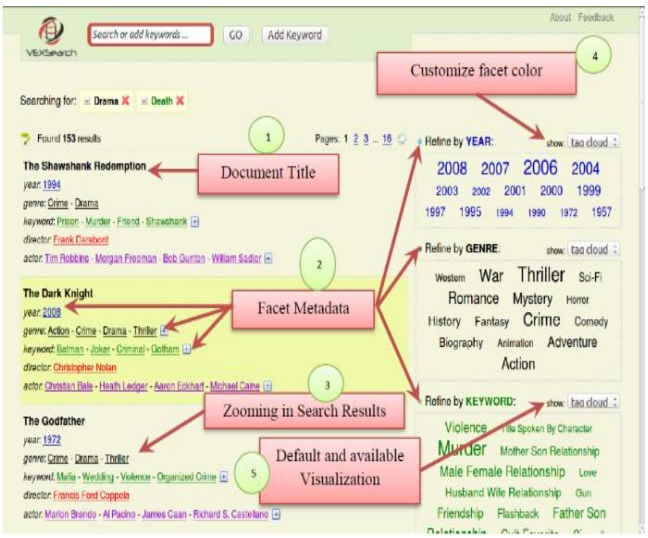

Fig. 5. VExSearch Instance before Zooming in

Fig 5 illustrates the Search Engine Result Page interface. In the diagram, the labelled part (indicated by the numbered circles 1-5) of the interface are as follows:
1. Document title: This is the generic interface that indicates the title of each item. 2. Facet Metadata: This part shows the available facet metadata. 3. Zooming in Search Results: This is the interface that can be zoomed-in by clicking on the keywords in the faceted metadata. Each keyword will bring up a new screen showing a more detail document surrogate which more is appealing, with user ratings and a description. (See Fig 6). 4. Customize facet color: This part allows the color facet to be modified; the facet color will change after every zooming in. 5. Default and available Visualization: The available facet visualizations, by default, will change for each zooming in.

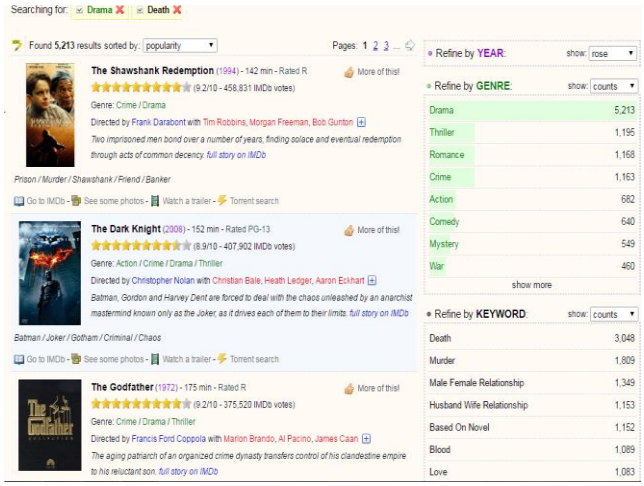

Fig. 6. VExSearch Instance after Zooming in

The overview of the VExSearch architecture ends in this subsection. To recap and emphasize, VExSearch is made up of independently usable modules for each exploratory search task. The VExSearch front-end as well as the back-end for faceted search, visualization, and item-based search will be discussed in detail in the following sections.

F. Front-End

There are many elements that must be considered including good informative document surrogates, query terms and

several sorting scenarios. VExSearch have the choices of using any layout that exists in many search engines. However, the conventional layout was selected because of the generic nature of the interface. The front-end concerns raised in section A, included the organization of facets and their values, the behavior of the search box, or the means to perform multiple selections within facets. Many of these concerns are best described and addressed by presenting a usage example.

Figure 7 shows the front page or an entry page of a VExSearch instance. A portal-like page appears which shows a search box, a short description of the instance, and several facet panels. In this Fig 9, although the instance has been slightly zoomed in for the IMDb dataset, the generic page is highly similar. Only a number of facet views and colors have been changed. Users can either type a query in the search box, or select facets value.

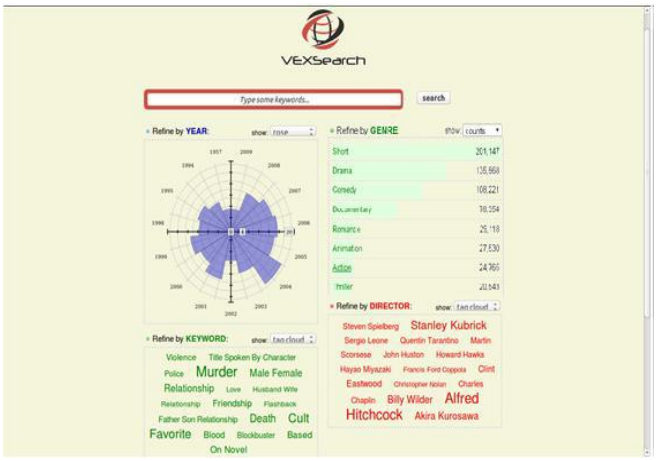

Fig. 7. VExSearch FrontPage with Many Different Kinds of Facets

Fig 8 shows the word "animation" from the genre facet. The search results are expectedly shown on the left hand side. Again note that for this example, the document surrogate has been zoomed in, which is more appealing. The labelled part (indicated by the numbered circles 1-6) of the interface are as follows:

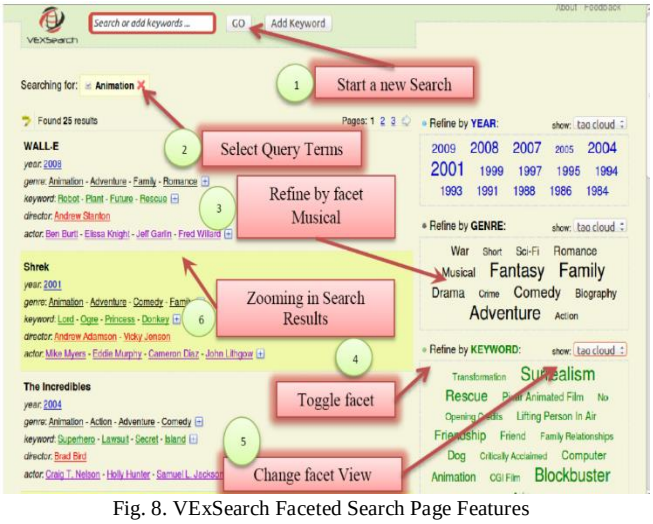

Fig. 8. VExSearch Faceted Search Page Features

1. Start a new search: Next to the log in, the search box is extended with an "Add Keyword" button or started a new search. 2. Select Query Terms: This lets the user add keywords to the current query; the user can do their searching (refining) within the current results, by Select Query Terms. 3. Refine by facet "Musical": Using the facets on the right hand side, the user can further refine his search, the moment the interface only supports conjunctive (AND selection.) facet metadata selections. 4. Toggle facet: Facet can also be toggled off or back on by clicking on the title, the user can hide the inter Refine by toggling off. 5. Change facet View: The facets have different views such as a tag cloud or a histogram count. The user can specify this view using a roll down menu. 6. Zooming in Search Results: Zooming in search results with the current search, by choosing different facet keyword to reduce the results.

