An Approach for Estimation – Reliability Model based Web Application Systems

John Blesswin
Karunya University
Coimbatore
Tamil Nadu
India

Sumy Joseph
Karunya University
Coimbatore
Tamil Nadu
India

Merlin Soosaiya
Karunya University
Coimbatore
Tamil Nadu
India

Priya K V
Karunya University
Coimbatore
Tamil Nadu
India

ABSTRACT
Structured retrieval of relevant data or information, based upon the user query is an essential factor while retrieving and traversing information resources on the World Wide Web. When the information is retrieved from the web, tags play a vital role to identify the relevant data and thereby providing the content to the user. Here we are proposing an approach in which the commonsense knowledge base will provide the gathered relevant information which is requested by the user query. To retrieve data from web, the use of commonsense knowledge will increase the accuracy of the result. When once the information is generated to the user by the common sense knowledge base, it requires evaluating for its quality and correctness. For this, a quantitative reliability estimation approach is explored. A general comparison between the existing approaches and the proposed approach has been also done.

General Terms
Commonsense knowledge Base, Tags, Reliability Estimation

1. INTRODUCTION
Tags are unconstrained keywords freely associated to a piece of information to describe it and assist in later re-finding [11]. Examples of tagging can be seen on sites such as Wikipedia and CiteULike etc. In Wikipedia, any registered user can create or edit entries and in CiteULike is a free service to help academics to share, store and organize the academic papers they are reading. Users tag citations, and add them to personal libraries.

Tagging ranges from a user's own tagging discipline, where the users are primarily tagging their own content for their own retrieval purposes and also to focus on other user's tagging discipline, where the user is tagging others' content for yet others to retrieve. The majority of the social bookmarking tools fall into the category of other's content tagged for their own retrieval purposes. Here, a couple of things [12] which are concerning about:

1. Server-side software aimed specifically at managing links with, crucially, a strong, social networking flavor

2. An open and unstructured approach to tagging, or user classification, of those links.

The Web is different with nodes of information connected in an open, free-form manner rather than being accessible only by navigating a pre-determined path hierarchy within a single authority domain [12]. The unstructured approach to classification of tags mainly associated with website. So the ability to sort out the proper content from the unstructured data is an important win over a web-based search engine. Normally, search engines tend to index and search a global space; not a user's local space. The major drawbacks include user privacy and tag spamming. So by publicizing the user's own tags or bookmarks, users are opening up to other users on the Web their own sphere of interests. Since blog comments are similarly vulnerable to attack as email spamming, the spamming of these new social tags can also possible to occur. Adware and spyware are already corrupting user's browsing experiences.

Commonsense is the ability to analyze a situation based on its context, using millions of integrated pieces of common knowledge. In other words, it is what people come to know in the process of growing and living in the world. A full understanding of any text then, requires a surprising amount of commonsense, which currently only people possess. Artificial intelligence is emerging to put commonsense knowledge into computers enabling machines to reason about everyday life. To retrieve data from web, the use of commonsense knowledge [10] will increase the accuracy of the result.

The knowledge base which is used so far includes Cyc, WordNet, and ConceptNet etc. Cyc project [16] was begun in 1984 by Doug Lenat and it was handcrafted by knowledge engineers. The main drawback is the unavailability to general public and it is very hard to use. In order to use Cyc knowledge base to reason about text, it is it is necessary to first map the text into its proprietary logical representation, described by its own language CycL [16] .But this mapping process is quite complex because all of the inherent ambiguity in natural language must be resolved to produce the unambiguous logical formulation required by CycL. The difficulties of applying Cyc to practical textual reasoning tasks, and the present unavailability of its full content to the general public, make it a prohibitive option for most textual-understanding tasks. Cyc [16] is optimized for formalized logical reasoning and the WorldNet is optimized for lexical categorization and word-similarity determination.

WorldNet [9] was begun in 1985 at Princeton University and it mainly a database of words, primarily nouns, verbs and adjectives which were organized into discrete senses. The main drawback is that the database contains only a small set of semantic relations such as Synonym, IS-A etc. Concept Net is the large-scale commonsense knowledge base[13] with an

integrated natural-language-processing tool-kit that supports many practical textual-reasoning tasks over real-world documents and it is optimized for making practical context-based inferences over real-world texts. ConceptNet was generated automatically from the English sentences of the Open Mind Common Sense (OMCS) [9] corpus. The main drawback is the lack of semantic knowledge in this knowledgebase.

