ISSN No. 0976-5697

Volume 5, No. 8, Nov-Dec 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Factverifier: Search Engine Based Question/Answering System to Verify Facts towards identifying and answering polar questions authoritatively by using pattern matching for web search engine

Gundeep Singh Bindra
Microsoft Corporation
Redmond, USA

Seema Khanna
Ministry of Communication and Information Technology
New Delhi, India

Dr. Harish Chaudhry
Indian Institute of Technology
New Delhi, India

Abstract: "Is the Eiffel Tower 324m tall?", "Was Thanksgiving Day on Nov 22 last year?", "Was Thanksgiving Day on Nov 22 in 2007?" If you have similar facts and you are not sure about them, verify the facts with your favourite search engine. With more and more complicated assertions people make these days, it is difficult to tell which is true and which is false. In this era of information technology where the major source of our information is Wikipedia, Google, Blogs etc. our belief and dependence in them is immense. This paper aims to discuss new techniques being evolved for using search engines beyond their basic purpose. Truth is a concern for most of us in this con world. It is an examination of the various search engines on how they respond to well known facts and even factual statements. In the name of VERITAS, the Paper proposes a new method of answering yes/no questions for the purpose of verification of facts based on pattern matching, we focus on questions that the search engine already knows the answers for. For example, "How tall is the Eiffel Tower?" In order to answer such questions the search engine applies mapping pattern matching its algorithm. This paper describes the algorithm that finds the answer to the queries in the form of a yes/no. Proposed here, is a method for evaluating the algorithm and report results that are obtained for a self-made question set. Finally, there are some suggestions for possible extensions of the methods. This conceptual paper is definitely expected to contribute to future research on similar and related topics as a spin off from this study.

Keywords: Search Engine, QandA, YES/NO Questions, Fact Verifier, TRUE/FALSE, Information Retrieval, Question Answering System, Answer extraction

I. INTRODUCTION

25GB of searches in second. Yes, that's the power of search today. Virtually countless searches performed daily have made the World Wide Web search engines the most heavily-used online services today. A major contributing factor is their ease of use. The core tasks for most of the search engines can be encapsulated as 1) query or user information request- do what is meant and not what is said!, 2) model for the Internet, Web representation-web page collection, documents, text, images, music, etc., and 3) ranking or matching function degree of relevance, recall, precision, similarity, etc. Their popularity is due, in part, also to their "How well are we doing? Are we doing human communication through shared knowledge? Let's look through the document side. On this side the languages are a natural language. They're people talking to people. They go out to search engines and they ask a question and the search engine gives these stupid answers. It has read a large proportion of the pages on the entire Web (which is of course amazing) but it doesn't understand any of them and it tries to answer the question on that basis. Obviously you get pretty unpredictable results.

The selected query doesn't need to exactly match the decision criteria .This gives the system a more human-like behavior. It produces results in a format that permits the user to interact dynamically to customize and personalize its search strategy. [9]. the commercially accomplished current web search engines undoubtedly demonstrate good performance, especially for homepage finding queries. However, their ability to find relevant information for hard queries such as those asking for opinions or summaries leaves much to be desired. These complicated user information needs can be satisfied by using question and answer (QA) services such as Yahoo! Answers, Baidu Zhidao , etc. whereby users can directly obtain answers rather than a list of potentially relevant documents. During the past few years, these Question Answering (CQA) services have been building huge question banks with answers. [8]. In order to avoid the lag time involved in waiting for a personal response, a QA service will automatically search the archives to see if the same question has been asked previously.

In question search, given a question as query, we are to return questions semantically equivalent or close to the queried question. Some methods have been investigated for tackling the problem of question search. For example, Jeon et al [3] have compared the uses of four different retrieval methods, i.e. vector space model, Okapi, language model, and translation-based model, within the setting of question search.