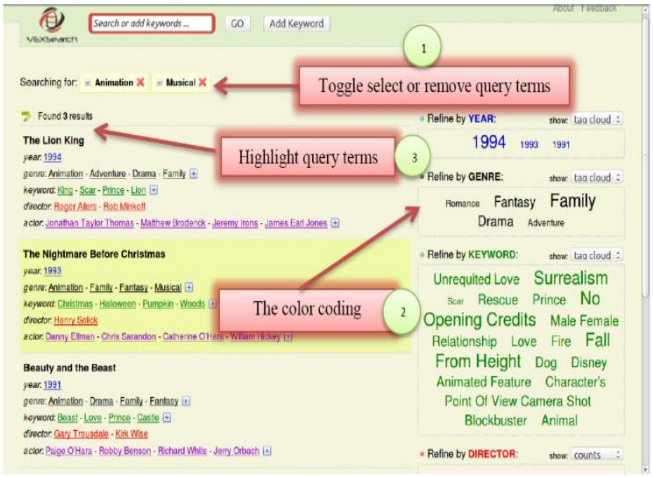

Fig 9 VExSearch Faceted Search Page: other Features

Fig 9 presents additional features of the interface. Also continue with results page above to presents new features, the detail description is given in points below:

1. Toggle select or remove query terms: The user has conjunctively refined the results by the facet value "Musical". To test different combinations of queries, the query terms can be toggled on or off or removed and started again. 2. The color coding: If the facet metadata are not apparently evident from the previous example, each facet metadata is assigned to a specific color. A common feature in the interface includes highlighted query terms within the document surrogates on demand, the user can check the highlighted within or without the results. 3. Highlight query terms: The user may be seemed initially confused in the test; they quickly determined the facets that a metadata belongs to upon understanding the color coding. This feature addresses a common user behavior pattern, which consists of re-starting a new query with the same keywords including new ones to redirect the search to a different path.

IV. RESULTS

In this section, search experiments were conducted to study the proposed VExSearch search engine (SE). We presented to the VExSearch some examples of queries that a user might have. The goal was to evaluate how the VExSearch performs. Each output was analysed with respect to accuracy and detail. A diagrammatic view of the test results and evaluation of the proposed SE are depicted in Fig.10 in two parts: evaluation of the searching capabilities of the SE and performance of the SE. Both parts will be discussed in detail in a subsequent section.

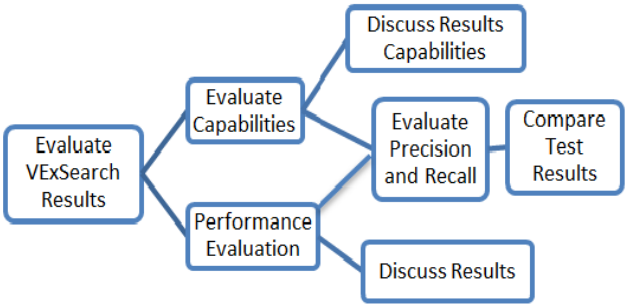

Fig 10 Overviews of Test

A. *Searching Capabilities Evaluating*

The first things to look into before diving into the details of the experiments were the setting and preparation of it all. A lot of time the preparation includes the main characteristic outlines of the dataset and user queries. Results related to the dataset and user queries will then be discussed and evaluated.

The present dataset was comprised of the IMDb database. An argument related to the definition of query results as tuples, reflected the presence of a browsing data framework. The structure of the browsing data framework for this dataset was presumed to reflect that of the IMDb webpage, which includes cast details such as the characters played by actors. The page of an actor showcases all movies the actor has been in with character descriptions in each screenplay. The character record however, displays the list of actors who played the roles. This observation provided insight on the queries found by Luo, et al. [49] in line with the assessments of prior studies.

The processing engine SPARK applied a database that differed in terms of film genres. Queries associated with genre are substituted with that related to casting members, directors or actors/actresses. A fresh query can substitute the original query within the SPARK assessment process as relevant results seem to be unidentifiable, perhaps due to unclear queries or limited tuples with precise keywords. The results, which should express the user queries, are determined based on how well a query is addressed with information provision. To keep up with this, eight queries were modified out of the original 22 queries, as shown in Table I.

TABLE I
EVALUATIONS OF IMDB QUERIES[49]

No	Modified Queries	No.	Modified Queries
Q1	Robbins Neil Giuntoli	Q7	Mathilda Leon
Q2	Harry Potter 2002	Q8	Mathilda Leon 1994
Q3	Harrison Ford President	Q9	Wachowski Trinity
Q4	Space Odyssey director	Q10	Hamill actor
Q5	Bourne	Q11	Wachowski Trinity Oracle
Q6	2004 Hanks	Q12	Woody Allen

Kumar and Pavithra [50] along with Uddin, et al. [51] conducted in-depth analyses to compare the searching capabilities of SEs (Google, Yahoo and Refseek) with Metasearch engines (Metacrawler and Dogpile) in order to determine two significant data: relative recall and precision value. Thus 15 queries associated with library and information science topics were been chosen for this study and submitted to the SEs and Metasearch.

The experiment took into account the study by Kumar [50] and Uddin, et al. [51] which compared the searching capabilities of three SEs (VExSearch and two benchmarks). The comparisons were based on the relative recall and precision value. Twelve (12) queries, which represented a broad range of IMDb queries (nine adapted from SPARK along with three new ones) were used and submitted to the SEs. The first 100 results for each query were evaluated. It was recommended that each query be executed in the two SEs at nearly the same time in order to avoid temporal variations and to retrieve relevant data from each.

TABLE II
SE EVALUATION CRITERIA

No	Category	Criteria
1	more relevant	"If the content of the web page closely matched the subject matter of the search query, then it was categorized as "more relevant" and it was given a score of 2".
2	less relevant	"If the content of the web page is not closely related to the subject matter, but consists of some relevant aspects to the subject matter of the search query, then it was categorized as "less relevant" and it was given a score of 1"
3	irrelevant	"If the content of the web page is not related to the subject matter of the search query, then it was categorized as "irrelevant" and it was given a score of 0"
4	links	"If the content of the web page consisted of a whole series of links, rather than the information required, then it was categorized as "links" and it was given a score of 0.5, if inspection of one or two of the links proved to be useful".
5	sites can't be accessed	"If the site can't be accessed for a particular URL, then the page was checked later. If this message repeatedly occurred, then the page was categorized as "site can't be accessed" and it was given a score of 0".

When a search operation is carried out in response to a search query, many a time the user is unable to retrieve relevant information. The quality of searching for the right information accurately is said to be the precision value of the SE[52]. In the present study, the search results retrieved by all the SEs were categorised into five groups as depicted in Table II on the basis of the criteria set up by Kumar and Prakash [53]. The scores for each site retrieved by the SEs were manually given based on category 3.

