Fall 2011

DISCOVERING KNOWLEDGE STRUCTURE IN THE WEB

Siddharth Ramu
San Jose State University

DISCOVERING KNOWLEDGE STRUCTURE IN THE WEB

A Project Report

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements for the Degree

Masters of Computer Science

By,

Siddharth Ramu

Dec 2011

SAN JOSE STATE UNIVERSITY

The Undersigned Project Committee Approves the Project Titled

DISCOVERING KNOWLEDGE STRUCTURE IN THE WEB

by
Siddharth Ramu

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

Dr. Tsau Young Lin	Department of Computer Science	Date
Dr. Soon Tee Teoh	Department of Computer Science	Date
Dr. Howard Ho	IBM Almaden Research Center	Date

APPROVED FOR THE UNIVERSITY

Associate Dean	Office of Graduate Studies and Research	Date

ABSTRACT

Discovering Knowledge Structure in the Web

by Siddharth Ramu

In this project, we implement a new concept to extract knowledge or the semantic meaning from the Internet data. We apply the Association Rule and the apriori principle to a chosen set of documents from the Internet and analyze the associated benefits and drawbacks with this new concept.

First, we find the highly relevant keywords in all documents by using the tf-idf. From these keywords, we find all keyword pairs (finite sequences of length 2), that are within a distance of 30 words. We then take keyword pairs with high frequency and find keyword triplets (finite sequence of length 3) … and so on, until there are no more high frequency finite keyword sequences. At each stage, we apply the Association Rule to find the primitive keywords that do not grow any longer. This is known as the Primitive Concept. We find the next set of keywords that can be associated from the non-primitive keywords. In this way, we find the longest keyword sequence that exists. We observe that this finite frequency of keywords often represents a concept in the web. For example, "closeness centrality" keyword pair represents social networking. Another generic example

could be “Wall Street”. This keyword pair represents the New York stock exchange.

In this project, we apply the Association Rule and the apriori principle on a set 21 IEEE papers on social networking. We limit our research to finding the primitive concept for each document and analyze the results obtained.

ACKNOWLEDGEMENTS

I wholeheartedly thank Dr. Tsau Young Lin for his guidance and encouragement throughout the course of this project. I also thank Dr. Chris Pollett and Dr. Soon Tee Teoh for providing valuable feedback and suggesting new ideas to improve my work. I ought to thank two of my SJSU friends Michael Crawford and Amrapali Dhavare for lending a helping hand in the initial stages of the project. I thank my family and close friends for their continuous support and cooperation, without which I would not have achieved anything.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

1 Introduction

In this project, we test our new approach of web mining. We have considered 21 documents for this research work. All these documents are IEEE papers in the field of social networking. The importance of considering similar set of documents for this research work is that we can closely study the performance of our algorithm on this data. Therefore it helps in applying our concept on a larger scale.

2 Tokenization

Tokenization is a technique in which a text document is broken into its individual words. These individual words are also known as tokens. For all the 21 documents that we consider in this project, we identify the tokens. In addition to identifying the tokens, we also store the position each token in every document. The position is stored because we want to text-mine and find the concept in each document, for every 30 words. Figure 1 depicts the format in which tokens are stored, along with its position and document location:

```
4   ieee    doc1.txt
5   international   doc1.txt
6   conference  doc1.txt
8   social  doc1.txt
9   computing   doc1.txt
11  ieee    doc1.txt
12  international   doc1.txt
13  conference  doc1.txt
15  privacy doc1.txt
17  security    doc1.txt
```

Figure 1: Tokens stored along with the position and document location

3 TF-IDF

The term frequency-inverse document frequency (tf-idf) of a token, as the name suggests, is a measure of the frequency of a token and its inverse document frequency. The Tf-idf is a weighing scheme that assigns each term in a document a weight based on its term frequency (tf) and inverse document frequency (idf) (1). The tf-idf weight of a token is a value that is used to determine how relevant that token is, in a particular document. A token having a high tf-idf value when compared to the other tokens implies that it is very important in the documents that have this token. The tf-idf weighting scheme is one of the most popular weighting schemes in information retrieval (1).

3.1 Term frequency

The term frequency of a token is a measure of how many times this token appears in a document. The term frequency for a token t in a document d is calculated from the below formula:

$$\log\left(\frac{1000}{200}\right) = 0.69$$

$$tf_{t,d} = \frac{t_d}{T_d}$$

Equation 1: Calculate Term Frequency

Here T is the number of tokens t in document d, divided by the total number of terms in that document. For example, if the term “computer” appears 5 times in a document containing 100 words, then tf(“computer”) will be 5/100 = 0.05

3.2 Document frequency

The document frequency of a token is a measure of how many documents contain that token. The formula for calculating the document frequency df of a token t is:

$$df_{t,d} = D_t$$

Equation 2: Calculate Document Frequency

Here, D is the number of documents that contain token t. It does not matter how many tokens are present in each document. We take the count of documents that contain atleast one token t. The document frequency is needed in order to calculate the inverse document frequency.

3.3 Inverse Document Frequency

The inverse document frequency (idf) of a token t and D number of documents is defined as:

$$idf_{t,d} = \log \frac{D}{df_{t,d}}$$

Equation 3: Calculate Inverse Document Frequency

In other words, the idf of a term t is the logarithm of total number of documents divided by the document frequency of that token. Thus the idf of a rare term is high, whereas the idf of a frequent term is likely to be low. (2)

From the above equation, we observe that if a token is present in all the documents, then its df will be equal to D. Therefore, the idf of that token will be a log(1), which is Zero. This is completely valid because, if a token is present in all the documents, then it is one of the stopwords such as “and”, “the”, “is” etc. Since the idf of these stopwords is 0, we can easily eliminate these tokens from our study as they are least important.

3.4 tf-idf

The tf-idf of a token t in document d is its tf multiplied with its idf:

$$tf - idf_{t,d} = tf_{t,d} \times idf_{t,d}$$

Equation 4: Compute tf-idf

In addition to the use of TF-IDF to filter out irrelevant words based on the TF-IDF threshold value, I used a list of 180 stop-words to filter out any remaining stop-word from the result. This further refines the resulting keywords, thus providing a list of highly meaningful keywords.