Design of any new intelligent search engine should be based on at least two main motivations: a) a logic that supports modes of reasoning which are approximate rather than exact This is essentially desirable because the web environment is majorly unstructured and imprecise and finding decision relevant and query-relevant information in this kind of environment is a challenging problem despite searches retrieving thousands of hits. b) Deduction, though this may seem less obvious given the huge stream of complex information. [16]

II. QUESTION-ANSWERING SYSTEMS

Using the current search engines, a user can find out whatever he wants to know. However, when users input keywords or questions, the results which are got from search engines are documents and not precise answers to questions. A happy situation is when QA system could accept natural language questions and give precise answers to the user. For instance, given the question ”Where is the capital of China”, QA system gives answers such as ”Beijing “or ”The capital of China is Beijing”, as against lots of documents containing the words ”capital” and ”China” that current search engines give. [4]

Question Answering (QA) is a computer science discipline within the fields of information retrieval and natural language processing (NLP) which is concerned with building systems that automatically answer questions posed by humans in a natural language. A QA implementation, usually a computer program, may construct its answers by querying a structured database of knowledge or information, usually a knowledge base. More commonly, QA systems can pull answers from an unstructured collection of natural language documents. 1

The current search engine is apparently an ideal source as knowledge of Q&A system, but any search engine can't cover 100 percent of online information resources. The repetition rate of results that different search engines return is between 10-30 percent [14]. Thus a search engine usually can't provide the results that user want to find. Information retrieval is the key module of QA system. [7]

Search engines, including the phenomenally popular ones are incredible in more ways than one. However, what they conspicuously lack is an inherent deduction capability- the capability to put together an answer to a query by drawing on the immense treasure of the body of information which is housed in the knowledge base. It is this capability that distinguishes a question- answering system, Q/A system from a search engine

A long drawn history stands behind the construction of Q/A systems. The seventies and eighties can justifiably boast of an all-time high in interest in Q/A systems for some time. However, the realization that the available tools were inadequate for construction of systems having satisfactory question-answering capabilities led to its decline. However, Q/A systems in the form of domain-restricted expert systems have proved to be of value, and are growing in versatility, visibility and importance.

The mother of search engines is the Web, for they owe their birth to it A typical search engine is not designed to come up with answers to queries exemplified by “How many Ph.D. degrees in computer science were granted by Princeton University in 1996”' or “What is the name and affiliation of the leading eye surgeon in Boston?” or “What is the age of the oldest son of the President of Finland?” or “What is the fastest way of getting from Paris to London?”

As a very simple illustration of the use of an epistemic lexicon, consider the query “How many horses received the Ph.D. degree from Princeton University in 1996.” No existing search engine would come up with the correct answer viz. “Zero,” since a horse cannot be a recipient of a Ph.D. degree. “To generate the correct answer, the attribute Eligibility in the Ph.D. entry in EL should contain the condition “Human, usually over twenty years of age.” Another example is which country is the biggest producer of tungsten (that is question 14 from TREC8). The expected answer is “China”. [13]

III. OBJECTIVES

The overriding objective is to be able to output an authoritative answer YES/NO to the user query.
The secondary objective is to exhaustively explore the various search engines' responses to “factual” queries.

a. Be able to leverage the use of Search engines to be used as Fact Verifiers i.e. as a means to verify facts.

IV. PROBLEM STATEMENT

Search engines do fairly well when it comes to question answering and few TRUE/FALSE questions. But when we search to verify any information, it fails.

A. *Talking about facts:*

In the light of queries, I would analyze various search engines on how they respond to factual queries such as 2+2, with minor variations. Let's see how our favorite search engines do to efficiently answer the query satisfactorily.

Table I. Basic Math searches

Search Engines	***Queries***		
	2+2	***2+2=***	***106+98=***
Baidu	-	4	204
Lycos	-	-	-
Yahoo	4	4	204
Excite	-	-	-
Dogpile	-	-	-
Google	4*	4*	204*
AOL	4	4	204
Webcrawler	-	-	-
Mywebsearch	4	4	204
Search.com	-	-	-
Blekko	4	-	-
EntireWeb	-	-	-
HotBot	-	-	-
Exalead	-	-	-
FactBites	-	-	-
Ask	4	4	204
Hakia	-	-	-
Yandex	-	-	-
Soso	4*	-	-
Sogou.com	4	4	204
Youdao	4	4	204

* Appears in a calculator

B. *Observations:*

a. Search engines such as Google and Soso, display the result in a calculator as it is a calculator function, giving the user to do more arithmetic operation directly on the calculator instead of passing it as a query.

