Using Semantics to Search Answers for Unanswered Questions in Q&A Forums

Priyanka Singh
University of Southampton
WAIS, ECS
Southampton, UK
ps1w07@ecs.soton.ac.uk

Dr. Elena Simperl
University of Southampton
WAIS, ECS
Southampton, UK
E.Simperl@ecs.soton.ac.uk

ABSTRACT

The expert based question and answering forums are crowdsourced and rely on people to provide answers for questions. This paper focuses on technology based Q&A systems like StackOverflow and Reddit. These websites are popular and yet many questions remain unanswered. The Suman system uses semantic keyword search in combination with traditional text search techniques to find similar questions with answers for unanswered questions. Furthermore, the Suman system also recommends experts who can answer those questions. This helps to narrow down the long tail of unanswered questions. The Suman system utilises Semantic Web and Linked Data technologies to integrate the datasets from two websites, structure them and link them to Linked Data Cloud. It uses available tools to solve name entity disambiguation problem and expands the query term with added semantics. The Suman system was evaluated and results were analysed to show its viability.

Keywords

Semantic search, Semantic Web, Q&A Forums, Search Applications, Linked Data, Keyword search, StackOverflow, Reddit

1. INTRODUCTION

The Web has provided a distributed platform for people to collaborate and leverage collective intelligence for distributed problem solving. Messaging boards, Q&A forums are some examples where people broadcast problems and experts provide solutions. These communities help to create an emerging knowledge.

This paper focuses on technology based Q&A forums like StackOverflow [1] and Reddit [2]. Here users ask questions and experts in the field provide solutions using crowdsourcing techniques. These websites are popular among software programmers to ask questions and have discussions. StackOverflow have more than 3.2 million questions and 1.2 million registered users. Despite so many users, 23.7% of questions in StackOverflow do not get any answers [33]. There is a long tail of questions that get no answer or votes.

The Suman system, presented in this paper, find answers for unanswered questions. *Suman* is a Sanskrit word meaning wise and good mind. The system combines keywords based semantic search with traditional text based search to find answers for unanswered questions from StackOverflow and Reddit. The algorithms performs SPARQL [27] queries and utilises the crowdsourced data (votes) to rank the results. The Suman system also search for experts and recommend them to provide answers to the unanswered questions.

StackOverflow and Reddit do not have common data structures and schema. It is difficult to integrate data, align their schema and link them with other datasets. The Suman system uses Semantic Web [2] and Linked Data [4] technologies to solve this problem. It structures the data into RDF [20], align their schema and link it to the Linked Data Cloud. The Suman system uses Wikipedia-Miner [25] and OpenCalais [28] for name entity recognition and annotate the data with keywords. This added semantics helps to identify the topics in each post that improves categorization and indexing.

Another issues in these forums are finding right answers and experts for questions. A given search engine can retrieve information when explicitly asked. It does not return the solution of a problem if the solution does not exist on a webpage. Search engines use the keywords in the search query to retrieve results. They do not expand the query to broader, narrower or similar fields. People searching for answers or experts in a forum can only see results from their own network, while losing a whole community of experts in other forums. Integrating the dataset of multiple forums and added semantics provide solution for these issues too.

The Suman system was evaluated by users in two experiments and the results were statistically analysed. It showed that the keywords generated by the Suman system were rated higher than the original keywords from the website. The analysis also showed that the participants agreed with the algorithm rating for answers provided by the Suman system.

2. THE SUMAN SYSTEM

The Suman system was designed as a proof of concept to show Linked Data and Semantic search techniques can be used in crowdsourced Q&A forums. The system uses Semantic Web technologies to integrate different datasets. It

[1] http://stackoverflow.com/

[2] http://reddit.com/

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2890569.

takes heterogeneous data and converts them into RDF. It uses vocabularies like FOAF [7] and SIOC, [5] and uses its own schema to align the datasets. It adds semantics by doing name entity disambiguation and other related keywords and categories. Furthermore, it creates a user model based on their activity (question, answer and vote) and recommends experts who can answer the unanswered questions. The Suman system takes advantage of semantics and crowdsourced information to improve search.

2.1 Search Algorithm

This section discusses the Suman search algorithm. The notations used are as follows.

K = Keywords; D = Documents; Q = Query; V = Vote; S = Score $S \in \mathbb{R} : 0 \leq S \leq 10$

Here, questions and answers are referred as documents D. Each document has a set of keywords K associated with it. The Suman system search for answers for an unanswered question. Hence, a query Q consists of an unanswered question and the keywords associated with it. Vote V is the vote given to questions and answers by the users in the forums. Score S is the score given to the search result by the algorithm based on its validity and importance.