These criteria had enabled the calculation of the relative recall and precision value for each of the queries, using the formula below [53]:

$$\text{Precision} = \frac{\text{Sum of the scores of sites retrieved by SE}}{\text{Total number of sites retrieved}} \quad (1)$$

$$\text{Relative Recall} = \frac{\text{Sum of sites retrieved by the two SE}}{\text{Total number of sites retrieved by SE}} \quad (2)$$

1) *IMDb Precision*

When searching for the 12 queries, not all of them returned results. A total of 410 sites had returned results and from that number alone, 134 sites were selected for a comparative study with Kumar [50] and Uddin, et al. [51]. The majority of the query results were from Q12 as it returned the most relevant results. Table III illustrates the relevant statistics for the 12 queries and the selected sites. From the table, it was clear that 36% of the sites were more relevant (category 1), while 8.5% were less relevant (category 2). The total percentage of relevant sites was 44.5% (almost half), while the irrelevant sites totaled 29% and about 0.32 was found to be the mean

precision of the whole search. As evident in the table, the query results with respect to the search precision for different queries (Q1 to Q12) can both be analysed and summarised. It is important to note that the main objective of the search process was to help researchers gain better understanding about reach via the exact query matching with relevant or irrelevant sites. More data layers were found in the analyses below:

TABLE III
PRECISION OF IMDB SEARCH

Queries	Sites no.	Sites	Categories					P
			1	2	3	4	5	
Q1	0	0	0	0	0	0	0	0
Q2	2	2	1	0	1	0	0	1
Q3	0	0	0	0	0	0	0	0
Q4	0	0	0	0	0	0	0	0
Q5	10	10	3	0	7	0	0	0.6
Q6	9	9	3	0	6	0	0	0.67
Q7	10	10	1	0	9	0	0	0.2
Q8	0	0	0	0	0	0	0	0
Q9	0	0	0	0	0	0	0	0
Q10	10	10	0	0	3	7	0	0.35
Q11	0	0	0	0	0	0	0	0
Q12	369	100	43	12	15	23	7	1.095
T	410	141	51	12	41	30	7	0.32
% T			36	8.5	29	21.2	4.96	

a) For queries Q1, Q3, Q4, Q8, Q9 and Q11, their search results did not return because the keywords used had no relation to the IMDb database for selected queries (exact keywords). The precision was calculated to be zero.
b) Q7 had returned 10 sites, and yet however only one of those was relevant (category 1) while the other nine were irrelevant sites (category 3). Due to this, the precision was calculated to be 0.2.
c) Out of all the selected returned results for Q10, three were irrelevant sites (category 3), while seven were links (category 4). Similar to Q9, these returned sites were web pages consisted of a series of links based on the scores for each retrieved site. There were no matches for the query selected which resulted in the precision to be calculated as 0.
d) For Q12, 100 sites were selected from 369 returned sites. The combined results for more relevant (category 1) and less relevant (category 2) sites were 55. Furthermore, the combined results for irrelevant (category 3) and links (category 4) sites were 30. Query Q12 had garnered more returned results with the highest precision value of 1.095, as compared to the other queries.

2) CloudMining Precision

Search experiments were performed on the 12 queries but not all of them returned results. The total sites that returned results were 268, out of which 127 were selected to emphasise comparison with the work of Kumar [54] and Uddin, et al. [51]. The majority of the selected query results were from Q12 as it returned the most relevant results. Table IV illustrates the relevant statistics for the 12 queries and the selected sites. As evident in the table, 43.6% of the sites were more relevant (category 1), while 13.53% were less relevant (category 2). The total percentage of relevant sites constituted almost half of the selection (57.13%), while 4.51% were irrelevant sites. The mean precision of the search revealed to be 0.76.

From the Table IV above, the query results with respect to the search precision for the different queries (Q1 to Q12) can both be analysed and summarised. Due to additional features, coupled with more accurate matching and similarity results, the idea that CloudMining gives better results than IMDb seemed to be true. This had been further proven by the following analyses:

a) For queries Q1, Q5, Q8 and Q9, the results with respect to the SERP did not show high precision values since the queries did not retrieve relevant sites due to lack of facet metadata. The precisions for those queries were all below 0.7.
b) No relevant sites (categories 1 and 2) were garnered by both Q3 and Q10. Two irrelevant sites (category 3) returned which indicated that the searching process in CloudMining lacked facet metadata keywords, resulting in no matches and very low precision value of 0.2.
c) For Q2, Q4, Q7 and Q11, one relevant site returned (categories 1 and 2) together with an irrelevant one (category 3). This was probably caused by the lack of facet metadata despite a high precision value of 1.
d) Both Q6 and Q12 resulted in high precisions (above 1). For Q6, all four returned sites were selected, out of which 3 were proven to be relevant (categories 1 and 2) while one was irrelevant (category 3). Its precision value of 1.5 was rather high due to the availability of metadata. On the other hand, 100 sites were selected out of the 140 that returned for Q12. A total of 61 sites were found to be relevant, which was a big proportion of the 66 relevant sites garnered by all the queries, with a high precision value of 1.095.

TABLE IV
PRECISION OF CLOUDMINING SE

Query	Sites no.	Sites	Categories					P
			1	2	3	4	5	
Q1	3	3	1	0	2	0	0	0.67
Q2	4	2	1	0	1	0	0	1
Q3	2	1	0	0	1	0	0	0
Q4	2	2	1	0	1	0	0	1
Q5	100	8	1	4	3	0	0	0.75
Q6	4	4	2	1	1	0	0	1.5
Q7	2	2	1	0	1	0	0	1
Q8	3	3	1	0	2	0	0	0.67
Q9	4	4	1	0	3	0	0	0.5
Q10	2	2	0	0	2	0	0	0
Q11	2	2	1	0	1	0	0	1
Q12	140	100	48	13	21	6	12	1.12
T	268	133	58	18	39	6	12	0.76
% T			43.6	13.53	29.32	4.51	9.02	

3) VExSearch Precision

Earlier experiments with the two SEs (CloudMining and IMDb) showed that not all the 12 queries returned results. Total returned sites were 525 with 133 of them selected for a comparison study, as suggested by Kumar [54] and Uddin, et al. [51]. The majority of the selected query results were from Q12 as it returned the most relevant results. Table V illustrates the relevant statistics for the 12 queries and the selected sites. While it is clear from the table that almost 73% of the results were relevant sites (categories 1 and 2), less than 30% of those were irrelevant with a mean precision of 1.64.

Table V analysed and summarised the query results with respect to the search precision for different queries (Q1 to Q12). As any SE on the Internet, the proposed VExSearch had better precision than IMDb and CloudMining because it adopted additional features and an item-based search that helped produce more accurate matches and similarity results. Given below are some analyses for the VExSearch:

TABLE V
PRECISION OF VEXSEARCH SE

Queries	Sites no.	Sites	Number of returned results for Categories					P
			1	2	3	4	5	
Q1	2	2	1	0	1	0	0	1
Q2	6	2	1	1	0	0	0	1.5
Q3	3	1	1	0	0	0	0	2
Q4	2	2	1	0	1	0	0	1
Q5	316	100	3	3	4	0	0	0.9
Q6	39	9	9	0	0	0	0	2
Q7	1	1	1	0	0	0	0	2
Q8	1	1	1	0	0	0	0	2
Q9	1	1	1	0	0	0	0	2
Q10	3	3	3	0	0	0	0	2
Q11	1	1	1	0	0	0	0	2
Q12	150	100	65	5	30	0	0	1.35
Total	525	133	88	9	36	0	0	1.64
% Total returned results			66.16	6.766	27.0			

a) The results for Q1, Q2, and Q4, as shown in the SERP, had a higher precision value. The order of queries showed matching and similarity results, together with a facet and an item-based search. The matching results retrieved all the relevant sites that were in the IMDb dataset. And because of that, the precision value for those queries was 1.

b) Respectively, the number of sites selected for Q5 was 10, and 100 for Q12. The returning results on the SERP showed matches with very high precisions. The most relevant site was placed in category 1, while the slightly less relevant one was added in category 2. Both categories retrieved more than 100 sites. Aside from this, the SERP showed a total of 30 irrelevant sites (category 3).

c) For Q3, Q6, Q7, Q8, Q9, Q10 and Q11 with respect to the SERP, their results showed a higher matching precision as well as similarities in the features, facet and item-based search. The sites selected the same SERP matching query results. The results then retrieved all the relevant sites in the IMDb dataset, which meant more than the precision value for those in category 2 queries. Summary of precision performance comparison is shown in Fig. 11.