4 Association Rule

Association rules are "if-then rules" with two measures which quantify the support and confidence of the rule for a given data set (3). Several large organizations that are into retail store their enormous sales data in large databases. This large amount of sales data that is stored is called as *basket data* (4). Each entry in this database represents one transaction. The transaction date and the items that are purchased with this transaction are stored in each record. Organizations use this data for their analysis to manage their inventories. For example, consider a large store that mainly specializes in selling Halloween costumes. Upon observing their basket data in the database, the organization observed that their Spiderman costumes have been the bestselling item for two successive weeks. Upon analyzing this data, the organization can come to the conclusion that the customers like their Spiderman costume more than any other costume that they have in the store. Now they can increase the supply of Spiderman costumes in all stores across various locations to meet the customer demand.

The above analysis is very simple and every organization, whether large or small, will be performing this analysis to meet the demands of their customers. However, if we want to know the relationship between two or more products that are sold in an organization, with the above discussed approach is not sufficient to accomplish the task. We need to perform an in-depth analysis of the transaction in the basket data. This is where the Association Rule comes into picture.

The Association Rule is an efficient algorithm proposed by Rakesh Agrawal, Tomasz Imielinski and Arun Swami of IBM Almaden Research Center, San Jose, in 1993 (5). This algorithm identifies all significant associations between items in a database. An example of an Association Rule is that 90% of transactions that purchase bread and butter also purchase milk (5).

Once this rule came into place, many supermarkets started using it for analyzing their transactions and taking decisions accordingly, so as to improve their sales and to satisfy their customers. This rule was applied on supermarkets to find out the relationship between the products that customers buy. For example, since it was found that if a customer buys milk and bread, he or she is most likely to buy butter as well, the supermarket can keep milk, bread and butter all in the same row so that customers can find them easily, without having to search for these items elsewhere in the store. An article published in The Financial Times of London (Feb 7th 1996) stated that a supermarket in the U.S discovered a strong

association for many customers between a brand of babies diapers and a brand of beer. So the retail outlet was able to exploit it by moving the products closer together on the shelves (6).

There are two steps in using Association Rule:

1. Firstly, all possible combination of items is found in the database
2. Secondly, a minimum confidence constraint is applied to find only the most significant set of items.

It is difficult to find all possible combination of items in a database since the set of all possible items is having complexity of 2^{n-1}, where n being the total number of items in the database. However, there is an efficient approach known as the downward-closure property, which is used by the Apriori Principle (7).

4.1 Apriori Principle

The Apriori Principle simplifies the complexity involved in finding all possible combinations, by selecting only the most frequent items for further processing. This eliminates the non-frequent items, which would not result in any frequent itemset.

Let us consider the example of a supermarket that uses the Association Rule for analyzing the buying pattern of its customers. From the below table, we see that there are five customers who bought different products.

Customer	Bread (B)	Milk (M)	Cereals (C)	Jam (J)
1	1			1
2		1	1	
3	1	1	1	
4	1	1	1	1
5	1	1		1

Table 1: Customer purchase in a supermarket

Applying the Association Rule on this data, first we find the most frequently purchased item combinations. To do this, we have to find all possible combinations of the items. Applying the apriori principle, we select a threshold value for the purchased items. The itemsets that are having a lower count are discarded. Only the items that are having the count above this threshold value are considered for further processing.

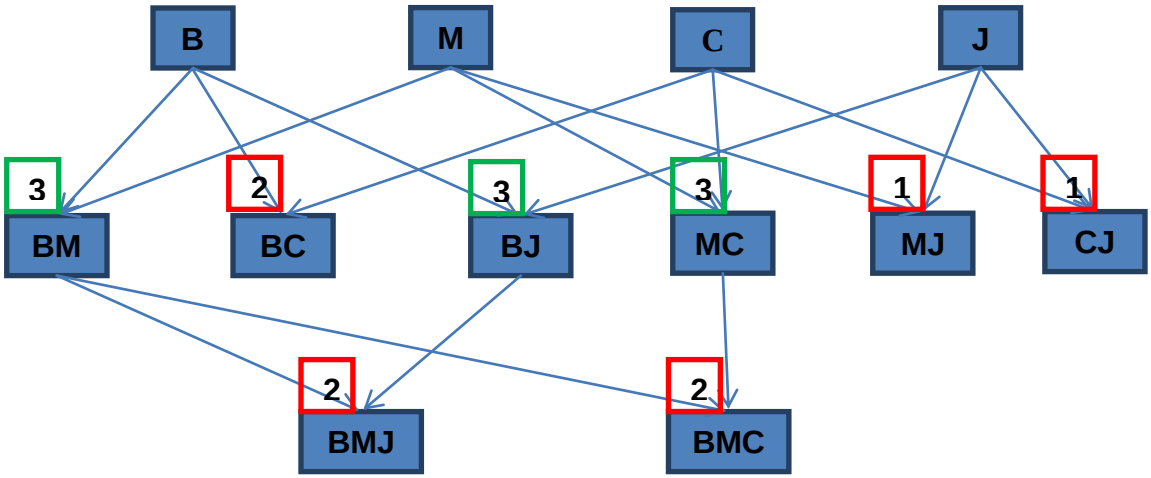


Figure 2: Selecting items with threshold frequency as 3

From the above figure, we first find all possible combinations of the items. On applying the apriori principle, we decide upon a threshold value, to filter out the less frequent items. In this example, we chose the threshold frequency as 3. Therefore, the items that have a count below 3 are discarded. In the above figure, we observe that only combinations of Bread and Milk, Bread and Jam and Milk and Cereals satisfy this condition. Therefore they are considered for further processing. These itemsets are termed as Frequent Itemsets, as they occur frequently (8). Applying the Association Rule on these items, we get two more itemsets, namely, Bread, Milk and Jam, and Bread, Milk and Cereals. However, these two itemsets do not satisfy the apriori principle threshold criteria and are therefore not processed further.

Thus, from the above analysis, we can conclude that majority of the customers those who bought Bread and Milk in the supermarket also bought either Jam or Cereals. It must be noted that, this is a very small example to apply data mining principles.