Algorithm 1 Suman Search Algorithm

```
$D = [], K = [], minScore = 0.7$
$[\bar{q}, K] \leftarrow FindQuestion(Random)$
$Q = k_1 \wedge k_2 .... \wedge k_n \forall k_i \in K$
$[D, S] \leftarrow FindDocs(Q)$
$EK \leftarrow Expand(K)$
$\bar{Q} \leftarrow ek_1 \vee ek_2 .... \vee ek_n \forall ek_i \in EK$
$[ED, ES] \leftarrow UpdateDoc(D, \bar{Q}, minScore)$
$[ED, S^{\bar{Q}}] \leftarrow TextSearch(\bar{q}, ED)$
$[ED, S] \leftarrow UpdateScore(ED, ES, S^{\bar{q}})$
$[ED, V] \leftarrow GetVote(ED)$
$[D^{Final}] \leftarrow ScaleScore(ED, S, V)$
$[D^{*Final}] \leftarrow GetContext(D)$
```

Detailed Explanation of the Suman Search Algorithm:

1. D=[ ],K = [ ], minScore = 0.7 = D is an empty list that stores all the documents returned after running the query. This is referred as cache in this section. k=[ ] is an empty list that stores all the keywords associated with the document. The minScore variable stores the value of threshold score. In this algorithm it is 0.7. Any documents with scores lower than the minScore is pruned to maintain the quality of the search result.

2. FindQuestion(Document) = This operation returns a randomly selected unanswered question $\bar{q}$ from the database. This question has a set of keywords K associated with it and they are also retrieved.

3. Query(Q) = The first query is automatically created by the Suman system using all the keywords associated with the unanswered question. The keywords are joined using the $\wedge$ (AND) operator to make sure it consist of all keywords.

4. FindDocs(KeywordSet) = This operation takes the sets of keywords K related to the query question Q. It uses the set of keywords to run a SPARQL query and search for documents that contain all the keywords. The SPARQL query is created automatically by the Suman system. This operation creates a cache [D, S] that holds all the returned results with a score associated to each result. The query gives higher score to documents that have all the keywords and multiple occurrence of the keywords associated to it. Any documents with score less that minScore are pruned.

5. Expand(KeywordSet) = This operation takes a set of keywords K associated with the query question Q and returns an *expanded* set of keywords. It takes each keyword and finds the parent keyword if it has one. Each keyword has been disambiguated and has a broader and narrower term associated with it. The broader terms are the parent keyword and the narrower terms are the children keywords. If the keyword has a parent keyword associated with it then that keyword is added to the keyword set. If there are no parent then children terms are searched and added to the keyword set if they exist. This generates an expanded set of keywords EK.

6. UpdatedQuery($\bar{Q}$) = The new query $\bar{Q}$ is updated by adding all the expanded sets of keywords joined using the $\vee$ (OR) operator.

7. UpdateDoc(DocumentSet, Query, minScore) = This operation takes a set of documents D, a query that consists of the set of expanded keywords $\bar{Q}$, and a score called minScore. If the minScore is greater than 0.7 and if the document already exists in the cache then the score is updated to give it a boost. Otherwise the documents are added to the cache with its score.

8. TextSearch(Query, DocSet) = This operation takes the text of the questions $\bar{q}$ and performs a text search using the Lucene search engine in the cache dataset. This search is not done on the whole database but the cached dataset that consist of all the documents related to the keywords. This operation returns a new score for the documents in the cache.

9. UpdateScore(DocumentSet, KeywordScore, TextSearch-Score) = The cached documents have two scores. The first score was given after the keyword query results and the second score was given after the text search results. This operation updates the score by taking average of the both scores. This is done to normalize the score and keep it within the range of 0.7 to 10.

10. GetVote(DocumentSet) = This operation gets the votes received to each documents in the cache.

11. ScaleScore(DocumentSet, Score, Votes) = This operation takes the vote V of the documents and scale it to the same range as score S. The scores are usually between 0.7 and 10 and votes are $\mathbb{Z}$. The votes are normalized to be $V^* \in \mathbb{R} : 0 \leq V^* \leq 10$. This is done by unity based normalization and the value is restricted in the range between *a* and *b*. Here, the range of a and b are same as the range of score S.

    $\bar{V}$ = a + $\frac{(V-V_{min})(b-a)}{V_{max}-V_{min}}$

    For example, if the cache had three documents with votes 10, 15 and 20 then unity based normalization turns their score into 0.7, 5.35 and 10 respectively to keep the score within the range of 0.7 and 10. The Suman algorithm does this with all the documents present in the cache.

If the votes were negative then the normalized vote value is deducted from score and if they were positive, normalized vote value is added to the score. Then the average is taken of the score and the normalized value of votes. Any documents with final score less than 0.7 are pruned from the list. This modifies the final ranking of the documents and the new sorted list of the documents is presented as the final result.

12. GetContext(DocumentSet) = This operation retrieves the parent posts of document, if available. The questions do not have parent posts but answers and comments have parent posts associated with them. All the parent posts of answers and comments are retrieved to provide context to the final result.

The final result consists of a ranked list of documents that could potentially answer the unanswered questions. The top 10 results are shown as the final result consisting of answers with the parent posts.

This search algorithm uses keyword based semantic search and combines it with the text based search. The final ranking of the search result is modified using the crowdsourced votes given to questions and answers by the community.