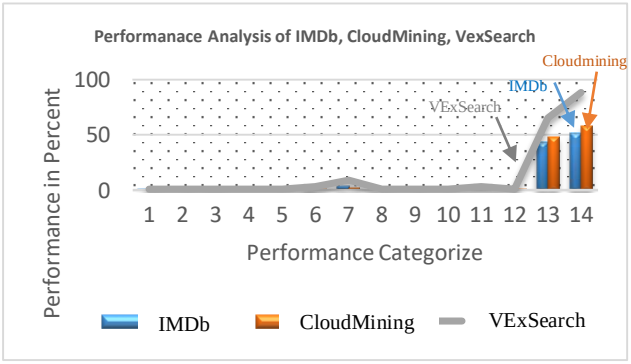

Fig 11 Performance comparison between of IMDb, Cloudmining and VExSearch

The table shows that the total returned results for category 1 was 66%. This was better than the 38% IMDb dataset results and CloudMining's 43%. Fig. 14 slots in the plot of search queries, comparing the precision values of IMDb, CloudMining and VExSearch.

In terms of search results, the graph shows the precision value of the retrieval performance capabilities at each query point. It summarised the effectiveness of all systems and the algorithms used per query method. The VExSearch outperformed the other two algorithms in CloudMining and IMDb. IMDb precision values started below -0.1 with a single query. As the queries grew, the precision became zeros before slightly increasing. This showed that there were no returns for the queries or that they had garnered results that were not related. Similarly, the results for CloudMining followed a comparable pattern.

On the other hand, the results for the VExSearch started at precision values above 0.9. In general, the values can increase when more queries are being created. The covering of the VExSearch was higher than the other systems. For instance, at query Q10, Fig. 12 shows that the highest precision value reached was 2, which was less than that of the VExSearch for the same query. Quite simply, the VExSearch retrieved more relevant data compared to the other system dynamics. Thus it made higher precision improvements with selected queries, as shown in the VExSearch performance capability curves in Fig. 12.

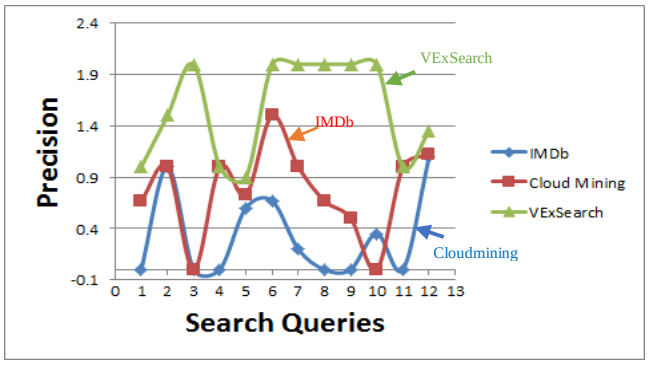

Fig 12 Precisions of IMDb, CloudMining and VExSearch

Referring to the ranking of the precision results for each database, in fig. 14, we can see that the precision on the Q3 position (green oval on the left) for VexSearch is very high as compared to both CloudMining and IMDb. At other positions, despite the precision of the other two databases increases slightly, VexSearch precision remain constantly high. As can be seen in figure 14, the precision values for VexSearch database are high at the 5th to 10th positions which confirm the superiority of VexSearch.

B. *Relative Recall*

'Recall' is based on the measure that determines if an item is retrievable. Thus, the notion 'recall' reflects if a retrieval system is able to gain relevant information from its collection. The calculation for relative recall is as follows:

$$\text{Relative Recall} = \frac{\text{Sum of sites retrieved by the two SE}}{\text{Total number of sites retrieved by SE}} \quad (3)$$

The relative recall for IMDb, CloudMining and VExSearch is calculated and presented in Table VI. As evident from the above table, the total relative recall for IMDb was 2.83; 8.465 for CloudMining and 9.12 for VExSearch. In the the VExSearch, the highest relative recall value was for search queries (1), followed by the least relative recall value of Q12 (0.28). The least relative recall value of search query Q2 (0.75).

The relative recall for IMDb, CloudMining and VExSearch

is calculated and presented in Table VI. As evident from the above table, the total relative recall for IMDb was 2.83; 8.465 for CloudMining and 9.12 for VExSearch. In the the VExSearch, the highest relative recall value was for search queries (1), followed by the least relative recall value of Q12 (0.28). The least relative recall value of search query Q2 (0.75).

The query results with respect to the search precision for the different queries (Q1 to Q12) could be analysed and summarised from the table below.

TABLE VI
RELATIVE RECALL OF IMDB, CLOUDMINING AND VEXSEARCH

Queries	IMDb		CloudMining		VExSearch	
	Sites no.	Recall	Sites no.	Recall	sites no	Recall
Q1	0	0	3	1	2	1
Q2	2	0.25	4	0.67	6	0.75
Q3	0	0	2	1	3	1
Q4	0	0	2	1	2	1
Q5	10	0.03	100	0.9	316	0.96
Q6	9	0.9	4	0.3	39	0.81
Q7	10	0.18	2	0.16	1	0.09
Q8	0	0	3	1	1	1
Q9	0	0	4	1	1	1
Q10	10	0.76	2	0.16	3	0.23
Q11	0	0	2	1	1	1
Q12	369	0.71	140	0.275	150	0.28
Total	410	2.83	268	8.465	525	9.12

Other than the aforementioned additional features, the item-based search was also the reason that the VExSearch continually produced better results than the IMDb and CloudMining. It went on to achieve more accurate matches and similarity results. Given below are some analyses for the VExSearch:

a) The query results for Q1, Q3, Q4, Q8, Q9 and Q11, when compared to the SERP, showed higher relative recall values. The VExSearch showed the exact relative recall values of the other SEs, together with matching results that retrieved all the relevant sites in IMDb.
b) The results for Q2, Q7, Q10 and Q12 showed that the VExSearch retrieved only the sites that were saved in IMDb.
c) The results for Q5 and Q6 showed the highest relative recall values when compared to the SERP from the total number of sites. The relative recall value for Q5 was 0.9 out of 316 (total number of sites). For comparison analysis, the relative recall values of IMDb, CloudMining and VExSearch are shown in Fig. 13.

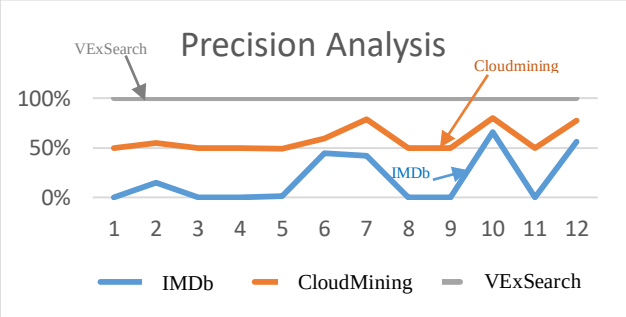

Fig 13 Comparative Performance Analysis of Relative Recall of IMDb, Cloud mining and VExSearch

The results were plotted into a search queries and a relative recall graph that was interpolated at 12 points to show the precision of the relative recall performance capabilities at each query point in the search results. Comparative performance analysis of relative recall, IMDb, Cloudmining and VExSearch is given in Fig 13.

Fig. 14 summarised the effectiveness of all systems and the algorithms used per graph. The item-based search and IMDb dynamics outperformed the other algorithms. CloudMining algorithms followed to VExSearch. The relative recall values for IMDb started below 0.1 because this type of SE had given more irrelevant results. The relative recall for CloudMining started similar with the VExSearch, but CloudMining gave irrelevant results due to lack of metadata which then slightly covered all retrieval of relevant data for selected queries. Query points in the search results were between the two curves. The VExSearch returned relevant results to address the selected queries, making the highest recall improvements with each query and returning the kinds of data that were most likely to be relevant.