5 Applying Association Rule for Web Mining

The above discussed Association Rule is mainly used to benefit supermarkets or any other factory inventories. However, we could also apply the Association Rule for Web Mining as well. In this project, we extend the Association Rule and apply it for Web Mining to obtain meaningful keywords. In this process, we first identify tokens from the web documents. Neighboring tokens, separated by a certain distance, are paired together to form keyword pairs. Two keyword pairs that are associated by the same first keyword are then joined together to form a three keyword. The most relevant three keywords are chosen by applying the Apriori Principle, which removes keywords that are least occurring in all documents. Similar approach is followed to find four, five, six or more keywords. The process is continued until we find the longest keyword, which cannot be associated with any keyword anymore. This longest keyword represents the concept contained in that particular web document. This keyword is also referred to as the Primitive Concept, since the documents containing this keyword are closely related to this concept.

The procedure for finding the longest keyword sequence is discussed below:

5.1 Keyword pairs

After obtaining the list of relevant keywords by applying the TF-IDF threshold value, in order to apply the Association Rule, we first pair each of these keywords, separated by a distance of 30 words. This means that the distance between each keyword in the pair is 30 words. We take 30 as the distance because we want to extract the meaning of the document, for every 30 words. Here we are assuming that on an average, around 30 words are required to convey some specific information.

5.2 Applying apriori principle

Once we find all possible combinations of highly frequent keywords, we apply the apriori principle to select only the most frequent keyword pairs. For this, we calculate the document frequency (df) of each keyword pair. The df of a keyword pair is the count of this pair in all documents in the Web. We calculate the df to ascertain the importance of a keyword. If the pair appears in very few documents, then this keyword pair is considered to be unimportant. If the keyword pair is appearing in several documents, then we keep this pair, as it could be possibly be important. Similarly, the df is calculated for all keyword pairs in every document. We identify a threshold value for the Document Frequency.

Keyword pairs having df below this threshold value are eliminated, whereas the rest are considered for further processing.

5.3 Three keywords

For each keyword pair thus obtained from the above step, we then find three keywords. This is where we apply the Association Rule. Using the concept of Association Rule, we consider every two keyword pairs that have the same first token, but different 2nd token. Also, all the tokens considered must be within the 30 word distance. Now a three keyword is formed by combining all the three tokens.

For example, we have the following two keyword pairs that have the first token same:

Keyword Pair	token1	token2	Position1	Position2
1	ieee	international	4	5
2	ieee	conference	4	6

Table 2: Keyword pair association

The numbers on the right of each pair denote the position of each token in a document. Since the two keyword pairs are within the 30 word distance, we form the following 3-keyword:

token1	token2	token3	Position1	Position2	Position3
ieee	international	conference	4	5	6

Table 3: Formation of a 3-keyword

Similarly, we find the 3-keywords for each document. For each of these 3-keywords, we then calculate the df. We identify a threshold for the df. 3-keywords that are below this threshold are discarded, whereas the rest are kept for further processing.

However, if the 3-keywords could not be formed because there aren't any keyword pairs that satisfy the Association Rule, then we could conclude that 2-keywods are the longest keywords that are possible for the set of web documents considered.