2.1.1 Expert Finder

The questions, answers and keyword graph is extended to users. Users are linked with the keywords and votes to create a user keyword graph. Every document (questions, answers, posts, comments) has a user associated with it. This helps in joining the keywords with the users. Also, keywords have categories related with them, it joins the user graph with related categories. This helps in recommending experts.

The Suman expert recommendation algorithm is discussed below. The notations used are as follows: K = set of keywords, E = set of experts, Q = Query, R = Reputation points of experts, S = Score $S \in \mathbb{R} : 0 \leq S \leq 10$

Algorithm 2 Suman Expert Recommendation Algorithm

```
1: $E = [], K = [], minScore = 0.7$
2: $[\bar{q}, K] \leftarrow FindQuestion(Random)$
3: $Q = k_1 \wedge k_2 .... \wedge k_n \forall k_i \in K$
4: $[E, S] \leftarrow FindExperts(Q)$
5: $EK \leftarrow Expand(K)$
6: $\bar{Q} \leftarrow ek_1 \vee ek_2 .... \vee ek_n \forall ek_i \in EK$
7: $[EE, ES] \leftarrow UpdateExperts(E, \bar{Q}, minScore)$
8: $[EE, R] \leftarrow GetReputation(EE)$
9: $[E^{Final}] \leftarrow ScaleScore(EE, S, R)$
10: $[D^{*Final}] \leftarrow GetDetail(E)$
```

Search of experts are done similarly to the answers. Step 1 to 7 are similar to the answer algorithm. The keywords of the question are matched with the users' keywords. The results are scored between $Score \in \mathbb{R} : 0 \leq Score \leq 10$. If there are no users with score greater than 0.7 then the keywords are expanded to include the categories. This expands the keywords to include broader and narrower terms. Any users with score greater than 0.7 is returned and the list is updated. Step 8 of the expert recommendation algorithm extracts the user's reputation points. Then the reputation points are normalized to be $R \in \mathbb{R} : 0 \leq R \leq 10$. Average of user's score and normalized value of reputation is taken and the final list is sorted and presented with users' details.

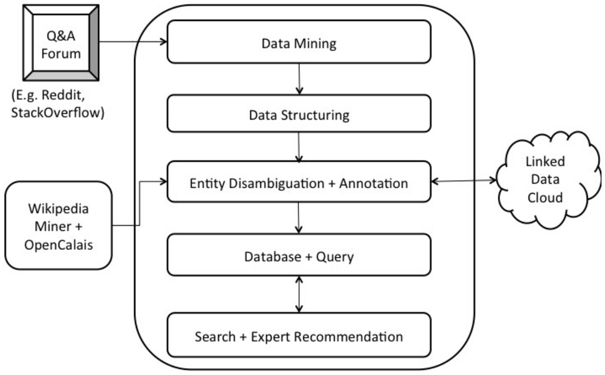

Q&A Forum
(E.g. Reddit, StackOverflow)
Wikipedia Miner + OpenCalais
Data Mining
Data Structuring
Entity Disambiguation + Annotation
Database + Query
Search + Expert Recommendation
Linked Data Cloud

Figure 1: The Suman system design

The final result consists of a ranked list of experts that could potentially answer the unanswered questions. The top ten results are shown as a query results. Search of experts are done similarly to the answers. The list of experts are best matched by the keywords of the question so they are assumed to be experts in those topics. The experts have additional information associated with them like their location, latest activity, posting history, etc. This can be used to modify the query result to find the experts in the same time zone, or experts who were recently active and post on the website regularly. This would potentially help in finding the right experts for the right needs.

2.2 Building the application

StackOverflow and Reddit websites were chosen to collect the data and test the Suman system. The system can be used with any Q&A systems but for this research these two websites were chosen because of their easy to use APIs.

The Suman system is divided into five main parts- data mining, data structuring, annotation and linking, database and query and finally searching and expert recommendation. Figure 1. gives a framework diagram of different components and how the system was designed. Each component is discussed in more details below.

2.2.1 Data Mining

StackOverflow provides a regular data dump of all their public data. The dump has data about questions, answers, comments, user information (public data only), badges and votes. The data is in XML format and the files are shared using P2P torrent. Data dump till August 2014 was downloaded for the Suman system. The API was used to get lists of tags, total number of questions in each tags, tag synonyms and related tags. The API returns JSON file and file was parsed and stored in the database.