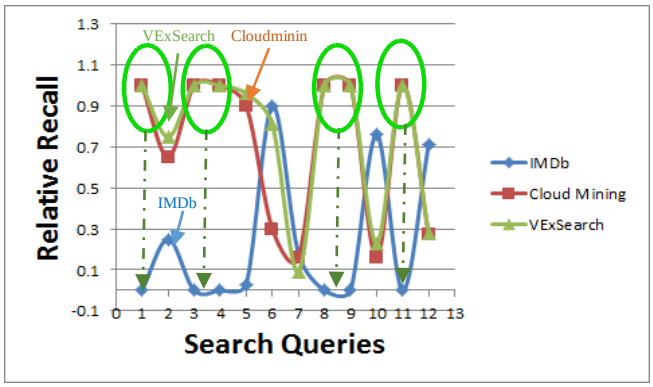

Fig 14 Relative Recall of IMDB, Cloudmining and VExSearch

The relative recall ranking for each database is as shown in fig. 16. We can see that the recall on the Q1 and Q3 positions (the two green ovals on the left) for VexSearch is very high for VexSearch and CloudMining as compared to IMDb but VexSearch recall remain relatively higher than CloudMining and IMDb at other positions. Thus, on recall, VexSearch wins over the other two as well as it covers more queries.

TABLE VII
SE EVALUATION CRITERIA

No	Category	Criteria
1	Relevant	"If the content of the web page closely matched the subject matter of the search query, then it was categorized as "relevant" and it was given a score of 1".
2	Same query, but different web addresses relevant	"If the content of the web page is not closely related to the subject matter, but consists of some relevant aspects to the subject matter of the search query, then it was categorized as "relevant" and it was given a score of 1".
3	Non-relevant	"If the content of the web page is not related to the subject matter of the search query, then it was categorized as "Non-relevant" and it was given a score of 0".
4	Duplicate document	"If the content of the web page consisted of a whole series of Duplicate document, also define non-relevant, it was given a score of 0".

C. *Performance Evaluation*

Khan, et al. [55] conducted an in-depth analysis to evaluate the search performances of keyword-based SEs (Google and Yahoo) and semantic web-based SEs (Lexxe, Bing, and DuckDuckGo). A total of 10 queries that reflected varied subject matters were employed on the SE studies to determine their precision ratios. As a result, a framework was developed for the semantic web-based SE to address issues related to polysemy and synonym [56], apart from replacing the web crawler with an ontology-based crawler to gain high precision

outcomes.

Kim, et al. [57] prepared 40 queries for a situation where a user searches details of a particular movie with some information, for instance, several title words, year of the movie released or the actor's name. Table VIII presents the details that were integrated into one keyword query. As the queries had been developed to retrieve a movie, the manual search did not produce any relevant information.

TABLE VIII
QUERY LISTS

No	Query	Description
Q1	"Love Letter Iwai"	Movie with title "Love Letter", directed by Iwai Shynji
Q2	"Ziyi Zhang hidden tiger"	Crouching Tiger Hidden Dragon" featuring Ziyi Zhang
Q3	"Meg Ryan war"	A war movie featuring Meg Ryan"
Q4	"Redemption crime"	The Shawshank Redemption", a crime movie
Q5	"Brokeback Ang Lee"	The Brokeback Mountain directed by Ang Lee

The initial experiment by Kim, et al. [57] on the IMDb dataset consisted of 437,281 documents or XML records. Each query corresponded to a movie and was constructed from text data. The attributes types were "title", "year", "release data"," language", "genre", "country", "location", "color info", "cast", "team" and "plot". Document content consisted mostly of keywords, with an exception of the plot element. The mapping probabilities could be estimated with high accuracy because the elements had minimal overlap in the word distribution.

The search experiments were conducted on the basis of the study by Park, et al. [58] to evaluate the search performance of three SEs (VExSearch, CloudMining and IMDb). Comparisons were made according to keyword definitions, the relationship between keywords and web pages, and the number of relevant documents retrieved. Five queries, which represented a broad range of IMDb queries and were adapted from Kim, et al. [57], were used and submitted to the SEs. The first 10 documents retrieved were evaluated using binary human relevance judgment, where every document was classified as 'relevant' or 'irrelevant'. The precision and recall ratio of the keyword-based SEs were calculated for five documents. Each query was executed in two SEs at nearly the same time to avoid temporal variations and to retrieve relevant data from each SE. On the basis of the first 10 documents retrieved by IMDb, CloudMining and VExSearch, the criteria for evaluation was defined using the four categories depicted in Table VIII.

TABLE IX
NUMBER OF RELEVANT DOCUMENTS RETRIEVER FOR IMDB

Queries	Total no. of Sites	Selected Sites	Categories			
			1	2	3	4
Q1	2	2	0	0	2	0
Q2	0	0	0	0	0	0
Q3	2	2	0	1	1	0
Q4	14	10	1	0	9	0
Q5	0	0	0	0	0	0
Total	18	14	1	1	13	0

1. IMDb SE

During the search process, most of the five queries returned results within 18 sites. Out of those 18 sites, 14 were selected for a comparison study with Kim. The majority of the selected query results came from Q5 who returned the most results. Table IX illustrates the relevant statistics for the five queries and the selected sites. Notice from the table that only one site was relevant (category 1); another was less relevant (category 2); 13 fell under non-relevant categories (category 3); and there was a presence of a duplicate document (category 4).

The abovementioned table presents the query results with respect to the relevant documents retriever for the different queries (Q1 to Q5), which could be analysed and summarised. A part of the main aim during the search process was to produce the exact query matches with relevant or irrelevant sites. The IMDb needed that of general improvement in keywords and polysemic words performance, as shown in the analyses given below:

a) For Q1 and Q3, the non-relevant sites (category 3) were 2. The results for both queries, as shown in the SERP, returned no relevant documents.
b) The results for Q2 and Q5 showed no relevant documents.
c) For Q4, 10 sites were selected from the total number of sites. The relevant sites (category 1) were among the nine non-relevant ones (category 3). IMDb found one matching site relevant to Q4 while the other nine were irrelevant sites. The searching process in IMDb was simply looking for the exact query matches but lacked the keyword-related meanings.

2. CloudMining SE

When searching for the five queries, not all of them returned results. The total sites that returned were 129. Out of that, 16 were selected to be compared with the study by Kim. The majority of the selected query results were from Q5 because it returned the most relevant sites. Table X illustrates the relevant statistics for the five queries and the selected sites. The table shows the query results with respect to the search evaluation performance for different queries (Q1 to Q5) that could be analysed and summarised. CloudMining's better results than IMDb was a no-brainer, pointing to the fact that CloudMining had adopted additional features to help it get more accurate matches and similarity results. Given below are some of the analyses:

a) As shown in the SERP, the results for Q1and Q2 returned with two selected sites. The relevant document retrieved for the selected queries was one, while the partial result was also one.
b) As shown in the SERP, the results for Q3 and Q4 returned. Two relevant sites (category 1) were retrieved for Q3, while 122 sites came back for Q4. The irrelevant sites (category 3) recorded the highest number in the SERP because there were no query matches.
c) As shown in the SERP, the results for Q5 returned with one irrelevant site (category 3) which was unable to be compared.