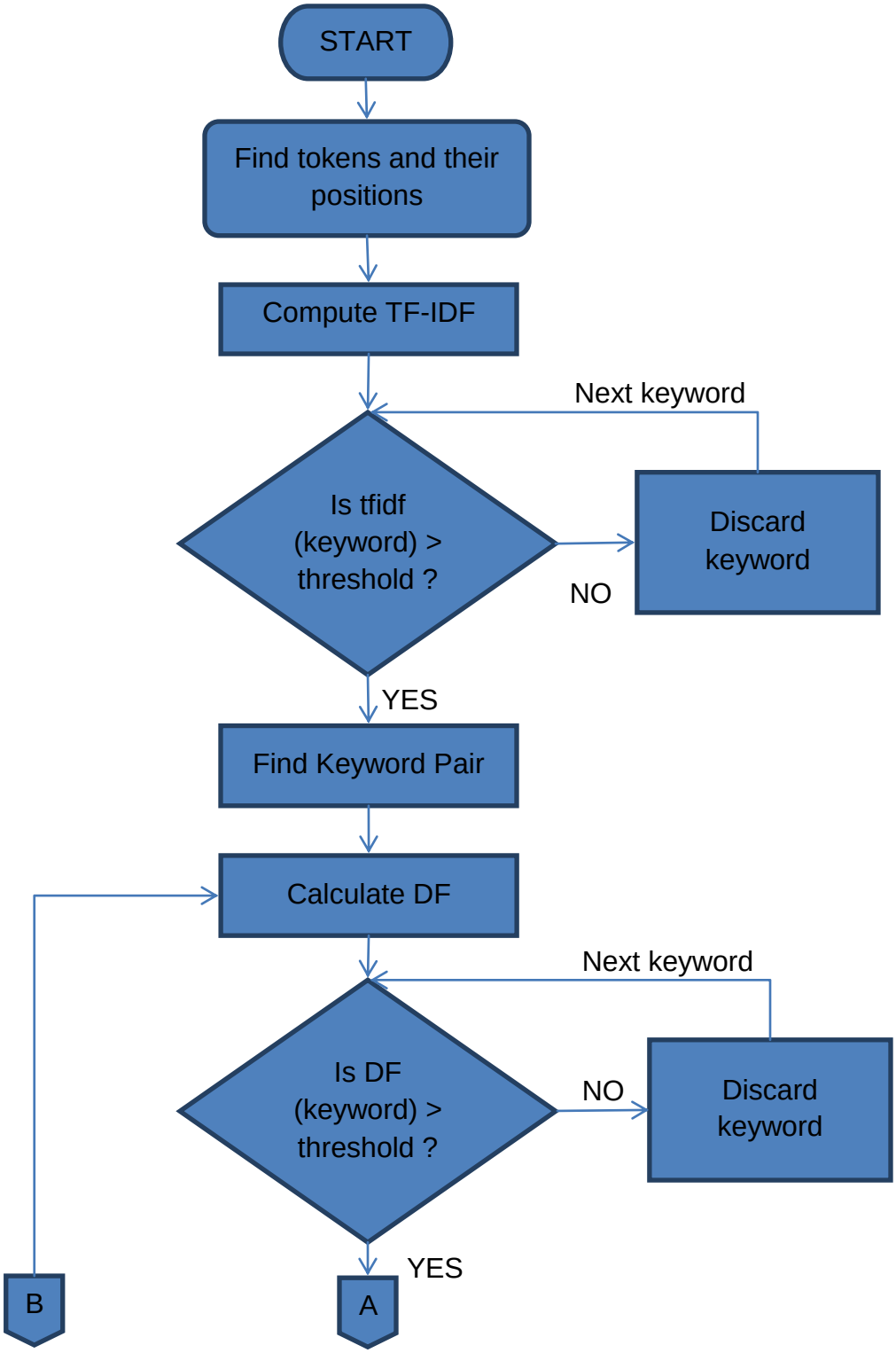
5.4 Finding n-keywords

Similarly, we apply the Association Rule to the 3-keywords and find the 4-keywords and so on. At each stage, we filter out keywords with low frequency by applying the Apriori Principle and proceed to the next keyword sequence. Continuing this way, we found a sequence of 5-keywords as the longest sequence of keywords, from our test data. One of the longest keyword sequences is shown in the below:

token1	token2	token3	token4	token5
build	mobile	network	Relations	logs

Table 4: Five keywords found using the Association Rule

5.5 Flow Chart



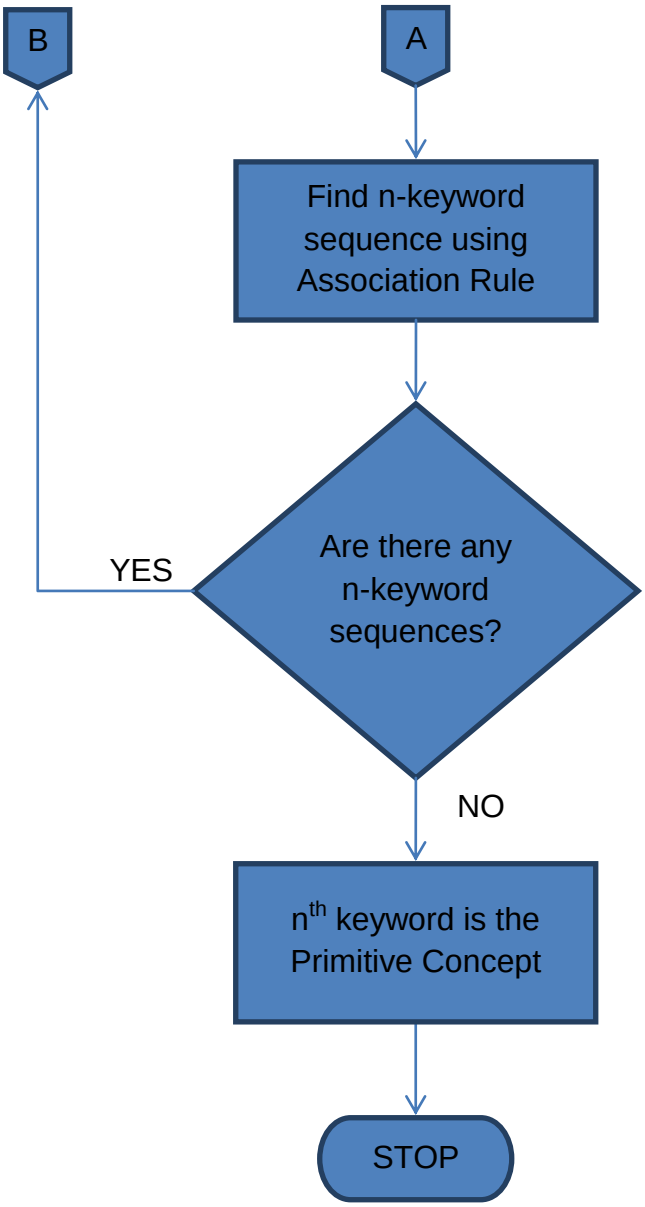


Figure 3: Flow Chart depicts the implementation

6 Primitive Concept

We define the Primitive Concept as the keyword sequence that does not grow further. The longest primitive concept represents the basic idea on which the entire document is written.

Let us consider two primitive concepts for three-keywords that we obtained in our project:

token1	token2	token3
centrality	conceptual	networks
centrality	conceptual	social

Table 5: Primitive Concept for three-keywords

These two primitive concepts do not form a four-keyword association, since their association did not satisfy the apriori principle. The document frequency of this keyword sequence was less than the threshold. Therefore, their three-keywords formed the Primitive Concept.

token1	token2	token3	token4
centrality	conceptual	social	networks

Table 6: Four-keyword not formed since it did not satisfy the Apriori Principle

Now let us consider a longest primitive concept:

token1	token2	token3	token4	token5
build	mobile	network	relations	logs

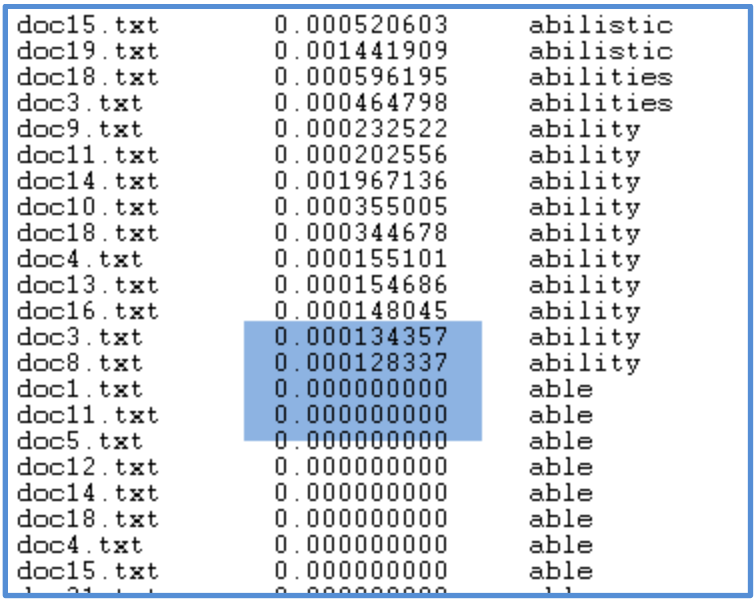
Table 7: Longest primitive concept

From this keyword sequence, we can interpret that the document containing this primitive concept is containing information on mobile networks logs.