Reddit has an API that allows to get posts from particular subreddit. For Suman system 11 programming related subreddits were chosen that corresponded to top 10 tags of StackOverflow. These subreddits were – Java, PHP,

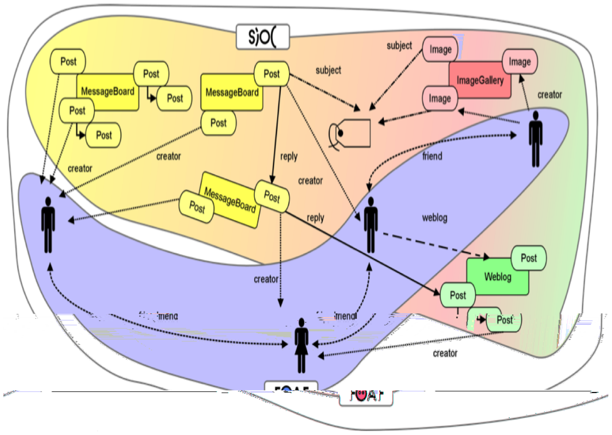


Figure 2: Data structured using FOAF and SIOC vocabularies

Python, Javascript, Ruby, C++, C#, Perl, Programming, Learnprogramming and Webdev. PRAW (Python Reddit API Wrapper) [3] library was used to get the posts, comments, votes, users, flairs, etc. information from every subreddit. The JSON file was parsed by the library and stored in the database. The main limitation of the Reddit API was that it did not give information after 200 pages and each page contained only 25 posts. So every subreddit only provided limited number of posts, it did not provide complete dataset in a particular subreddit like StackOverflow. Hence, the Reddit dataset and user profile was incomplete.

2.2.2 *Data Structuring*

All the data was encoded in Unicode and stored in the MySQL database. The StackOverflow dataset consisted of more than 15 million questions, 28 million answers, 43 million votes and 1.5 million users. To make the dataset manageable and still keeping it complete within the community, top 10 tags were chosen. Reddit data consisted of more than 19 thousand posts, 0.41 million comments, 4.6 million votes and 71 thousands users.

For the Suman system, all the data was converted into RDF. RDF is a data structure format to describe the data and its relationship with the URI. The subject-relation-object model is also called triples [20]. To describe the data in RDF, vocabulary is used to define the relationship. FOAF ontology is used to describe the users and their profile information [7]. Both StackOverflow and Reddit only show basic user profile information due to privacy reasons and FOAF ontology is used to describe the data. Similarly, the posts created by users, questions and answers are described using SIOC and DC ontology [5].

There are limitation to FOAF and SIOC ontology. They do not have terms to describe data like votes, favourites, flairs, etc. In that case, RDF schema (RDFS) was defined to describe the properties like votes, favourite counts, flairs given to the questions, answers and comments.

The main issue of this stage was to give both StackOverflow and Reddit data a common structure. The datasets mined from these websites were different. StackOverflow data consisted of questions and answers. Reddit data consisted of posts and comments. The Reddit posts were of two types - text posts that were questions or information, and link posts that linked to external source.

The issue was resolved by considering every questions, answers and comments as posts. These posts were of two types to differentiate their main role. Type 1 was for StackOverflow questions and Reddit main posts. Type 2 was for StackOverflow answers and Reddit comments (parent and children comments). Each post had a parent post associated with it. The Type 1 post had no parent posts. Type 2 posts that consisted of answers and main comments had their corresponding parent posts (questions in StackOverflow and posts in Reddit) as parents. The children comment posts in Reddit had the parent comment post as their parent post to maintain the thread structure of the conversation.

2.2.3 *Annotation and Linking*

One of the main benefits of using Semantic Web and Linked Data technologies are adding semantics to the data and linking it to other datasets. Adding semantics helps in providing the context and domain specific meaning to the data. Linking the data helps in using the links and relationships to find related information about any resource [4].

The main problem in adding semantics to the data is name entity disambiguation. E.g., it is necessary to know if 'Java' mentioned in a post is about Java programming language or the Indonesian island Java. Once name and entities are resolved, the context of the domain could be used to add semantics, categories and other relevant information. This step adds keywords in the domain of technology to the collected datasets. It also adds categories to the keywords to link them and form semantic relationships between them.

StackOverflow dataset is sparsely annotated by user-generated tags and it is not linked with any other datasets. The tags help to categorize the questions into different topics and show it on different tags page and notify users subscribed to that tag. The answers on the other hand do not have any tags. They inherit the tags from the questions. During data collection, question tag was added to the answers by the data mining script. Reddit dataset has no tags associated with posts. During data mining process the name of the subreddit was added as the tag to both posts and comments. This worked for language specific subreddits like Python and PHP but did not work for general programming subreddit like 'learnprogramming'.

The questions, answers, comments and tags data were annotated with the links from DBpedia and OpenCalais datasets to resolve the name disambiguation. DBpedia knowledge base describes 4.58 million things [1]. OpenCalais is Thompson Reuters initiative that tags keywords, topics, etc. [28]. Wikipedia-miner [25] and OpenCalais [10] are used to do name entity recognition and match it to a known vocabulary and taxonomy. [24] [17] states that adding semantics from different data sources improves the quality of the metadata and semantic context significantly. It also overcomes any false match done by one application. Hence, both OpenCalais and DBpedia dataset were used to resolve the name entity issue and link the data. Both tools do natural language processing of text and annotate with keywords. This annotation is then matched with the Wikipedia topics and OpenCalais entities. By using the links to the matched topics StackOverflow and Reddit data was linked to the DBpedia and OpenCalais dataset in Linked Data Cloud.