The query results for Q4 found that relevant documents were retrieved by CloudMining. Overall, this performance was the best in terms of average percentage, which was higher than IMDb. The number of relevant documents retrieved by CloudMining was based on the features it used to reduce the user's effort in searching for relevant sites. However, a problem will likely show up with the metadata keywords when a user conducts a deep search into the repository layer.

As CloudMining is faced with the growing negative effect of large dataset performance, the way forward was to improve its keywords and polysemic words. Table X, in essence,

brought to light the importance of adapting users to learning the capability performance of the system fundamentals.

TABLE X
NUMBER OF RELEVANT DOCUMENTS RETRIEVER FOR CLOUDMINING

Queries	Total no. of Sites	Selected Sites	Categories			
			1	2	3	4
Q1	2	2	1	1	0	0
Q2	2	2	1	1	0	0
Q3	2	1	1	0	0	0
Q4	122	10	1	0	9	0
Q5	1	1	0	0	1	0
Total	129	16	4	2	10	0

3. VExSearch SE

The outcome of the 11 search queries was largely fruitful. A total of 165 sites returned results, out of which 16 were selected for comparison analysis with the studies by Kim. Q1 returned the most results, which made up the majority of query results selected. Table XI illustrates the relevant statistics for the five queries and the selected sites. As shown in the table, nine sites were relevant (category 1) while four others were less relevant (category 2). Three sites were grouped under the non-relevant category (category 3) and no duplicate document (category 4) was found.

The table shows the query results with respect to the search precision for the different queries (Q1 to Q5) that could be analysed and summarised. The VExSearch served as the better search engine here, as shown in Table XI. Its additional features and an item-based search transcended boundaries in order to give users a more accurate match and similarity of the content. Given below are some of the analyses:

TABLE XI
NUMBER OF RELEVANT DOCUMENTS RETRIEVER FOR VEXSEARCH

Queries	Total no. of Sites	Selected Sites	Categories			
			1	2	3	4
Q1	1	1	1	0	0	0
Q2	1	1	1	0	0	0
Q3	3	3	3	0	0	0
Q4	159	10	3	4	3	0
Q5	1	1	1	0	0	0
Total	165	16	9	4	3	0

a) As shown in the SERP, the results for Q1, Q2 and Q5 returned with one relevant document. The VExSearch garnered matching results that retrieved all the relevant sites in IMDb.
b) All queries were selected for Q3, with three more relevant sites returned (category 1). The SERP showed that VExSearch had the ability to search and find all sites related to the selected keywords.
c) As shown in the SERP, the results for Q4 returned with 158 sites, out of which 10 showed a higher relevant site matching (category 1). Although the same query had returned different relevant (category 2) and non-relevant (category 3) results, it showed a higher match and similarity display with the selected sites. The matching results retrieved all relevant sites that were saved in the IMDb dataset.

Put together as a search engine, the VExSearch's keyword and polysemic word performance did work better, coupled with a positive growth.

In this paper, we evaluated the performance of Semantic Search Engine and Keyword Based Search Engine. Needless to say, in terms of overall performance, the VExSearch retrieved more relevant documents compared to the other search engines. Our experiment showed that the performance of the VExSearch Semantic Search Engine was undoubtedly higher than the Keyword Based Search Engine as it had reduced the problems with synonyms and polysemic words.

The number of relevant documents retrieved by each search engine for the first ten retrieved documents is shown in Table XII. It shows the average improvement percentage of the proposed approaches in terms of the five cases of keyword, as compared to the other approaches with different keyword in IMDb. The improvement percentage achieved by the proposed approach was listed with appropriate different keyword values. For instance, the VExSearch obtained an average improvement of 20% in terms of recall. The average improvement percentage was calculated for all the query values.

TABLE XII
SUMMARIES NUMBER OF RELEVANT QUERY RETRIEVER

Query	IMDb	CloudMining	VExSearch
Q1	0	1	1
Q2	0	1	1
Q3	1	2	3
Q4	10	122	159
Q5	0	1	1
Total	11	127	165
Average %	5.5	63.5	82.5

Fig. 15 presents the overall graphical precision ratios of the SEs for the first ten documents, which clearly showed that the VExSearch retrieved more relevant documents than IMDb and CloudMining. IMDb retrieved approximately 5.5%, while CloudMining 63% for the original query. The precision ratios of IMDb, CloudMining and VExSearch are also shown in Fig. 15. The results were plotted into a precision graph interpolated at 10 points to show the precision value of the retrieval performance capabilities at each query point in the search results.

Fig. 15 summarises the effectiveness of all systems and the algorithms used per graph. The item-based search and IMDb outperformed the other algorithms, followed by the CloudMining and VExSearch algorithms. No relevant data was found for the selected queries or different results emerged that were not related to the selected queries. The system has had precision values that started at 0 before slightly increasing at every query point. The query point had a more relevant data. For example, at Q4, the Fig. 15 showed a high precision that reached 160 for the selected query as well as more relevant data retrieved by VExSearch.

Fig. 15 shows the relative precisions for each search engine. It can be seen that VexSearch has higher relative precision at Q4, outperforming CloudMining, but the difference is small. It can be noticed also IMDb is very low in relative precision at Q4, as a result of it giving a larger number of results judged as completely irrelevant. VexSearch produce more relevant results for Q4 than the other two. Overall across Q1 to Q5, VexSearch outperforms CloudMining slightly and way too big for IMDb.

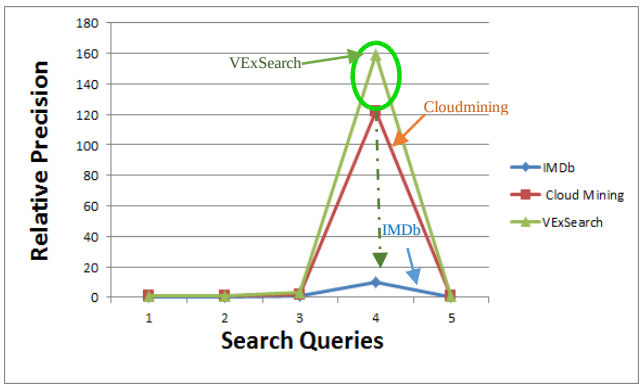

Fig 15 Precision Ratios of SEs for First Ten Documents

V. COMPARISON BETWEEN SEs

The evaluation metrics were used to measure the performance, the standard recall, and the precision of the proposed approached VExSearch. These metrics judge how relevant a set of search results to the users. The recall, precision computed at different values of queries. In each of the result obtained by the proposed approaches are compared with the best run obtained by each baseline approach at individual query. Furthermore, the average recall, precision in the two cases of queries was compared. The reason of considering the average values in the two cases was to find the total improvement of VExSearch approach.

The following provides discussion of the finding according to the result on the dataset with two cases of different keywords. The result obtained by the proposed approaches was compared and discussed in different situation in each keyword.

From the experimental result, the proposed VExSearch engine developed outperformed IMDb and CloudMining SE in term of recall, precision and the mean precision. The results proved that VExSearch was able to enhance the learned, suggestions or recommendations of users and thus improve the relevance quality. The VExSearch expect information seekers to make possibly vague information needs explicit in textual search queries and assess large amounts of individual search results.

Furthermore, for first keywords cases, the results show that VExSearch engines obtained an improvement in term of recall in all queries compared with the proposed approaches IMDb and CloudMining engines respectively. However, the VExSearch approach have better precision values, which subsequently result in high mean precision. This is because learned, suggestions or recommendations and the search results and facets should be extended with different views. In this case more relevant items can be retrieved compared to the total number of relevant items in others benchmarks.

