Semi Automated Semantic Matched Concept Extraction Model for E-Content Development

D. Elangovan
Research Scholar, M.S University, Tirunelveli, Tamil Nadu, India.

K. Nirmala
Associate Professor, Dept. Of CSE, Quaid-E-Millath Government College for Women, Tamil Nadu, Chennai, India.

Abstract
This paper deals with an effective methodology which is very useful for e-content. Various researchers have used this area and Naive based algorithm is widely used. Whereas in this paper proposed stemming based algorithm is superior than the existing one. In terms of speed, accuracy and efficiency. In this approach the words are stemmed from the document and classified based on knowledge base and organize into a tree structure. By applying the probability techniques the semantic word retrieved is used for e-content.

Keywords: semantic matched concept, e-content, Naive based algorithm, SAS based algorithm

Introduction
Accessing and extracting semantic information from web documents is beneficial to both humans and machines. Humans can browse and retrieve documents in a semantically manner whereas machine can easily process such structured representations. Integrating extracted information from multiple documents can provide users with a global knowledge model of some domains. Due to the structure of human knowledge, the tasks of extracting semantic information in web documents, however, proved to be difficult. The vision of Semantic Web (Berners-Lee et al, 2001) offers the possibility of providing the meanings or semantics of web documents in a machine readable manner. However, the vast majority of 1.5 billion web documents are still in human readable format, and it is expected that this form of representation will still be the choice among content creators and developers due to its simplicity. Due to this phenomenon and the desire to make the Semantic Web vision a reality, two approaches have been proposed (van Harmelen & Fensel, 1999): either furnish information sources with annotations that provide their semantics in a machine accessible manner or write programs that extract such semantics of Web sources. Semantic meaning and representation of the input query in describing user’s needs remain as major challenges have been presented by Russ, J. C. (1999). The semantic extraction techniques exploring the strength and the weaknesses of the existing semantic extraction techniques has been described in Wang et al.,2010. Object/region semantic extraction is provided by researches (Rui, Y., Huang, T.S., and Mehrotra, S. (1997), Mori, Y., Takahashi, H., and Oka, R. (1999), Zhao, R., and W. I. Grosky. (2001), Wang, L., Khan, L. (2006)) from manual . In [11], user evolving methodologies are discussed using K-means clustering. Based on the literatures, we come with the model which is used to develop the semi automated semantic match extraction model for developing the e-content.

Architecture of the SAS Model For E-Content
The Document classification is a big problem in information science technology. The problem is to classify the document to one or more classes or categories. This can be in different ways using manually or algorithmically. The documents to be classified may be texts, images, music, etc. To concentrate only on the text classification problems, the text may be classified according to their verb; adverb and adjective to map based on the knowledge extraction concepts. There are two main classifications of texts, one is content based approach and second is request based approach. Content based approach is classification based on the subject in a document determines the class to which the document is assigned. Request oriented classification (or-indexing) is classification in which the anticipated request from users is influencing how documents are being classified.

Figure 1: Architecture of the SAS e-content Development

Stemming is used in information retrieval to describe the process for reducing inflected words to their word stem, base
or root form. As of part of supporting language is analyzing words to find their stem or root form. The stem needs not to be identical to the root of the word, it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. An example of stemming is reduction of the words “run”, “running”, “runs” and “ran” to their stem “run”. For this process, we have different type of algorithm to implement the stemming process. Here we going to use SAS algorithm, this is a pure java based algorithm and it will support different languages. SAS uses a file containing the grammatical rules for languages encoded in standardized format and a dictionary file containing the languages valid stems. The analysis of a word loops over the grammatical rules applying those applicable and then checking if valid stem is found. The advantages of SAS are that very complicated grammatical rules can be applied such as the removal of multiple suffixes and prefixes. Suffix stripping algorithm may differ in the result for variety of reasons. One such reason is whether the algorithm constrains the output word must be a real word in the given language. Some approaches do not require the word to actually exist in the lexicon database. Some suffix stripping approaches maintain a database for all known word roots that exist as real words. The quality of the analysis will most likely be much less than the analysis of English. As such the impact on the SAS index and search result are hard to predict. Semantic is a defined over a set of documents, where the idea of distance between them is based on the likeness of their meaning or semantic content as opposed to similarity which can be estimated regarding their syntactical representation. Based on text analyses, semantic relatedness between units of language can also be estimated using statistical means such as a vector space model to correlate words and textual contexts from a suitable text corpus.