[3] https://github.com/praw-dev/praw

Next, Wikipedia categories were extracted with the keywords. Every keyword is linked to its parent and child categories. This will later help in expanding keywords with other related keywords in the same category.

2.2.4 *Database and Indexing*

More than 45 million RDF was stored in the Stardog database [19]. Stardog is a semantic graph database. It supports RDF graph data model and SPARQL query language. It supports OWL 2 and user defined rules for inference, reasoning and constraints. It uses HTTP protocol and provides with SPARQL endpoint for applications to perform queries. The next step in Suman system was to optimize the database index to improve the search of answers and experts.

Stardog indexing system is configurable. It has RDF aware semantic search functions. It indexes RDF literals and creates a search document per RDF literal. The database is customized by adding the keyword-categories graph to the 'commongram' analyzer. That constructs n-grams for frequently occurring keywords. The n-gram is a contiguous sequence of n keywords for each post. This is also extended to the categories .

Stardog uses Lucene text analyzer to index the database to perform text based search and this analyzer is customizable. It follows the same principle as Latent Semantic Analysis [12] and tf-idf Weighting [31]. There is bidirectional relationship between a document (questions, answer, comments) and keywords. The frequency of occurrence of each keyword in a document is calculated and total frequency of documents for each term is calculated. This helps to determine the importance of a keyword in each document and also minimises the query run time when searching for documents for particular keywords or set of keywords. The final search is done using the Suman search algorithm stated earlier.

3. SYSTEM EVALUATION AND RESULT

The Suman system was tested using the unanswered questions from StackOverflow from the month of July 2014. There were 20,326 unanswered questions in the top 10 tags and the system searched for relevant answers with confidence score more than 75% for 13,209 questions. 23.62% of unanswered questions had one or more answers with confidence score of 85% and 82.27% of unanswered question has confidence score of more than 50%.

To test the viability of answers it was evaluated by users. The Suman system algorithm generates following information - a) Keywords with degree of confidence. b) Answers with rank and score. c) Experts with rank and score. These ratings were verified by humans to make sure it's correct and does not have too many errors and contextual inaccuracies.

3.1 Experiment Design

Two experiments were designed to test the quality of the keywords and the answers. Expert generator was not evaluated because: a) The Suman system returns a list of recommended experts and the algorithm score. It does not provide with any other information. b) To test the recommended expert list, participants need to know the complete user profile of the experts in the list. c) We cannot provide complete profile of the experts to the participants (due to size and time constraints) and d) Participants cannot accurately judge the expertise level of an expert by seeing their name and algorithm score.

For the keyword experiment, T-Test was chosen and ***26 random questions*** out of all tested questions were used to get the proper sample size to represent the whole population of keywords in the system. Similarly, for answers experiment, Correlation test was chosen and ***46 answers*** were tested by users to get the proper sample size. Calculating the population sample resulted in 20 participants. They are needed to get the proper sample size to evaluate the Suman system.

Java and Python programming language were chosen as these were the popular programming languages among the participants and both StackOverflow and Reddit. Since the participants needed to answer programming questions, they needed to be competent in programming. So, all the participants had two or more years of experience with the given programming language.

The experiments were done online and consisted of different topics to get wide variety of subject. The answers experiment was designed to see the correlation between the algorithms ratings and users ratings. So, a wide range of answers was selected of different quality and they were- good, medium and bad.

For the keyword evaluation experiment, participants were shown a question with the original keywords used in the website. They were asked to rate how well the keywords describe the question. Then the participants were shown the same question with the top 10 keywords generated by the Suman system. Again, the participants were asked to rate how well the new sets of keywords describe the question. The keywords experiments use the rating from 1 to 5 where 1 was for very bad and 5 was for very good. 30 questions were tested to evaluate the quality of keywords and each question receive 20 responses.

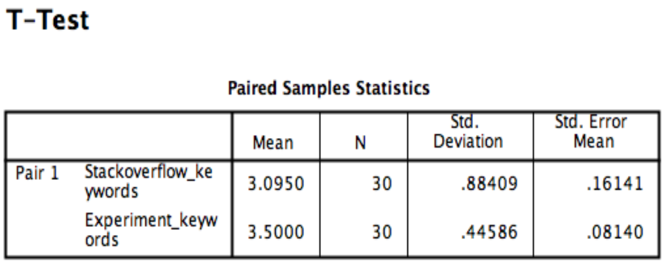
For the answer evaluation experiment, participants were shown a) an unanswered question, b) a similar question to question (a) and c) answer to the question (b). This answer (c) was expected to provide a solution for the unanswered question (a). The participants were asked to rate the quality of the answer based on how well the answer (c) could give a solution to the unanswered question (a). For the answers' evaluation the scale of 1 to 10 was used where 1 was very bad and 10 was very good. 46 questions and answers were tested. All the questions had 20 responses from the participants.

3.2 Keywords T-Test

Comparing the means of the ratings for the two depended pair could provide the usefulness of a certain set of keywords. This could be calculated using a dependent T-Test. So, this particular statistical test was done to figure out if the generated sets of keywords were adding more value to the questions then the original keywords.