b. There are few search engines that link to other search results and display their search results. Example Astalavista and go.com uses Yahoo search results.
c. Similarly, like above, faganfinder uses Google search results and displays them.
d. Search Engines like Dogpile and Excite use a combination of search engines to display their search results, that are combined results from Google, Yahoo and Yandex
e. The MetaSearch Engines such as HotBot, Excite, Webcrawler, Dogpile perform badly in cases of “fact” queries

C. Basic math quries:

Now, keeping the basic aim of this paper, let’s move ahead. Now we further move to queries that have the answer to the question in the query itself and we expect from the search engine to reply us with a YES or a NO. Example of such a search is 3+3=6, where “3+3” is my “fact” query and 6 is my assumed answer.

Table II. Math queries

Search Engines	***Queries***		
	pi	***2+2=4***	***2+2==4***
Yahoo	-	-	-
Google	3.14159265359*	-	-
AOL	Pi=3.14159265	-	-
Mywebsearch	“The value…….end”	-	-
Ask	“value to 5 decimals 3.14159”	-	-
Sogou.com	Pi = 3.1415926535898	-	-
Youdao	-	-	-
Bing	Pi = 314159265358987931 /100000000000000000	-	TRUE
Wolfram|Alpha	31415926535898793238462643383279502884197169399..	TRUE	TRUE
DuckDuckGo	*Graphical result*	…4	TRUE
Browsys	*Graphical result*	4	4
Chacha	*Graphical result*	“Yes,.”	Agree

* Appears in a calculator

D. Factual queries:

Now when we go further, let us check how well our search engines (that performed well in the previous tests) do with simple questions, whose answers are majorly in number (with metrics), date or/and few English statements.

Table III. Basic English Statements

Search Engines	***Queries***	
	When is Thanksgiving?	***Height of statue of liberty***
Bing	Thursday, November 22, 2012 - today	305 feet 1 inches (93 meters)
Google	Thursday, November 22, 2012	305 feet (93m)
Ask	It is today…Thursday, November 22, 2012..year	The… 305 feet tall (93meters)…flame
Wolfram|Alpha	Thursday, November 22, 2012	305 feet/0.0578 miles / 102 yards / 93 meters / 0.093km
Chacha	Don’t forget…. Thursday, November 22, 2012	Your answer is 305 feet.

* Appears in a calculator

Table IV. Eiffel Tower

Search Engines	***Queries***	
	Height of Eiffel tower?	***How tall is Eiffel tower?***
Bing	1,063 feet (324 meters)	-
Google	1,063 feet (324 m)	1,063 feet (324 m)
Ask	The Eiffel tower is 324m (1,062m) tall	The Eiffel tower is 324m (1,062m) tall
Wolfram|Alpha	1063 feet *(with ranking)**	1063 feet (with ranking)

* Gives a lot of additional relevant information

E. Fact Verifier:

Some of this is factual, such as “Who is the current President of the US?” But most of the needed content is more like rules of thumb, such as “Why you should carry a glass of water open-end up? “Almost every search engine did well in answering the questions listed in the table.

Table V. String Results

Search Engines	***Queries***	
	Who is the President of US?	***Who wrote Hamlet?***
Bing	-	William Shakespeare
Google	Barack Obama, *USA, President*	William Shakespeare. *Hamlet, Author*
Ask	The Chief of State..President Barack Obama	William Shakespeare. *On a dark night…*
Wolfram|Alpha	Barack Obama	William Shakespeare

Table VI. Fact Queries

Search Engines	***Queries***	
	Is thanksgiving on November 22, 2012?	***Is Eiffel tower 324m tall?***
Bing	-	-
Google	-	-
Ask	-	-
Wolfram|Alpha	-	-
Chacha	We’re already working on that answer for you*	-

* There are many people working behind chacha to figure out the query and put the answer in database.

V. METHODOLOGY

In this section we propose a method to solve this problem and answer such question types with either a YES/NO. Now as the problem statement of this paper describes, we saw that the all the search engines had answers to all the questions in the table. Similar questions were asked before , just putting the queries in a form of a YES/NO question results in displaying the documents related to the queries, making the search engine dumb all over again. The question is if we could we do something to solve this problem.