Web based information retrieval is increasingly being used in various applications. So ensuring quality for the system is very crucial [1]. For complex systems, ensuring high quality with less verification is complex task. Reliability is one of the illusive targets to achieve in the relevant information retrieval for the successful web systems. There were many existing approaches used for estimating the reliability of the system. The existing approaches for reliability estimation were the scenario based evaluation [2], in which the parameters are determined by the runtime execution of the application. The experimental approach in which the parameters are extracted from the architecture based models by using the Automatic Test Analyzer (ATAC) tool [2].

Another method for predicting the reliability and performance of an application is by using some of the reliability estimation tool (SREPT). SREPT [3], is a unified framework tool used for evaluating the reliability and the performance. In this approach, the parameters evaluated are the test coverage and the inter-failure time data. The major drawback of this approach is, it does not provide the instantaneous fault detection [8]. And all these standards and methodologies assign predefined risk level to components [1], based on the criticality of the services in which they are involved. Here we are estimating how reliable and relevant the information, which is given to the user based upon his submitted query. A commonsense knowledge base is used as a repository for the generation.

Here in this paper the reliability can be estimated by the quantitative approach [1] with the calculation of some parameters. The important parameter is the transition probability [1] and the expected visit count. System reliability is the product of the individual reliabilities of the different tag component raised to the power of the number of visits to each units of application [2]. The test process of the obtained tag will be verified for its quality and analyzed. The fault tolerant mechanism [4] is also considered as one of the failure mitigation technique in this system.

The remainder of the paper is organized as follows: Section II deals about the proposed work and the architecture. The Commonsense Knowledge Base Construction [14, 15] and the Commonsense Knowledge Content Extraction is also described and the approach used for the estimation of reliability and the implementation of fault tolerant mechanism is explained. In order to evaluate the quality and the correctness of the data that has extracted is tested for its accuracy. Finally concludes the paper.

2. TAGGING REQUIREMENTS ON THE WEB

Unlimited number of tags can be associated with one website. Thus, managing a growing sea of tags is a concern for all social tagging systems [11]. There are many reasons for tagging of content on the Web. The tagging is a simple task where we will be having data and then we will describe the data. After that we will find appropriate keyword for the data under the consideration and finally add the labels to the data so that it is easy to track the data from the web. But in most cases tagging becomes ambiguous. Here user will be providing a particular tag for retrieving information or data from the web. A number of top ranked URL's will be considered for that particular tag entered for the evaluation process. In each content of the URL will be taken and the common sense knowledge creation is started at this point based on this content.

3. PROPOSED WORK

The proposed approach is using different techniques to retrieve the relevant data or information based upon the user query. The extraction of tag data for the relevant information is gathered. From this, a common sense knowledge base is constructed. Then, the evaluation for the quality of the obtained data is tested and analyzed.

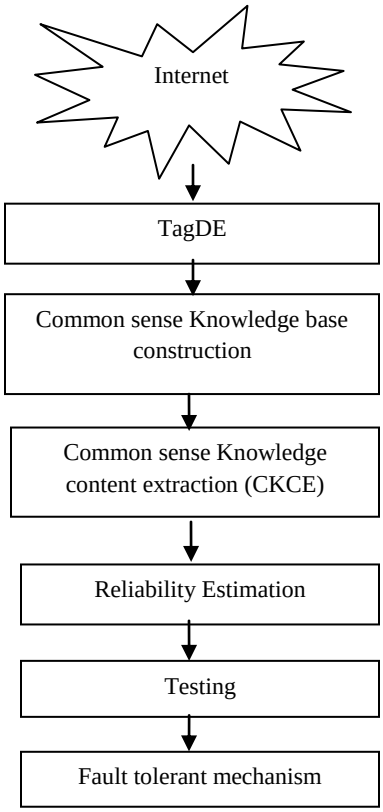

Fig 1: The Proposed Architecture

The Figure 1 shows the proposed architecture. It constitutes the following steps:

1. TagDE, the tag data extraction is the initial process of extracting the tag provided by the user and checks the relevant data from the web.
2. Common sense Knowledge Base construction, where the content of each URL will be considered based on the tag entered.

3. Common sense Knowledge Content Extraction (CKCE), in which the extraction of Common sense Knowledge content will take place.
4. Reliability Estimation of the selected content which is the outcome of CKCE.
5. Testing is conducted to check the relevance of the content.
6. Fault tolerant mechanism is finally provided.

3.1 Tag Data Extraction

The tags are keywords which will be given by the user for retrieving the information from the internet. So the tags will be entered with a particular context. User will be expecting to have relevant data for the specified tag entered. In order to provide the relevant data to the user common sense knowledge is created.