The analysis showed that on average the keywords generated by the Suman system were useful and described the question better (Mean = 3.5, Standard Deviation = 0.44, Standard Error = 0.81) than the original keywords (Mean = 3.09, Standard Deviation = 0.88, Standard Error = 0.16). There was a significant difference in the usefulness of system generated keywords than the original keywords. T(28) = -2.254, p = 0.32, r = 0.38. Here p <0.5 and the effect size r = 0.38 which is medium. The system generated keywords add some benefit to the questions and answers as they improve the categorization of topics.

The data showed that participants rated the system generated keywords better than the original in 63.3% of the

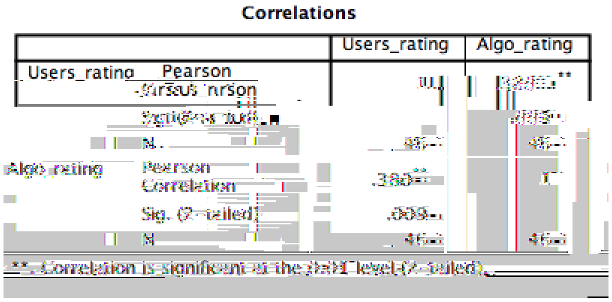


T-Test

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	Stackoverflow_keywords	3.0950	30	.88409	.16141
	Experiment_keywords	3.5000	30	.44586	.08140

Figure 3: Keywords T-Test



Correlations

		Users_rating	Algo_rating
Users_rating	Pearson Correlation	1	.380**
	Sig. (2-tailed)		.009
	N	46	46
Algo_rating	Pearson Correlation	.380**	1
	Sig. (2-tailed)	.009	
	N	46	46

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 4: Answers correlation test

cases and worse in 26.6% of the cases. A quick glance at the lower rated generated keywords shows that limitation of the system. The main drawback of the system was that it generates the keywords and links it to the DBpedia and OpenCalais dataset. If the topic does not exist in the DBpedia and OpenCalais then they keywords is not linked to it and completely ignored. Limitation of those external systems is the limitation of this system.

The other drawback of the system is that the data collected is a technical data. In these datasets lots of misspelling, abbreviation and initials are used. These colloquial are easy to understand for programmers but the keywords generator find it difficult to interpret and link. Also, in some cases the keywords are linked to the wrong topic. Like in one example the keyword 'Eclipse' was linked to the natural phenomenon of 'Eclipse' not the 'Eclipse (software)' topic. Also, the versions of software and programming languages and topics like Python 2.7, Python 2.7.3, etc, is not individual pages or topics, they are the section and subsection of bigger topics. These are harder to link to the Linked Data Cloud.

Overall, the system generated keywords performed better than the original keywords and provided additional information in regards to the questions but they have some limitation and drawback.

3.3 Q&A Correlation Test

Pearson correlation test was chosen for analysis because the data values were at regular intervals and there is a linear relationship between the two variables (algorithm's rating and participants' rating). Pearson correlation test was performed to measure the relationship between the participants rating and the Suman algorithm rating. There was a positive correlation between the two variables (r = .380, n = 46, p (two-tailed) = .009). The correlation between the two ratings is moderately strong and the significance is <.01.

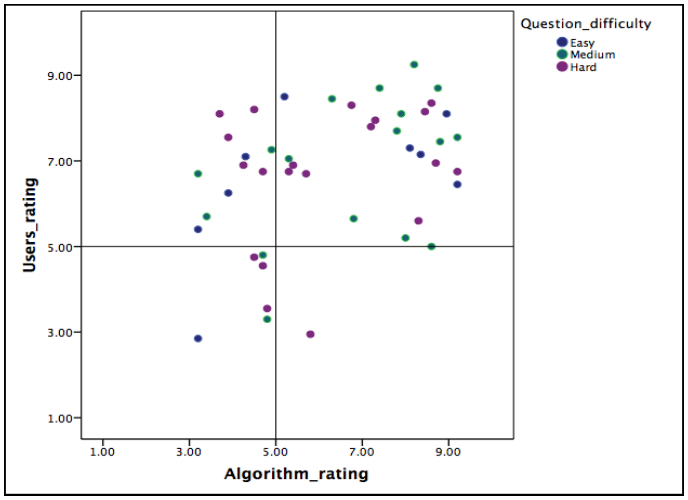


Figure 5: Q&A data scatter plot diagram showing questions' difficulty.

The analysis shows the algorithm is quite efficient in finding the right answers. The lower left quadrant in Fig 5. shows all the questions that got low ratings from the Suman algorithm as well as from the participants. The upper right quadrant shows all the questions that received high ratings from the Suman algorithm as well as the participants. The analysis of the data at the upper left quadrant shows that there are some answers that the participants gave high ratings but the algorithm did not. Fig. 5 also shows that they are mostly difficult questions.