What we propose here is, first we take the query from the user which is of type 2 and read it. Now we need to identify if it is a fact query or not i.e. type 2 or its variant. To be able to do this efficiently, we need to analyze many forms of fact question types. Most queries intended to ask facts will mostly begin with Is.......? Or Did......? Or if it’s past event was.......?.

Since we know the query is a fact type, we make a comparison between the Type 2 query and Type 1. If we closely look at the two queries,

Query: "Who is the author of Hamlet?"
Query: "Is Shakespeare the author of Hamlet"

The only difference is that we have an additional term in the query that confuses the search engines and it results in displaying the documents.

So our aim here will be to split the query to extract the value. We will do it in a manner that we will obtain two substrings. One will contain the query minus value and the other will contain the value.

A paper on Predictive Annotation analyzed its performance in the TREC8 Question-Answering track by Radev et al. Shallow pattern-matching of the textual material was used. The purpose was to augment it with QA-Tokens which are a set of named entity identifiers. These QA Tokens are indexed along with the text and they replace the question words using Question analysis. Using a very simple scoring mechanism a bag-of-word matching of short passages takes place. We project that our techniques can in principle handle 90% of the questions used in this track as against the 72% achieved in practice. [10]

Predictive annotation is a new technique for handing answers to natural language questions in text corpora. A system based on predictive annotation delivers superior results in comparison to other competing systems. [12]

We can efficiently do this by using pattern matching and having a split by choosing a separator. There can be many ways to divide the query into two sub parts. But our efficiency will depend on how good our separator is.

Let's analyze a few more such queries (the variants of the Type 2 query).

a. Is Shakespeare the author of Hamlet?
b. Did Shakespeare write Hamlet?
c. Was Hamlet written by Shakespeare?
d. Was Hamlet authored by Shakespeare?
e. Shakespeare wrote Hamlet?
f. Shakespeare authored Hamlet?
g. Shakespeare is the author of Hamlet?
h. Hamlet was written by Shakespeare?
i. Author of Hamlet is Shakespeare?

....

A. *Observations:*

Mostly all queries begin with Is/Did/Was and this was taken into account to identify the query as a fact query.

Every query ends with? But the user may choose to skip or put a full stop. We need to keep that in mind while doing pattern matching.

Now, we notice that in every query the verb separates the value and the query to a great extent.

Passive voice: Now if we look at the queries in some the first sub-part on the left contains the value and the second part contains the query. Example: Queries 1,2,5,6 and 7. On the other hand, queries 3, 4, 8 and 9 have the value in the second sub-part on the right.

Now let's delve deep into Observation 3 in the following queries, the verbs are highlighted.

a. Is Shakespeare **the** author of Hamlet?
b. Did Shakespeare **write** Hamlet?
c. Was Hamlet **written** by Shakespeare?
d. Was Hamlet **authored** by Shakespeare?
e. Shakespeare **wrote** Hamlet?
f. Shakespeare **authored** Hamlet?
g. Shakespeare is **the** author of Hamlet?
h. Hamlet was **written** by Shakespeare?
i. Author of Hamlet **is** Shakespeare?

....

Hence, if we identify the verb and choose it as our separator, it will work great!! Now, there could be many more queries, for which our separator should hold.

Similarly I could have many ways to extend these queries. For example, Hamlet could be replaced by the book Hamlet.

Is William Shakespeare the author of the book Hamlet?

....

Author of the book Hamlet is Shakespeare?

....

Also, the word "author" could be used with write with some query variants.

Summing it up mathematically, we have 3 ways (at least) of writing the author name and 2 ways (at least) of writing Hamlet and not considering any other factors we could have the following number of variants.

$$Number of variants = 9 queries * \binom{3}{1} * \binom{2}{1}$$

$$= 9 * \frac{3!}{1! * (3! - 1!)} * \frac{2!}{1! * (2! - 1!)}$$

$$= 9 * 3 * 2$$

$$= 54 queries.$$

Hence, we need to account for these as well and all variant queries will work with our separator and we will be able to extract the value from the query.

After we have decided on our separator, we get two substrings: One which contains the value and the other one contains the sub-query.

We have to now interact with the search engine, we could use many interesting techniques to decide which search engine to send our sub-query to. For example, if we break the query into two parts and apply == sign between two queries and send it to Wolfram|Alpha it returns true. Wolfram|Alpha is the only search engine of all those listed that returns TRUE for some queries.