3.2 Knowledge Base Construction

When the users submit the query or the tag, we will consider the top ranked urls which are more relevant to the particular query. By visiting each of the URL, we will develop the common sense knowledge base. For this purpose, first we have to extract all the text from each urls. Each sentence in the text has a verb and some arguments associated with the particular verb. For the construction of the knowledge base we are considering the arguments like subject, object, locative information and temporal information. We will store these arguments of every verb in the text in a database. This technique is known as semantic role labeling [17].

Sometimes, these arguments may not be valid for the corresponding verbs. So the next phase is to evaluate the accuracy of those arguments. For this we have to perform lexical, syntactic and semantic analysis .In order to verify the arguments , we will create some dummy statements[14] and then substitute each of this arguments which is stored in the database by considering the verbs. In addition to this we will use some rules to discard the unwanted data from the database. Sometimes the subject of a particular sentence may contain a lot of words, if it happens we can say that it is not referring to commonsense. The other situation is that the subject may contain “it”, “him”, “her” etc we can also discard those record from the database. After performing this operation we will get databases which contain only the commonsense knowledge.

3.3 Common Sense Knowledge Content Extraction

Once the creation of commonsense knowledge base is completed, the next phase is to extract the relevant data from the top ranked urls. This extraction process will make use of the newly created commonsense knowledge base [13]. By considering the commonsense knowledge base the content of each url will be analyzed and retrieving the exact data content. Each time after the content is extracted; it should be stored in a repository. After visiting every top ranked urls we will get accurate information.

3.4 Reliability Estimation

The quality of the content which is extracted by this approach should be evaluated for its correctness. There are different parameters to be estimated for the evaluation. The user needs to get only the appropriate results for their queries. Reliability estimation can provide the assurance for the content. The Reliability is nothing but, the fault free service provision of software. Software reliability is defined as the probability of failure-free operation for a specified period of time in a specified environment [4],[1]. The model for the evaluation of reliability of the extracted content can be estimated with the support of the visit count and the transition probability among the different components. The transition probability matrix can be designed for the evaluation, which will capture the probability of transition of successful extraction of the content. The second parameters which can be used for the evaluation are the expected number of visit to the particular urls. This defines, how many times the particular url is traversed successfully [4, 7]. The transition probability matrix consists of the 'n' states and 'm' absorbing state which represents the number of urls and the number of visit to each url [1], [2]. We can represent the probability matrix values as in a form like

$$\begin{pmatrix} m & n \\ 0 & 1 \end{pmatrix} \quad (1)$$

We can represent the number of times the particular tag search as Si,j which is the number of times the tag is searched from the state i to state j and it can be shown as the expected number of visits from state i to j, Si,j=E[Si,j][1,2,4]. The visit of each component can be used for describe the usage of each tag in the application [2]. The system architecture can be represented by a sample model control flow diagram [1]. The DTMC [1, 2, and 5] architecture can be used as a sample model, in which we consider that there n URL which are searched for the required content.

The initial component is represented as 1 and the final component by n. DTMC represents the each tag component and the transition from state one to it related tags. By this we can count the expected visit to different components and its variance. The parameters like the expected visit count and the transition probability can be used to evaluate the overall system reliability as a function of component reliability [3].

3.5 Testing

During the testing process, we have to evaluate the relevance of both the commonsense knowledge and the content retrieved. For this purpose, during the first visit itself to the URL we need to store all the content of each url into a storage repository. Then by using this stored data we can make a comparison with the knowledge base and the content. For that we can integrate the proper testing tool. The test result can models, the effectiveness of the data content and how much relevant of the information which is obtained based on the required user query.

3.6 Fault Tolerance Mechanism

The model considers the failure mitigation mechanism [1] also for the successful operation of the system. Fault tolerant mechanism is nothing but it is the ability of the system to handle the failure when any unexpected event occurs [8, 1, and 3]. Here we can implement the failure mitigation technique in such a way that, we will be evaluating the data with the respect of the query. If it is not relevant with the query, we will be reran king the URL. The Figure 2 shows the sample model for the architecture of the different tag transition probability.

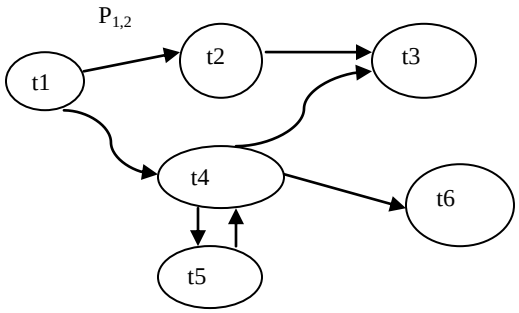

Fig 2: A Sample Model

3.7 Comparison

There are many existing tools and methods available for the reliability estimation. But there are some drawbacks among these approaches. All these, approaches do not consider the architecture of the system. Moreover the tools do not consider the effect of the operating system and the operational effect on the reliability. These drawbacks are overcome in the proposed method of reliability estimation.