7 Experimentation results

In this section, we discuss the results we obtained.

We first calculate the tf-idf for all the tokens. The below figure shows the tf-idf values obtained for every token in each document. We can observer that one of the stopwords “able” is having the tf-idf value as 0, whereas other words are having a higher value. Thus, the use of tf-idf eliminates most of the stopwords in the documents.

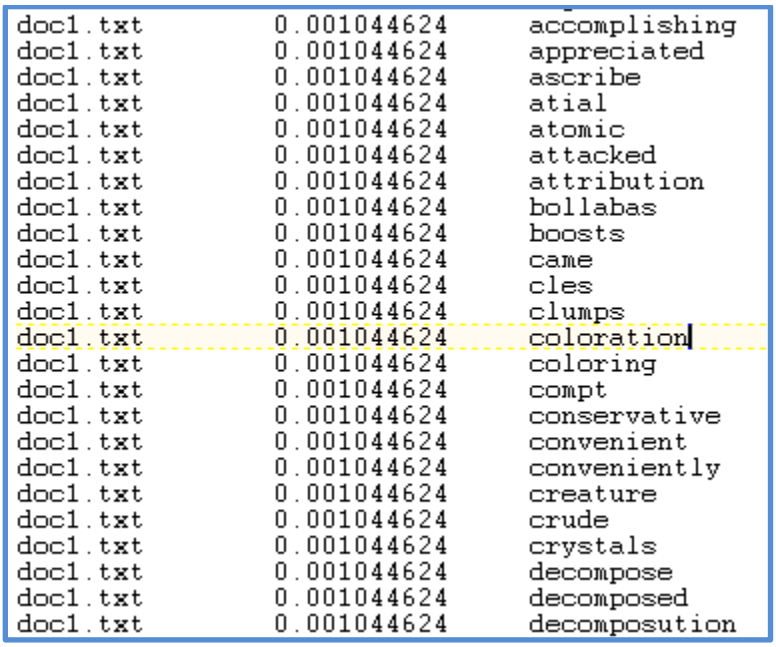


doc15.txt	0.000520603	abilistic
doc19.txt	0.001441909	abilistic
doc18.txt	0.000596195	abilities
doc3.txt	0.000464798	abilities
doc9.txt	0.000232522	ability
doc11.txt	0.000202556	ability
doc14.txt	0.001967136	ability
doc10.txt	0.000355005	ability
doc18.txt	0.000344678	ability
doc4.txt	0.000155101	ability
doc13.txt	0.000154686	ability
doc16.txt	0.000148045	ability
doc3.txt	0.000134357	ability
doc8.txt	0.000128337	ability
doc1.txt	0.000000000	able
doc11.txt	0.000000000	able
doc5.txt	0.000000000	able
doc12.txt	0.000000000	able
doc14.txt	0.000000000	able
doc18.txt	0.000000000	able
doc4.txt	0.000000000	able
doc15.txt	0.000000000	able

Figure 4: Keywords and their TF-IDF values

Once we compute the tf-idf for all the tokens, we decide upon a threshold value for the tf-idf weight. The threshold is decided such that we eliminate most of the stopwords and other unimportant keywords.

In this project, we chose the tf-idf threshold to be 0.001. So the keywords with the tf-idf value below 0.001 were eliminated. Thus, the remaining list of tokens consisted of high frequency keywords. The below figure shows the list of high frequency keywords:



doc1.txt	0.001044624	accomplishing
doc1.txt	0.001044624	appreciated
doc1.txt	0.001044624	ascribe
doc1.txt	0.001044624	atial
doc1.txt	0.001044624	atomic
doc1.txt	0.001044624	attacked
doc1.txt	0.001044624	attribution
doc1.txt	0.001044624	bollabas
doc1.txt	0.001044624	boosts
doc1.txt	0.001044624	came
doc1.txt	0.001044624	cles
doc1.txt	0.001044624	clumps
doc1.txt	0.001044624	coloration
doc1.txt	0.001044624	coloring
doc1.txt	0.001044624	compt
doc1.txt	0.001044624	conservative
doc1.txt	0.001044624	convenient
doc1.txt	0.001044624	conveniently
doc1.txt	0.001044624	creature
doc1.txt	0.001044624	crude
doc1.txt	0.001044624	crystals
doc1.txt	0.001044624	decompose
doc1.txt	0.001044624	decomposed
doc1.txt	0.001044624	decomposution

Figure 5: High frequency keywords

The below figure shows the list of high frequency keywords for one of the documents:

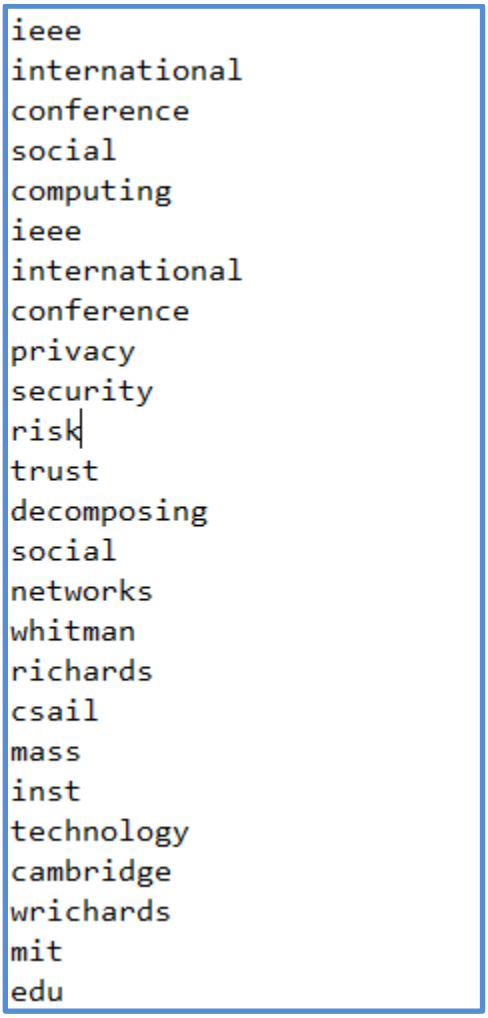


Figure 6: High frequency tokens in one of the documents

After finding high frequency keywords, we find keyword pairs. High frequency keywords that are 30 positions apart are considered and are paired together. Below figure shows the keyword pairs formed from the high frequency keywords. The first column represents the document frequency for each pair:

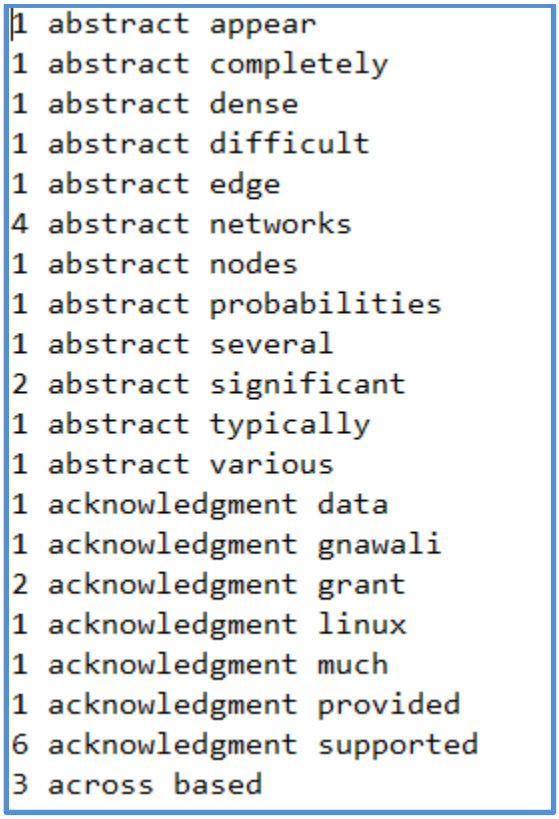


Figure 7: Keyword pairs with their df (document frequencies)

On applying the Apriori Principle, we select a threshold value to eliminate the keyword pairs that are less important. We selected the df (document frequency) of 3 to remove the keyword pairs that are least important. The figure below shows high frequency keyword pairs that are having the df > 2:

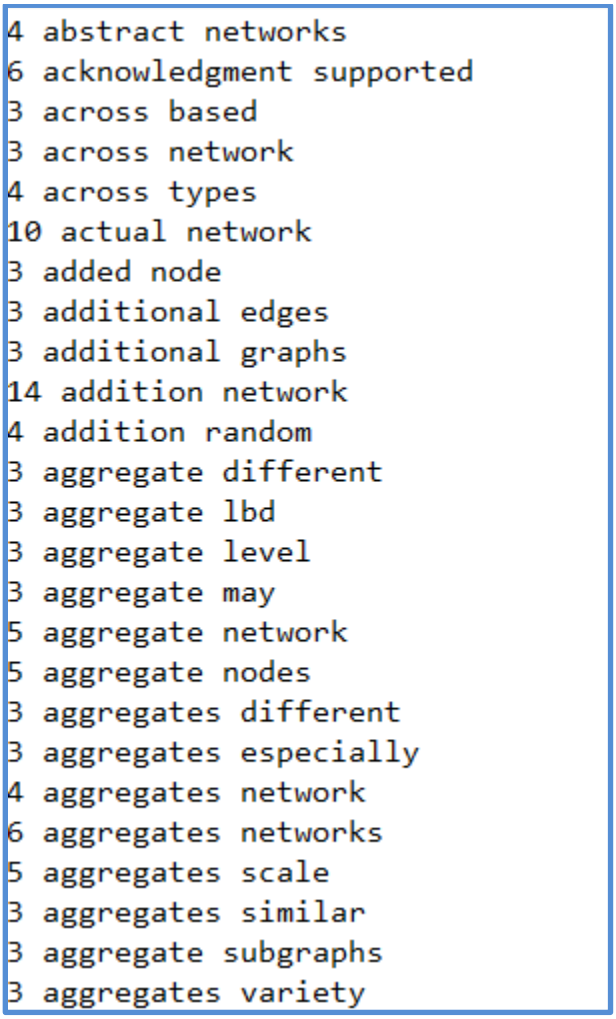


Figure 8: High frequency keyword pairs

Three keywords are formed by associating high frequency two-keywords. On applying Apriori Principle, we get high frequency three-keywords. The below snapshot shows a list of high frequency three-keywords:

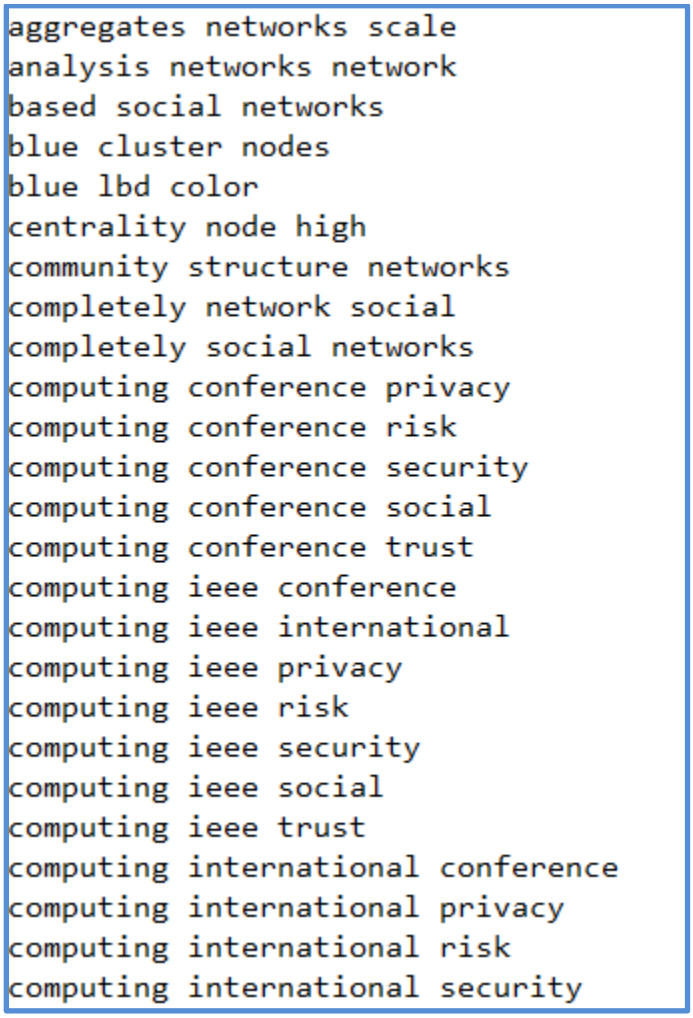


Figure 9: High frequency three-keywords

Primitive concepts are keywords that do not grow further. For three-keywords, the below figure shows the primitive concepts:

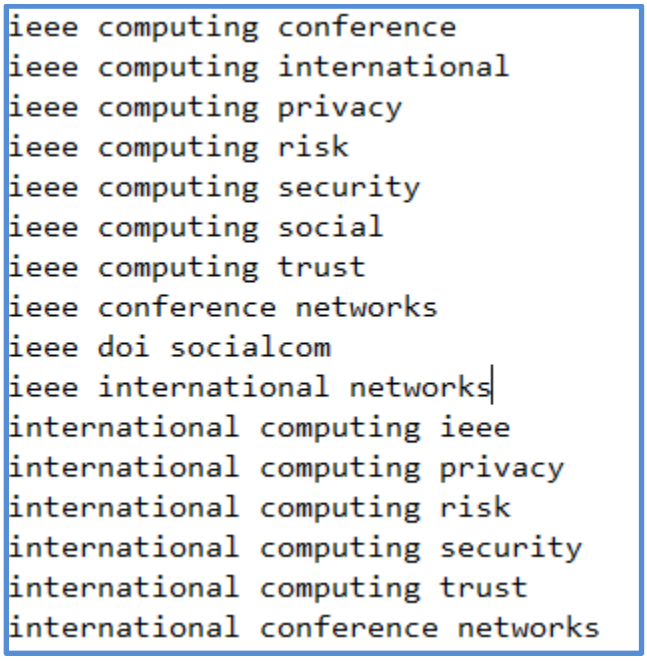


Figure 10: Primitive Concepts for three-keywords

It can be noticed that the above list of three-keyword primitive concepts did not occur in any of the four or more keyword lists. The next sets of keywords are formed by finding an association with the non-primitive keyword list.

The below figure shows the list of high frequency 4-keywords:

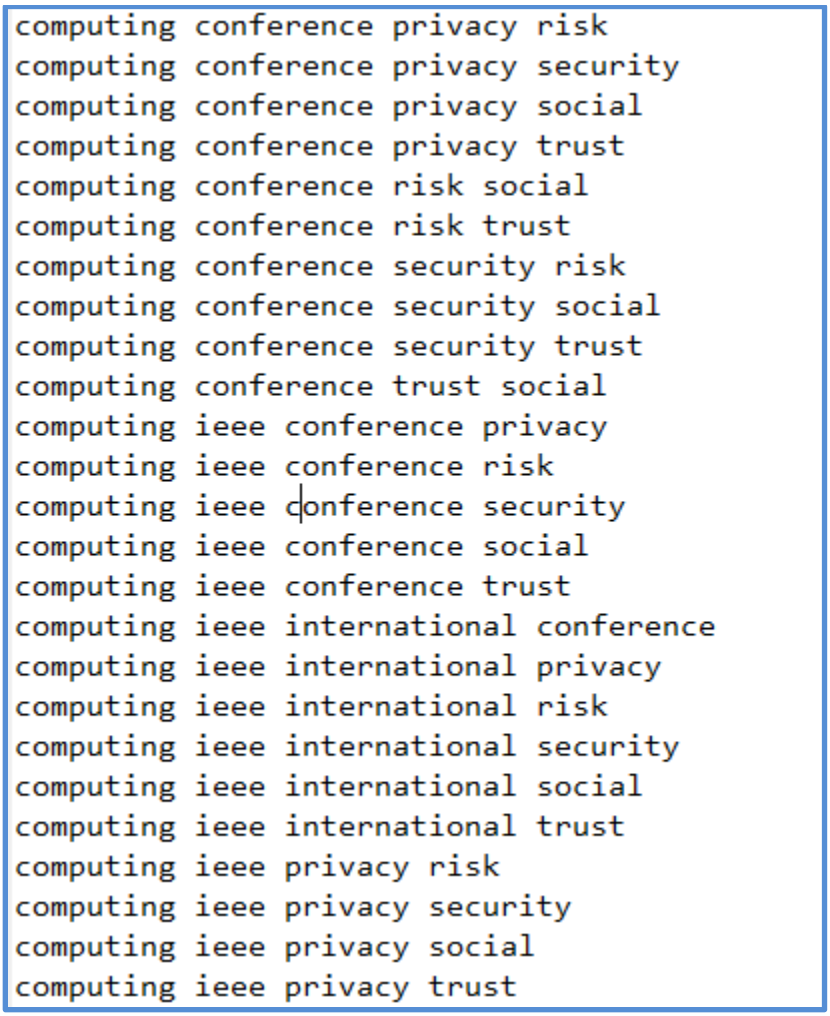


Figure 11: High frequency four-keywords

The above list of keywords was created by finding associations in the high frequency three-keyword list. On applying the apriori principle, we eliminate the low frequency four-keywords, thus leaving behind a list of high frequency four-keywords.