If we simply compare the height of the Statue of Liberty (returns 93m) and compare it with a = sign and then the answer, it returns is a TRUE value. (As of November 07, 2012).

Query: "Height of Statue of Liberty==93m?"
Result: True

Else, we send the query to the search that returns the answer to the query of our sub-query.

Here are some suggestions to choose the right search engine to send our sub-query:

We could look at the various tables in this paper to see which search engines does the best in a particular query type. For example, Bing does great job with basic arithmetic operation on integers. But sometimes Bing gets the result from freebase and doesn't show it in its Web results when doing programmatically. Wolfram Alpha handles very sophisticated mathematical searches.

If the decision seems to difficult or trusting just one search engine with the query may not be a good idea. So you could send the query to all the popular search engines store the first result in a table and then check the value with all the value we extracted.

A lot will also depend on, which query we are sending to search on the search engine. Hence we need to make the query as proper as possible. In some cases, while interacting with some search engines removing of stop words yields better results and in some it fails completely. For example, when we remove the noise words that don't take part in the query and search it on WolframAlpha we get better results, but if we remove the stop words and search on Google our query results in listing the documents.

We send the optimized query to the search engine. Now, looking at observation 4, we learnt that the query can be present in any of the two subparts we get, as the user could use passive voice of the statements too. Hence we need to pass both the sub-query and store the result and check the other query with both the sub-queries check our both subparts with the result returned from the search engine.

If any of the results returned from various search engines match to the value return YES if not then return NO.

The value can also be the subset of the result returned. For example, User may only enter Shakespeare instead of William Shakespeare which will be returned by the search engine.

We successfully reach a situation wherein on entering the query in the form of a fact like "Is Condon the director of Breaking Dawn?" we get the answer as YES instead of the listing of various documents containing the keywords or returning a result which doesn't answer the question.

B. Algorithm:

Here I propose an algorithm to carry out such functionality to be able to answer the user question with an authoritative answer (YES or NO).

Algorithm 1 is the skeleton logic of our solution.

Algorithm 1: FACTVERIFY: Returns an authoritative result to a Fact Query

Input: Fact Query from the user
Output: YES/NO

```
initialization
q <— query
(query-part, value) = PatternMatching (q)
URL: search-url+query-part+api-key
Result = Parsing(URL)
if Result contains value then
  | Print YES
else
  | Print NO
```

Algorithm 2: PATTERN-MATCHING: Compares with all possible Regular Expression

Input: Query(q)
Output: query-part,value

```
if q matches regex then
  | [] = q.split(seperator)
  | [0]= value
  | [1] = query-part
```

Algorithm 3: PARSING: Sends the query-part to the Search Engine and gets the answer

Input: Query-part
Output: Answer returned by the Search Engine

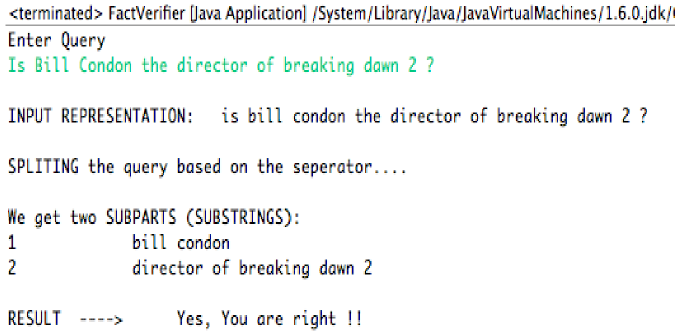
```
Establish a connection with the Search Engine
URL: search-url+query-part+api-key
content <— text of the URL
Get answer from the <tag> return answer
```

VI. IMPLEMENTATION

Based on the proposed algorithm, the code was developed and it took care of the basic functionalities for demonstration purposes and more importantly, proof of concept

A. Output:

Query: Is Bill Condon the director of breaking dawn 2?


```
<terminated> FactVerifier [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/
Enter Query
Is Bill Condon the director of breaking dawn 2 ?

INPUT REPRESENTATION:   is bill condon the director of breaking dawn 2 ?

SPLITING the query based on the seperator....

We get two SUBPARTS (SUBSTRINGS):
1          bill condon
2          director of breaking dawn 2

RESULT  ---->     Yes, You are right !!
```