In the case of the second query, the evaluations were on the search performance of VExSearch and keyword based SE. In the experiment, VExSearch retrieved more relevant documents compared to CloudMining and IMDb. In term of overall performance, VExSearch retrieved more relevant documents compared to all selected SE. VExSearch would reduce the problem of partial information spanning over many elements or if the user doesn't have the right words to use in the query. The experimental results show that search performance of VExSearch was higher than any keyword based SE.

The overall results illustrate that FS combines faceted navigation with full text search. This provides the user with the opportunity to work successfully with content that is semi-structured. Full text search is used to obtain those results which do not have structural characteristics. While on the other hand, faceted navigation provides a means of browsing and refining by metadata structured information. This greatly reduces the chance of generating no results, while still providing refinements when too many results are returned. The searcher can create an interface that fosters improvements to all aspects and stages of the search. Better interface designs will assist the users in articulating better queries, help them understand the results and facilitate query modifications if necessary.

TABLE XIII SHOWS THE EVALUATION OF ALGORITHM THAT HAS BEEN IMPLEMENTED AND IMPROVED USING THE SAME DATASETS BY USING EXPLORATORY SEARCH FEATURES.

No	Exploratory Search Features	CloudMining	IMDb	VExSearch
1	Add search views as pluggable widgets. These could include a map view and a space-time view.	□		□
2	Support feature different skins, for different devices such as tablets or phones.			
3	Entity extraction module should be provided			
4	The lack of metadata issue.			□
5	Each instance, could make use of a social module to vote, comment, edit or curate the documents found.			
6	Similarity search should be improved in order to allow for queries made of items present in the document collection.			□
7	Using the cloud computing to improve the performance of the system.	□		□
8	Support learning and understanding			□

This paper evaluated the search performance of IMDb, CloudMining and VExSearch. From the experiments, it is observed that VExSearch retrieved more relevant information compared to IMDb and CloudMining. Also, when dealing with a large dataset, VExSearch was more accurate than other two. VExSearch was capable of presenting a large dataset and can present multi datasets at the same time. Thus, it would reduce the problem of partial information spanning over many elements or if the user doesn't have the right words to write the query. The results of the experiments show that search performance of VExSearch was better than IMDb and CloudMining. Table XIII shows that the features presented are still inadequate in differentiating between CloudMining, VExSearch and IMDb when different sizes were used. However, the proposed algorithm can work on large scale datasets. IMDb don't have any features to show in the table.

VExSearch can be deployed on the cloud due to the many benefits. However, some challenges come with it such as the added distance and components between the users and VExSearch applications in the cloud can cause performance and availability challenges. Secondly, the increased latency and reduced bandwidth between user who do the search in VExSearch and services deployed to the cloud can cause performance issues in the VExSearch application. Finally, the shared environment and lack of control over the connectivity between VExSearch servers in the cloud could mean connections between servers do not meet the needs of the application design. Some VExSearch functions require highly specialized hardware as part of communication requirements between systems that use multiple datasets which lead to additional latency that can have significant impact on performance.

VI. CONCLUSIONS

The general conclusions that can be drawn from the research on the issue of insufficient number of relevant results are derived from the investigation on the impact of ambiguity in query upon search outcomes. It was hypothesised that unambiguous queries generate highly relevant results. Besides, this study looked into the inefficient results sorting by analysing three associated approaches.

Enhancement of web search results relevancy seems to have gained interest among many in these recent times as the web is a trove of massive data and information with varied structures, thus posing a challenge when one attempts to retrieve the most accurate search results. Thus, interactive graphics and visualisation approaches have been recommended for displays to be able to control the massive amount of search results, along with presentation of some features of the related websites. Additionally, the rebuilding and reformulation of queries are managed by SEs, where relevant results are provided in the hit list. The format of textual list and the long list of matching documents pose a problem for users to retrieve the most relevant results.

VExSearch retrieved more relevant documents compared to all selected SEs (IMDb and CloudMining). The research also shows that if the VExSearch have a large dataset it will be more accurate than other systems. Besides that, the VExSearch is capable of presenting large datasets and can present multiple datasets at the same time. This would reduce the problems of synonyms and polysemy words.

The proposed VExSearch system was tested and evaluated in two parts, namely, the searching capabilities of SEs and the A performance of SE parts. After testing the issues, the results including precision value and relative recall were able to address the SE problems. The relevant results returned by VExSearch proved its capability of resolving the ambiguity of SE results. To further verify the evaluation issues, we compared the results with those of other SEs. The comparison showed that the VExSearch system provides more efficient, reliable and accurate results than its counterparts. Moreover, the system can show all the documents stored in the system relative to the benchmark and other approaches.

Having achieved the objectives of this study, several contributions have been identified in VExSearch work of extend the current paradigm of FS into the one of exploratory search as follows: a. The search results and facets have been extended with different views. b. Query by example, has been integrated with Bayesian Sets as it reduces the handling of complex content based searches to choosing the right plugin.

ACKNOWLEDGMENT

This research was sponsored and supported under the Universiti Tenaga Nasional (UNITEN) internal grant no J510050783 (2018). Many thanks to the Innovation & Research Management Center (iRMC), UNITEN who provided their assistance and expertise during the research.

REFERENCES

[1] R. Qumsiyeh and Y.-K. Ng, "Searching web documents using a summarization approach," International Journal of Web Information Systems, vol. 12, pp. 83-101, 2016.
[2] P. Bailey and N. Craswell, "Ingrams: A Neuropsychological Explanation for Why People Search," in Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, 2016, pp. 877-880.
[3] A. Qin, Y. Yuan, D. Tan, P. Sun, X. Zhang, H. Cao, et al., "Feisu: Fast Query Execution over Heterogeneous Data Sources on Large-Scale Clusters," in Data Engineering (ICDE), 2017 IEEE 33rd International Conference on, 2017, pp. 1173-1182.
[4] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "SAFQuery: a simple and flexible advanced Web search interface," The Electronic Library, vol. 34, pp. 155-168, 2016.
[5] S. Dumais, E. Cutrell, J. J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins, "Stuff I've seen: a system for personal information retrieval and re-use," in ACM SIGIR Forum, 2016, pp. 28-35.
[6] S.-X. Li and J.-S. Wang, "Improved cuckoo search algorithm with novel searching mechanism for solving unconstrained function optimization problem," IAENG International Journal of Computer Science, vol. 44, no. 1, pp. 8-12, 2017.
[7] K. Collins-Thompson, S. Y. Rieh, C. C. Haynes, and R. Syed, "Assessing Learning Outcomes in Web Search: A Comparison of Tasks and Query Strategies," in Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval, 2016, pp. 163-172.
[8] G. Marchionini, "Exploratory search: from finding to understanding," Communications of the ACM, vol. 49, pp. 41-46, 2006.
[9] J. Savoy, "Why do successful search systems fail for some topics," in Proceedings of the 2007 ACM symposium on Applied computing, 2007, pp. 872-877.
[10] S. Gallova, "Fuzzy ontology and information access on the web," IAENG International Journal of Computer Science, vol. 34, no. 2, 234-238, 2007.
[11] J. Teevan, K. Collins-Thompson, R. W. White, S. T. Dumais, and Y. Kim, "Slow search: Information retrieval without time constraints," in Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval, p. 1, 2013.
[12] R. W. White, G. Marchionini, and G. Muresan, "Evaluating exploratory search systems: Introduction to special topic issue of information processing and management," ed: Pergamon, 2008.
[13] M. L. Wilson, "Improving exploratory search interfaces: Adding value or information overload?," 2008.
[14] M. J. Bates, "The design of browsing and berrypicking techniques for the online search interface," Online review, vol. 13, pp. 407-424, 1989.
[15] M. N. Mahdi, A. R. Ahmad, and R. Ismail, "Paradigm Extension of Faceted Search Techniques A Review," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 9, pp. 149-153, 2017.
[16] D. Tunkelang, "Faceted search," Synthesis lectures on information concepts, retrieval, and services, vol. 1, pp. 1-80, 2009.
[17] R. K. Logan, "The Service and Disservice of Information Overload: Ways To Cope," in THE FUTURE INFORMATION SOCIETY: Social and Technological Problems, ed: World Scientific, 2017, pp. 273-284.
[18] W. Dakka, R. Dayal, and P. G. Ipeirotis, "Automatic discovery of useful facet terms," in SIGIR Faceted Search Workshop, 2006, pp. 18-22.
[19] R. W. White and R. A. Roth, "Exploratory search: beyond the query-response paradigm (Synthesis lectures on information concepts, retrieval & services)," Morgan and Claypool Publishers, vol. 3, 2009.
[20] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, "Query expansion for short queries by mining user logs," IEEE Trans. Knowl. Data Eng, vol. 15, pp. 829-839, 2002.