A quick glance at the data shows that the participants gave high ratings to some answers but the algorithm did not. This could be because these questions were quite difficult and might be out of scope of the participants. They were merely guessing the answers and the answers looked valid.

There is one case where the participants gave low rating to an answer but the algorithm gave it a high score. Looking at the question and answer it is evident that the question asked for a solution for a problem that did not exist. The answer said so, and the algorithm rated the answer high because it provided enough information. The participant might have thought the failure of not providing an answer for the question that has no solution as a failure of the system.

The algorithm performed well in the user evaluation but still there is certain limitation of the system. The system uses keywords and categories to first find the subset of possible answers and then performs the text search. The limitation of the keyword annotator is the limitation of the search algorithm. The algorithm does not perform full text search again to the remaining answers in the database. Also, the system used crowdsourced data and votes to rank the answers, so it is highly depended on people's contribution. If there are malicious users or not enough votes then the algorithm is like a text search algorithm.

Overall, the answers performed well and provided relevant solutions to the unanswered questions.

4. BACKGROUND

The strength of Semantic Web and Linked Data is in searching the related information based on different categories and concepts. Semantic search uses the contextual meaning and relationships of the keywords for information retrieval. Semantic search is the combination of the conventional Information Retrieval (IR), web search and knowledge management methodologies. Semantic queries enables the retrieval of derived information based on semantic and structural information contained in the data [18].

The traditional IR methodologies are based on occurrence of words in the documents and returns a list of relevant documents with those keywords with different degree of relevancy. Term frequency-inverse document frequency (tf-idf) is a widely used syntactic measure to determine the importance of a word based on the number of occurrences in a document [31]. Some of the classic and widely used IR models are Vector-Space model [31], Probabilistic model [13], Latent Semantic Indexing [12], Machine Learning based models [32], etc. Many search systems uses some form or combination of these models [21].

Search engines augmented the existing IR methodologies with the hyperlinks and started to rank the search results using the PageRank [22], HITS [9], etc. algorithms. Some of the popular models used in the web page retrieval include the combination of content-based approach and link analysis methods. The content-based approach uses the IR methods to analyses the content of the web pages to find the best matches to the search query [14].

The text-based search provides results for the exact keywords and phrases. This sometimes does not provide the search result for users who do not know what exactly they are looking for. This limits the results in the research based queries [18]. The availability of well structured machine understandable information offers opportunities to improve the traditional search methods. Semantic based IR focus on understanding the meaning of the document instead of frequency of words in the documents. It utilises domain knowledge and ontology navigation to modify the query and apply context model to search [8].

The early research in Semantic Web added meanings and structure to the text using ontological approach or by finding similarities between the words. [24] proposed a method to find similarities between keywords by using WordNet thesaurus. [15] used ontology navigation to expand query. Another widely used approach is linking the keywords to concepts and categories [3]. This provides with broader and narrower terms to query and helps to find better search results by exploiting their relationships [34]. [24] and [26] used the semantic query expansion approach by using the concepts relationships using Wordnet and Wikipedia.

The next step after making queries is ranking the results. [23] has added two steps document ranking to the initial keyword based search to improve the results. The limitations of keywords based search are sometimes overcome by exploiting domain ontology of the knowledgebase. [16] used vector space model approach to find related ontology and improve document ranking. [29] compute the document relevance by comparing the similarities of words using ontology. [11] and [6] convert the free text content into semantic graphs and use graph matching algorithm to rank documents. [35] considered queries as concepts and documents as instances then uses ontology reasoning to calculate document relevance. These models uses semantic relations defined by ontology for query expansion or semantic similarity calculations and then rank the documents.

Keywords only search approach is popular because of ease of retrieving information. However, it lacks the in depth knowledge of user's search intentions. It also does not have enough expressivity in the search query. This might lead to less effective ranking of results when user's intentions are not completely clear. The hybrid approach to keyword search minimizes the issue [30]. This is seen in different cases when keyword search is expanded into extended query terms using ontological knowledge, or using graph traversal to find related objects. Semantic linkage also improves the accuracy of the search results when added to keywords based search.

5. CONCLUSION

The Suman system uses combination of keywords based semantic search, text based search and crowdsourced data to search for answers for unanswered questions in Q&A systems. It used Semantic Web technologies to integrate two datasets and semantically enrich them. The Suman system can be extended to include more datasets. It annotated the dataset with different concepts and then categorized the questions, answers and users. That provided broader and narrower search terms. The system also used crowdsourced information like votes, favourites and reputation points to rank the search results. There is evidence that the broader and narrower search terms and crowdsourced data have the potential to improve accuracy in searching for answers to unanswered questions. This can also be implemented to recommend experts in the field to get answers.

Statistical analysis was done on the results. It showed that the keywords generated by the Suman system were more prolific than the tags given to the original website data. It also showed that the participants agreed with the algorithm rating for answers provided for unanswered questions.