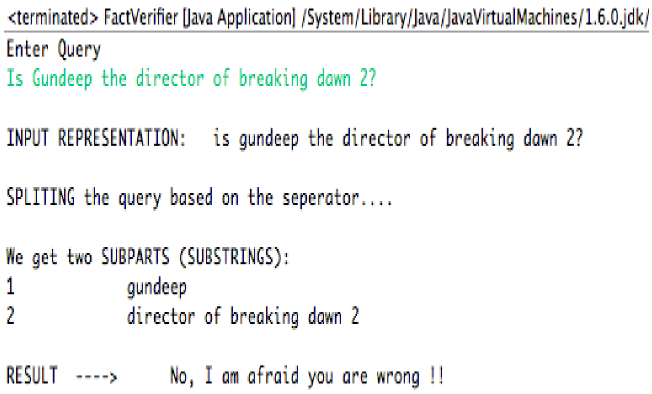
Figure 1. Screenshot running Query1

Here, the separator is the verb, so the split is on "the", hence we get two substrings i.e.

a. bill codon
b. director of breaking dawn 2

Now we send the director of Breaking Dawn 2 to Wolfram|Alpha and check the result with the value specified by the user i.e. "Bill Codon". We get a match. Hence it returns that we are right!! Now, let's check for a false case. So I test it with my name and input the following query:

Query: Is Gundeep the director of breaking dawn 2?


```
<terminated> FactVerifier [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/
Enter Query
Is Gundeep the director of breaking dawn 2?

INPUT REPRESENTATION:   is gundeep the director of breaking dawn 2?

SPLITING the query based on the seperator....

We get two SUBPARTS (SUBSTRINGS):
1          gundeep
2          director of breaking dawn 2

RESULT  ---->     No, I am afraid you are wrong !!
```

Figure 2. Screenshot running Query 2

Of course, I am not the director of the movie, so it returns NO to me. Now, in case the query is entered by the user that doesn't make any sense.

Query: "Is tall Eiffel Tower 324m?".

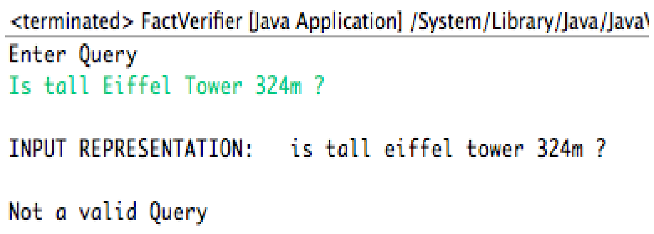


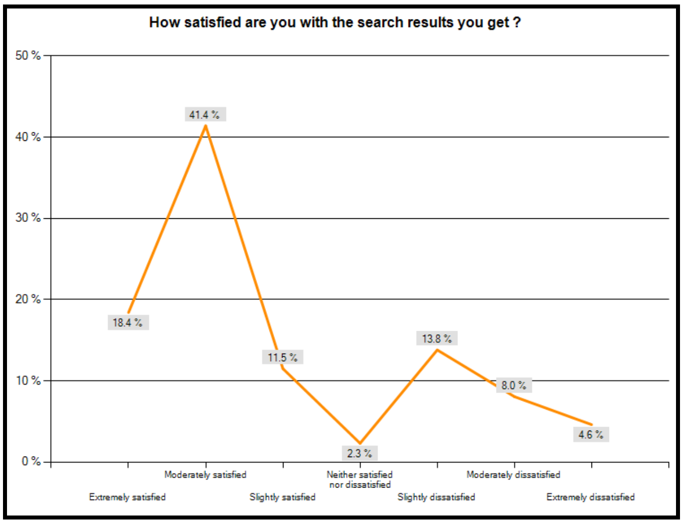
Figure 3. Screenshot running Query 3

Then it returns "Not a valid query". In that case, we could revert back to our keyword based searching and return the list of documents to the user containing these keywords, as it not a valid question.

We will also present the integration of our technology into commercial search engines such as Google, Bing and Yahoo! as a framework that can be used to integrate our model into any other commercial search engines, or development of the next generation of search engines.