[21] C. Costa and M. Y. Santos, "Big Data: state-of-the-art concepts, techniques, technologies, modeling approaches and research challenges," 2017.
[22] J. Teevan, E. Adar, R. Jones, and M. A. Potts, "Information re-retrieval: repeat queries in Yahoo's logs," in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 2007, pp. 151-158.
[23] R. Jones, B. Rey, O. Madani, and W. Greiner, "Generating query substitutions," in Proceedings of the 15th international conference on World Wide Web, 2006, pp. 387-396.
[24] R. W. White and S. M. Drucker, "Investigating behavioral variability in web search," in Proceedings of the 16th international conference on World Wide Web, 2007, pp. 21-30.
[25] N. Dalum Hansen, K. Mølbak, I. J. Cox, and C. Lioma, "Seasonal Web Search Query Selection for Influenza-Like Illness (ILI) Estimation," in Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 1197-1200.
[26] C. Ahlberg, C. Williamson, and B. Shneiderman, "Dynamic queries for information exploration: An implementation and evaluation," in Proceedings of the SIGCHI conference on Human factors in computing systems, 1992, pp. 619-626.
[27] K. Doan, C. Plaisant, B. Shneiderman, and T. Bruns, "Query previews for networked information systems: A case study with NASA environmental data," SIGMOD Record, vol. 26, pp. 75-81, 1997.
[28] E. Tanin, B. Shneiderman, and H. Xie, "Browsing large online data tables using generalized query previews," Information Systems, vol. 32, pp. 402-423, 2007.
[29] R. Capra and G. Marchionini, "Faceted Exploratory Search Using the Relation Browser," in NSF Workshop on Information Seeking Support Systems, 2009, pp. 81-83.
[30] B. Lee, G. Smith, G. G. Robertson, M. Czerwinski, and D. S. Tan, "FacetLens: exposing trends and relationships to support sensemaking within faceted datasets," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2009, pp. 1293-1302.
[31] ManyEyes, BM T.J. Watson Research Center (Yorktown, Cambridge) https://researcher.watson.ibm.com/researcher/view_group.php?id=7352 "ManyEyes."
[32] F. B. Viegas, M. Wattenberg, and J. Feinberg, "Participatory visualization with wordle," IEEE transactions on visualization and computer graphics, vol. 15, 2009.
[33] E. R. Tufte, "Envisioning information," Optometry & Vision Science, vol. 68, pp. 322-324, 1991.
[34] W. L. Hays, "The Visual Display of Quantitative Information," Psyccritiques, vol. 29, p. 671, 1984.
[35] A. Ksikes, "Towards exploratory faceted search systems," University of Cambridge, 2014.
[36] A. Swartz, "web.py," January 16, 2013 2013.
[37] A. Aksyonoff, Introduction to Search with Sphinx: From installation to relevance tuning: " O'Reilly Media, Inc.", 2011.
[38] Craigslist, Craigslist is an American classified advertisement website "Craigslist."
[39] livingsocial, online marketplace that allows its registered users to buy and share things to do "Living Social."
[40] metacafe, video-sharing website that specializes in short-form video entertainment "Metacafe."
[41] M. Hearst, "Design recommendations for hierarchical faceted search interfaces," in ACM SIGIR workshop on faceted search, 2006, pp. 1-5.
[42] Flamenco, search interface framework has the primary design goal of allowing users to move through large information spaces in a flexible manner without feeling lost. "Flamenco ".
[43] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst, "Faceted metadata for image search and browsing," in Proceedings of the SIGCHI conference on Human factors in computing systems, 2003, pp. 401-408.
[44] D. Cutting, free and open-source information retrieval software library "Lucene," ed, 2004.
[45] T. Grainger, T. Potter, and Y. Seeley, Solr in action: Manning Cherry Hill, 2014.
[46] Z. Ghahramani and K. A. Heller, "Bayesian sets," in Advances in neural information processing systems, 2006, pp. 435-442.
[47] C. Li and H. Hao, "Likelihood and Bayesian estimation in stress strength model from generalized exponential distribution containing outliers," IAENG International Journal of Applied Mathematics, vol. 46, no. 2, pp. 155-159, 2016.
[48] P. Mestre, J. Matias, A. Correia, and C. Serôdio, "Direct search optimization application programming interface with remote access," IAENG International Journal of Applied Mathematics, vol. 40, no. 4, pp251-261, 2010.
[49] Y. Luo, W. Wang, X. Lin, X. Zhou, J. Wang, and K. Li, "Spark2: Top-k keyword query in relational databases," IEEE Transactions on Knowledge and Data Engineering, vol. 23, pp. 1763-1780, 2011.
[50] B. Kumar and S. Pavithra, "Evaluating the searching capabilities of search engines and metasearch engines: A comparative study," 2010.
[51] J. Uddin, S. M. Ahmad, S. U. Jan, and A. Reba, "Precision and Relative Recall of Search Engines using Education Keywords: A Comparative study of Google, Yahoo and Refseek," PUTAJ-Humanities and Social Sciences, vol. 25, pp. 99-112, 2017.
[52] T. A. Usmani, D. Pant, and A. K. Bhatt, "A comparative study of google and bing search engines in context of precision and relative recall parameter," International Journal on Computer Science and Engineering, vol. 4, p. 21, 2012.
[53] B. S. Kumar and J. Prakash, "Precision and relative recall of search engines: A comparative study of Google and Yahoo," Singapore Journal of Library & Information Management, vol. 38, pp. 124-137, 2009.
[54] G. Kumar, "Top 10 search Engines List Learn more about them," 2016.
[55] J. A. Khan, D. Sangroha, M. Ahmad, and M. T. Rahman, "A performance evaluation of semantic based search engines and keyword based search engines," in Medical Imaging, m-Health and Emerging Communication Systems (MedCom), 2014 International Conference on, 2014, pp. 168-173.
[56] K. Abdalgader, "Clustering Short Text using a Centroid-Based Lexical Clustering Algorithm," IAENG International Journal of Computer Science, vol. 44, no. 4, pp523-536, 2017.
[57] J. Kim, X. Xue, and W. B. Croft, "A Probabilistic Retrieval Model for Semistructured Data," in ECIR, 2009, pp. 228-239.
[58] H.-C. Park, J. Park, K.-H. Kim, K. Kim, and G.-N. Wang, "An efficient generation mechanism of HMI information for heterogeneous PLCs," in Ubiquitous and Future Networks (ICUFN), 2015 Seventh International Conference on, 2015, pp. 46-48.