The Suman system has limitations, it uses Wikipedia-Miner and OpenCalais to do entity disambiguation. The limitations of those tools are limitation of the system. The Suman search algorithm uses crowdsourced data to rank the results. In the future another crowdsourced layer could be added on top where users can vote the search result to verify if it answered the unanswered question. The Suman system has not been compared to any existing search engines and algorithms. This is because the application is built on Stack-Overflow and Reddit dataset. It does not have resources of popular search engines and has not been designed to be better alternative than those search engines. It is a proof of concept to show that Semantic Web and Linked Data technologies can help to find answers for unanswered questions in Q&A forums.

6. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web: Scientific american. *Scientific American*, 284(5):34–43, 2001.

[3] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Information processing & management*, 43(4):866–886, 2007.

[4] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee. Linked data on the web (ldow2008). In *Proceedings of the 17th international conference on World Wide Web*, pages 1265–1266. ACM, 2008.

[5] U. Bojars, A. Passant, R. Cyganiak, and J. Breslin. Weaving sioc into the web of linked data. In *Linked Data on the Web (LDOW2008)*, 2008.

[6] F. Brauer, W. Barczynski, G. Hackenbroich, M. Schramm, A. Mocan, and F. Förster. Rankie: document retrieval on ranked entity graphs. *Proceedings of the VLDB Endowment*, 2009.

[7] D. Brickley and L. Miller. Foaf vocabulary specification 0.98. *Namespace Document*, 9, 2010.

[8] P. Castells, M. Fernandez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *Knowledge and Data Engineering, IEEE Transactions on*, 19(2):261–272, 2007.

[9] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg. Mining the web's link structure. *Computer*, 32(8):60–67, 1999.

[10] S. Corlosquet, R. Delbru, T. Clark, A. Polleres, and S. Decker. *Produce and Consume Linked Data with Drupal!* Springer, 2009.

[11] M. Daoud, L. Tamine, and M. Boughanem. A personalized graph-based document ranking model using a semantic user profile. In *User Modeling, Adaptation, and Personalization*, pages 171–182. Springer, 2010.

[12] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JAsIs*, 41(6):391–407, 1990.

[13] S. R. Eddy et al. A new generation of homology search tools based on probabilistic inference. In *Genome Inform*, pages 205–211. World Scientific, 2009.

[14] N. Eiron and K. S. McCurley. Analysis of anchor text for web search. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 2003.

[15] C. Fellbaum. *WordNet*. Wiley Online Library, 1998.

[16] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta. Semantically enhanced information retrieval: an ontology-based approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):434–452, 2011.

[17] H. Glaser, A. Jaffri, and I. Millard. Managing co-reference on the semantic web. In *WWW2009 Workshop: Linked Data on the Web*, April 2009.

[18] R. Guha, R. McCool, and E. Miller. Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, pages 700–709. ACM, 2003.

[19] C. Inc. Stardog 4: The manual, 2015.

[20] G. Klyne and J. J. Carroll. Resource description framework (rdf): Concepts and abstract syntax. 2006.

[21] R. Kosala and H. Blockeel. Web mining research: A survey. *ACM Sigkdd Explorations Newsletter*, 2(1):1–15, 2000.

[22] R. M. Lawrence Page, Sergey Brin and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[23] J. Li, J.-J. Yang, C. Liu, Y. Zhao, B. Liu, and Y. Shi. Exploiting semantic linkages among multiple sources for semantic information retrieval. *Enterprise Information Systems*, 8(4):464–489, 2014.

[24] Y. Li, Z. Bandar, D. McLean, et al. An approach for measuring semantic similarity between words using multiple information sources. *Knowledge and Data Engineering, IEEE Transactions on*, 15(4):871–882, 2003.

[25] D. Milne and I. Witten. An open-source toolkit for mining wikipedia. *Artificial Intelligence*, 2012.

[26] D. I. Moldovan and R. Mihalcea. Using wordnet and lexical operators to improve internet searches. *IEEE Internet Computing*, (1):34–43, 2000.

[27] E. Prud'Hommeaux, A. Seaborne, et al. Sparql query language for rdf. *W3C recommendation*, 15, 2008.

[28] T. Reuters. Opencalais. *Retrieved June*, 16, 2008.

[29] A. M. Rinaldi. An ontology-driven approach for semantic information retrieval on the web. *ACM Transactions on Internet Technology*, 9(3):10, 2009.

[30] C. Rocha, D. Schwabe, and M. P. Aragao. A hybrid approach for searching in the semantic web. In *Proceedings of the 13th international conference on World Wide Web*, pages 374–383. ACM, 2004.

[31] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[32] F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[33] P. Singh and N. Shadbolt. Linked data in crowdsourcing purposive social network. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 913–918. International World Wide Web Conferences Steering Committee, 2013.

[34] R. Trillo, J. Gracia, M. Espinoza, and E. Mena. Discovering the semantics of user keywords. *J. UCS*, 13(12):1908–1935, 2007.

[35] M.-y. You, L. Liang, J. Peng, and C.-y. Chen. Semantic information retrieval study based on knowledge reasoning. In *Fuzzy Information and Engineering Volume 2*, pages 271–280. Springer, 2009.