VII. SURVEY

A survey was conducted [6] with a sample size of 1000, after successful implementation of FactVerify. The users were asked for their feedback. This gave a fair idea of the success and whether it would be of great value for this feature to be in real search engines.

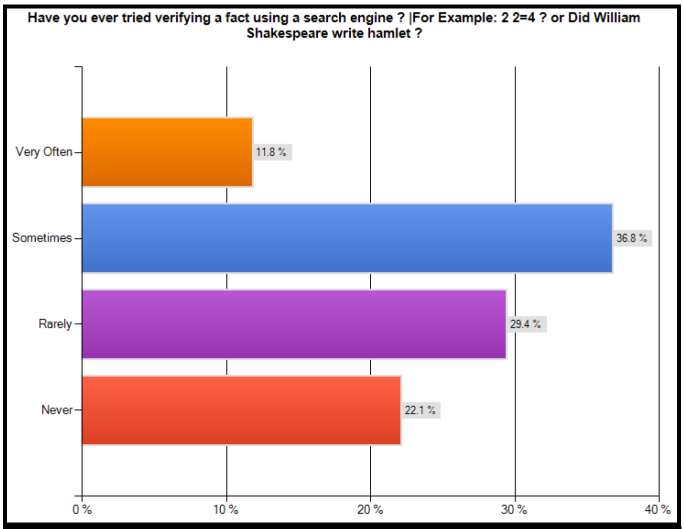


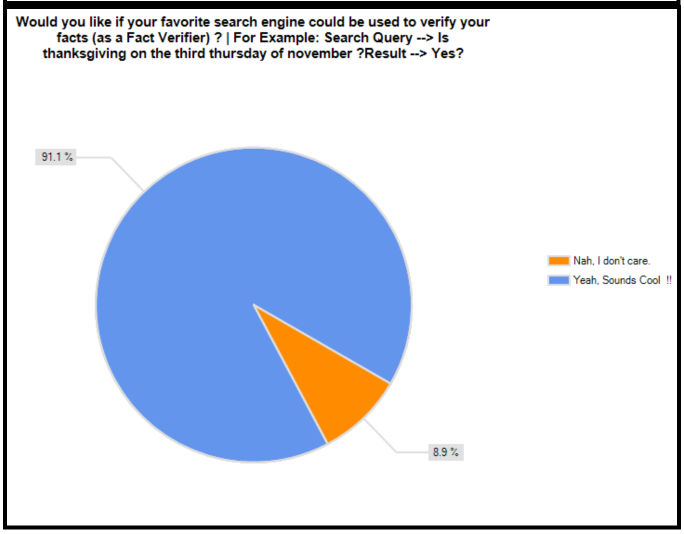
A. Observations:

a. Users were in generally, happy with the results, but not much satisfied. Moderately Satisfied and Slightly Satisfied together took about half of the total.

b. Nearly 1/3 of the users were not very pleased with the results they got from their favorite search engine.

c. About Every 2 out 10 users were extremely satisfied with the search results they get from their favorite search engine.





B. Observations:

a. A very central split was seen. There were half users who used fact queries very often or have done it few times in the past. On the other hand there were the second half of the users who have done it rarely or never.

b. There are chances that users that have searched fact queries rarely or never is probably due to the nature of search results users get.

c. This feature was loved by the users who used the FactVerify and loved it. 91.1% of the users found it really useful, i.e. nearly about 9/10 of the users liked FactVerify.

VIII.CONCLUSION

Our work is related with fact-based queries in information retrieval, where the answers to the query are authoritative in nature to produce a resulting YES/NO hit. In other words, current search engines can be used as a means to verify facts. FactVerify was produced with such functionalities and explicit user feedback was taken by conducting an online survey. The results suggested that 88.9 percent of users liked FactVerify and found it useful.

Query: Could I use my favorite Search Engine to verify my facts?

Result: Yes

To improve the question type through pattern matching, developments for more complicated query matching schemes, with choosing efficient separators would be needed to be developed. FactVerifier accurately determined the query type, is .... <answer>........ ? And gave successful results in all cases without fail. It also suggests plausibility of exploring the possibility of adding more and more queries with the suggested proposed technique.