4. CONCLUSION

Tagging the content on the web is becoming a very significant activity while retrieving relevant content from the web. The relevance of the web content depends on the way the user tag the content for own purposes or for other user's retrieval options. To retrieve data from web, here by using the commonsense knowledge base will increase the accuracy of the result and also provide the relevant data for the user query. The quality of the content which is extracted by this approach should be evaluated for its correctness. Hence, this approach helps to retrieve the relevant data or information efficiently from the web.

5. REFERENCES

[1] Software Reliability and Testing Time Allocation: An Architecture-Based Approach Roberto Pietrantuono, Member, IEEE, Stefano Russo, Member, IEEE, and Kishor S. Trivedi, Fellow, IEEE, IEEE transactions on software engineering, vol. 36, no. 3, may/June 2010

[2] S.Ramani, S.Gokhale and K. Trivedi,"SREPT: Software Reliability Estimation and Prediction Tool", Performance Evaluation, Special issue on Tools for Performance Evaluation, vol.39, no. 1,pp. 37-60, 2000 [39]

[3] S.S.Gokhale, W.E Wong, J. R.Horgan and Kishor S. Trivedi, "An Analytical Approach to Architecture-Based Software Performance and Reliability Prediction", Performance Evaluation, vol.58, no. 4,pp. 391-412, 2004.

[4] A. Mettas,"Reliability Allocation and Optimization for Complex Systems," Proc. Ann. Reliability and Maintainability Symp. pp. 216-221, 2000.

[5] K.Goseva-Popstojanova, and K. S Trivedi, "Architecture-based approach to reliability assessment of software systems", Performance Evaluation, vol.45, nos2/3, pp.179-204, 2001.

[6] K. Goseva-Popstojanova, A.P Mathur, K. S. Trivedi,"Comparison of Architecture Based Software Reliability Models".

[7] M. R. Lyu,, S. Rangarajan, and A. P. A. van Moorsel,"Optimal Allocation of Test Resources for Software Reliability Growth Modelling in Software Development", IEEE Trans. Reliability, vol.51, no.2,pp. 183-192, June 2002

[8] S. Yacoub, B. Cukic, and H. H. Ammar, "A Scenario-Based Reliability Analysis Approach For Component-Based Software", IEEE Trans. Reliability, vol.53, no.4, pp.465-480, Dec. 2004 [36]

[9] Sheng-Hao Hung, Chia-Hung Lin, Jen-Shin Hong," Web mining for event-based commonsense Knowledge using lexico syntactic pattern matching and semantic role labeling", Expert Systems with Applications, Volume 37, Issue 1, Pages 341-347, January 2010

[10] Palmer, M., Kingsbury, P., & Gildea, D."The proposition bank: An annotated corpus of semantic roles", Computational Linguistics, 31(1), 71–106, 2005

[11] Margaret-Anne Storey, Jody Ryall, Janice Singer, Del Myers, Li-Te Cheng, and Michael Muller, "How Software Developers Use Tagging to Support Reminding and Refinding,"IEEE Transactions on software engineering, vol.35, No.4, pp. 470- 483, Jul/Aug.2009

[12] T. Hammond, T. Hannay, B. Lund, and J. Scott, "Social Bookmarking Tools: A General Review," D-Lib Magazine, vol. 11, no. 4, Apr. 2005.

[13] Richardson, D. S., Dolan, B. W., & Vanderwende," MindNet: Acquiring and structuring semantic information from text", In Proceedings of the 17 international conference on computational linguistics (pp. 1098–1102), 1998.

[14] Aone, C., & Ramos-Santacruz, M., "REES: A large-scale relation and event extraction system", In Proceedings of the sixth conference on applied natural language processing (pp. 76–83). Seattle, Washington, 2000.

[15] Liu, H., & Singh, P.,"MAKEBELIEVE: Using commonsense knowledge to generate stories", In Proceedings of the 18th national conference on artificial intelligence, AAAI, (pp. 957–958). Edmonton, Alberta, Canada, 2002.

[16] Lenat, B. D., & Guha, V. R.,"The evolution of CycL, the Cyc representation Language", ACM SIGART Bulletin, 2(3), 84–87, (1991).

[17] Gildea, D., & Jurafsky, D.," Automatic labeling of semantic roles". Computational Linguistics, 28(3) pp 245–288, (2002).