The below figure shows a list of the longest keyword sequences found in our sample data of 21 documents:

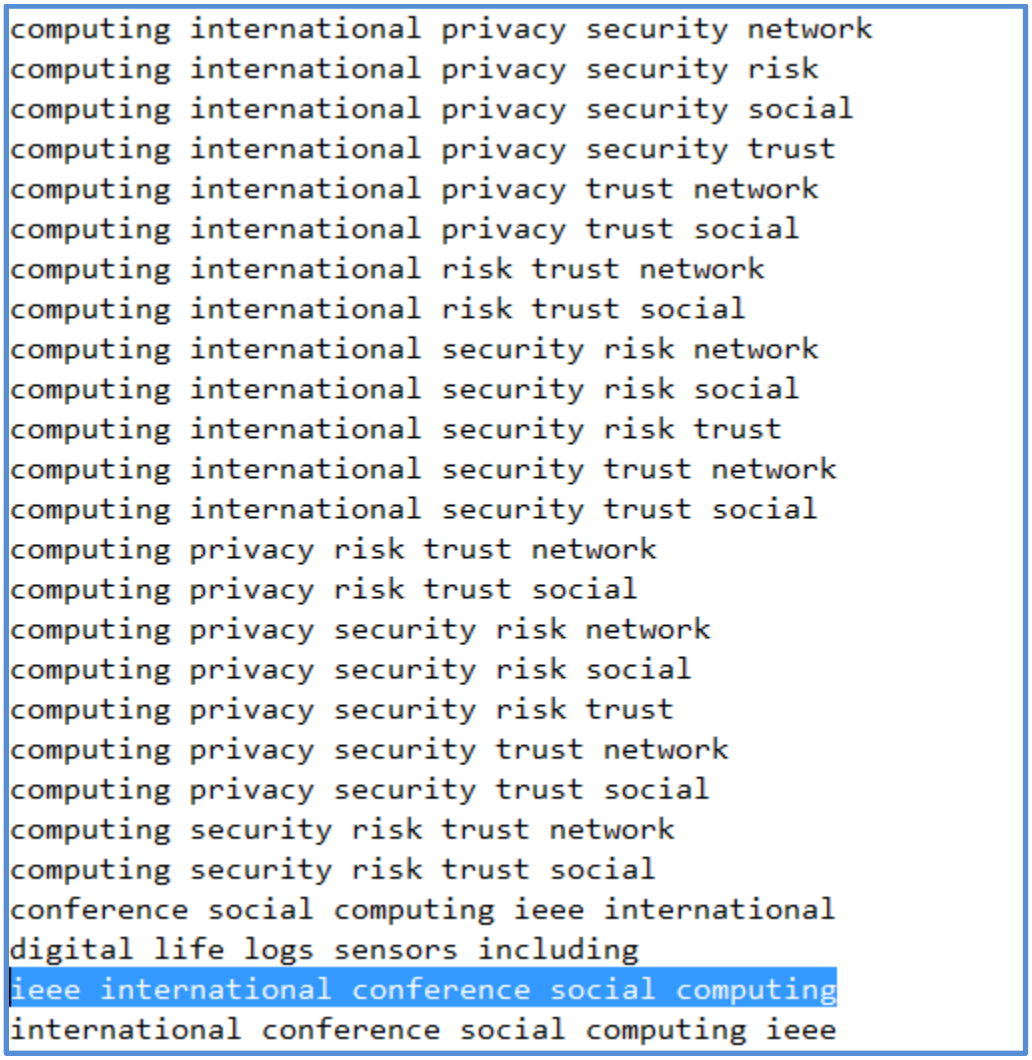


Figure 12: Longest sequence of high frequency keywords – Primitive Concept

This list of keywords did not grow further and thus did not result in any six-keyword associations.

8 Advantages and drawbacks

Our concept of finding the knowledge structure through the use of Association Rule and Apriori principle is primarily intended to create a pre-computed web search index for a semantic search engine. From our experimentation results, we observe that the results obtained contain the semantic meaning of the document. Creating a search index by this concept, the index would consist of highly meaningful and relevant search terms. A search engine using this index for its searching would thus provide search results that are highly relevant to the search string.

However there are some drawbacks too. While on one hand there is an advantage with the search results being relevant to the search terms, the computation time involved in creating the search index is high. The time required in identifying the associations at every stage is expensive. To overcome this, the search index must be pre-computed.

Identifying the knowledge structure in any set of web document has several applications, depending on where it is put to use. This concept could also be applied to Web Usage Mining, where we extract valuable information from web server logs, which contain the user's browsing history. Extracting the knowledge structure from these logs, we can get to know the users Internet browsing trend.

One drawback in the case of mining through web log files is that we might end up collecting data that could be personal information of the users. The most criticized ethical issue involving web mining is the invasion of privacy. (9)

9 Conclusion and future scope

From our study, we observe that using the concepts of data mining on a set of data, we get results that are very meaningful. This concept could be extended and applied to creation of a pre-computed inverted index, used in building a semantic search engine.

10 Bibliography

1. **Dertat, Arden.** *How to Implement a Search Engine Part 3: Ranking tf-idf.* July 17, 2011.

2. **Manning, Christopher D., Raghavan, Prabhakar and Schütze, Hinrich.** *Introduction to Information Retrieval.* s.l. : Cambridge University Press, 2008. 0521865719.

3. **Hegland, Markus.** *The Apriori Algorithm - The Tutorial.* Canberra : s.n., 2005.

4. **Suh, Sang C.** *Practical Applications of Data Mining.* s.l. : Jones & Bartlett Learning; 1 edition , 2011.

5. *Mining Association Rules between Sets of Items in Large Databases.* **Agrawal, R., Imielinski, T. and Swami, A.** 1993. SIGMOD Conference. pp. 207-216.

6. **Power, D. J.** Ask Dan! - What is the "true story" about data mining, beer and diapers? *DSS News.* November 10, 2002, p. 23.

7. Association rule learning. *Wikipedia.* [Online] 11 19, 2011. http://en.wikipedia.org/wiki/Association_rule_learning.

8. **Lovin, Radu.** Data Mining 101 - Part 3. *Data Mining 101.* [Online] [Cited: 11 19, 2011.] http://www.dataminingarticles.com/closed-maximal-itemsets.html.

9. Web Mining. *Wikipedia.* [Online] October 29, 2011. http://en.wikipedia.org/wiki/Web_mining.