This paper takes a step towards being able to get the best answer from the search engine, modify and return results for maximum user's satisfaction.

IX. FUTURE WORK

As most services of this kind, Google provides an incentive system to motivate people to answer questions. It is based on assigning points for actions and a system of levels loosely based on the Russian system of academic degrees. An interesting feature of the incentive system is that Google's reward for visiting is higher than for posting an answer. More languages were added, including English. The site is known as Google Baraza,[7].

These principles and algorithm proposed in this paper are general, and can be applied to virtually any type of textual media. The further work can include how to make this method better. Further, new models can dig into complex pattern matching capabilities to many variants of fact queries.

Variants:
Is thanksgiving on the 22nd this year?
Is thanksgiving on the 4th Thursday?
Is thanksgiving on the third Thursday of November?

Work can be done to make semantic analysis in question processing module in order to understand the user's intentions fully and extract the more accurate answers. Researchers are working on developing search engines with growing complexity and technological challenges.

X. ACKNOWLEDGMENT

At the outset, in all humility, we thank the Almighty for the plentiful blessings He has showered on us to undertake and see through to completion this paper. It would not be out of place to thank all the authors and researchers whose work we have consulted. Our thanks are due to the founders of various softwares (LaTeX, PDF, TeXMaker etc...) that I have used in writing this paper.

XI. REFERENCES

[1] D. Farber. Wolfram alpha: Next major search breakthrough? March 8 2009.

[2] J. Grimes. The thread of discourse, volume 207. Walter de Gruyter, 1975.

[3] J. Jeon, W. B. Croft, and J. H. Lee. Finding similar questions in large question and answer archives. In Proceedings of the 14th ACM international conference on Information and knowledge management, CIKM '05, pages 84–90, New York, NY, USA, 2005. ACM.

[4] X. JI, X. Wang, and A. LI. Intonation patterns of yes-no questions for chinese efl learners. Proc. Of O-COCOSDA2009, Beijing, 2009.

[5] B. Johnson. British search engine 'could rival google'. The Guardian, March 2009.

[6] L. K. Lawless. English closed questions yes/no questions. 2012.

[7] F. Li, H. Kang, Y. Zhang, and W. Su. Study on question-answering system based on meta search engine. In Computer Application and System Modeling (ICCASM), 2010 International Conference on, volume 3, pages V3–354. IEEE, 2010.

[8] M. Liu, Y. Liu, and Q. Yang. Searching semantically similar questions from a large community-based question archive. In Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009. International Conference on, pages 1–8. IEEE, 2009.

[9] M. Nikravesh. Neuroscience and precisiated natural language-based search and question answering system: Neusearch. In Granular Computing, 2005 IEEE International Conference on, volume 1, pages 45–50. IEEE, 2005.

[10] J. Prager, J. Chu-Carroll, E. Brown, and K. Czuba. Question answering by predictive annotation. Advances in Open Domain Question Answering, pages 307–347, 2006.

[11] D. Radev, W. Fan, H. Qi, H. Wu, and A. Grewal. Probabilistic question answering on the web. Journal of the American Society for Information Science and Technology, 56(6):571–583, 2005.

[12] D. Radev, J. Prager, and V. Samn. Ranking suspected answers to natural language questions using predictive annotation. In Proceedings of the sixth conference on Applied natural language processing, pages 150–157. Association for Computational Linguistics, 2000.

[13] D. Radev, H. Qi, Z. Zheng, S. Blair-Goldensohn, Z. Zhang, W. Fan, and J. Prager. Mining the web for answers to natural language questions. In Proceedings of the tenth international conference on Information and knowledge management, pages 143–150. ACM, 2001.

[14] R. Soricut and E. Brill. Automatic question answering using the web: Beyond the factoid. Inf. Retr., 9(2):191–206, Mar. 2006.

[15] W. Team. Wolfram|alpha and duckduckgo partner on api binding and search integration. April 18, 2011.

[16] L. A. Zadeh. From search engines to question-answering systems: the need for new tools. In Proceedings of the 1st international Atlantic web intelligence conference on Advances in web intelligence, AWIC'03, pages 15–17, Berlin, Heidelberg, 2003. Springer-Verlag.