Navie bayes algorithm:
Navie bayes is a Machine Learning Algorithms need to be trained for supervised learning tasks like classification, prediction etc. or for unsupervised learning tasks like clustering. By training it means to train them on particular inputs so that later on test them for unknown inputs for which they may classify or predict etc based on their learning. This is very important that the test set be distinct from the document: simply reused the document as the test set, then a model that simply cache its input, without learning how to generalize to new examples. In Naive Bayes, the concept of ‘probability’ is used to classify new entities. In conditional probability is the probability that something will happen, given that something else has already happened. In Naive Bayes is a way to go from P (Evidence/ Known Outcome) to P (Outcome/Known Evidence). Often frequently some particular evidence is observed, given a known outcome. To use this known fact to compute the reverse, to compute the chance of that outcome happening.

SAS Model:
In SAS Model, the Probability of having both the Outcome O and Evidence E is: (Probability of O occurring) multiplied by the (Prob of E given that O happened). The evidence, P(Outcome or Evidence) = P(Evidence given that the Outcome) times Prob(Outcome), scaled by the P(Evidence)
In Naive Bayes, to predict an outcome of multiple evidence that case, the math gets very complicated. To get around that complication, one approach is to ‘uncouple’ multiple pieces of evidence, and to treat each of pieces of evidence as independent. This approach is called as SAS. P(outcome/evidence) = P(Likelihood of Evidence) x Prior prob of outcome/ P(Evidence)
Figure 1, the Prob (outcome/ evidence) is 1, just multiplying by 1or Prob (some particular evidence/outcome) is 0, then the whole prob. becomes 0. If you see contradicting evidence that can rule out outcome. Since divide everything by P(Evidence). The intuition behind multiplying by the prior is so that gives high probability to more common outcomes, and low probabilities to unlikely outcomes. These are also called base rates and they are a way to scale our predicted probabilities. The formula above for each possible outcome is trying to classify, each outcome is called a document and it has a document label. Our job is to look at the evidence, to consider how likely it is to be this document is assign a label to each entity.

Algorithm for SAS:
In this algorithm, it mention D as document, N as no of words, C as Evidence, T as outcome.
Document(C, D) V < ExtractVocabulary(D) N < Countword(D) for each c with C do Nc < countwordinDoc(D, C) prior[c] < Nc/N text < concatenateTextofallWordAsDoc for each t with V do T < countwordofT(erm(Text, t) for each t with V do condprob[t][c] < -(T+1)/sum(T + 1)
return V, prior, condprob
In the document iterate the evidence to extract the vocabulary text content and count the number of words. The word will match the knowledge type to find the evidence in the document and concatenate the text evidence as knowledge set. In each set is loop through to find the outcome. This is repeated for all the set of document until get the extract knowledge of outcome. Then find the probability of the outcome. The document that has the highest probability is declared the winner and that document label gets assigned to that combination of evidences. Assign the document label of whichever is the highest number. SAS turns out to be excellent in this application to classify the text using the knowledge extraction.

Result of knowledge based extraction (factual, procedural, and conceptual):
Knowledge extraction is the creation of knowledge from structured and unstructured text, documents sources. The main criterion is that the extraction result goes beyond the creation of structured information. It requires either the reuse of
existing formal knowledge or the generation of a schema based on the source data. There is different knowledge extraction type such as procedural knowledge, conceptual knowledge and functional knowledge. Conceptual knowledge is rich in relationships and understanding the text by classifying as verb. It is connected web of knowledge, a network in which the linking relationships are as prominent as the discrete bits of information. And procedural knowledge is formal language or symbolic representation as noun; it has set of rules, algorithms and procedures. Functional knowledge is knowledge that is concrete and usable rather than abstract and theoretical. It has different terminology to understand and solve simple problems using the adjective and adverb.

Factual:
What is it, how to represent

Procedural:
What is it, how to represent

Conceptual:
What is it how to represent

Table 1: SAS knowledge based extraction from semantic matched concept

<table>
<thead>
<tr>
<th>No. Of Words</th>
<th>No. Of Knowledge</th>
<th>HAD SAVES The...</th>
<th>FACTUAL</th>
<th>PROCEDURAL</th>
<th>CONCEPTUAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>45</td>
<td>12</td>
<td>0.5</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>

In SAS knowledge based extraction from semantic matched concept, the result of knowledge base shows the comparison of SAS and Naive Bayes. From the Table 1 result shows SAS has low probability than Naive Bayes, even in the multiple semantic document extraction.

Conclusion
This proposed methodology is more efficient and accurate with respect to the existing naive based algorithm. The existing methodology does not support classification and it lacks accurate and not efficient. The proposed methodology also helps in getting relevant and meaningful; words because it uses semantic classification and stored based on the knowledge base, the advantages to the user to fetch the semantic word from the e-content.

References