CONCEPT INTEGRATION USING EDIT DISTANCE AND N-GRAM MATCH

Vikram Singh, Pradeep Joshi and Shakti Mandhan

National Institute of Technology, Kurukshtra, Haryana, India

ABSTRACT

Information is growing more rapidly on the World Wide Web (WWW) has made it necessary to make all this information not only available to people but also to the machines. Ontology and token are widely being used to add the semantics in data processing or information processing. A concept formally refers to the meaning of the specification which is encoded in a logic-based language, explicit means concepts, properties that specification is machine readable and also a conceptualization model how people think about things of a particular subject area. In modern scenario more ontologies has been developed on various different topics, results in an increased heterogeneity of entities among the ontologies. The concept integration becomes vital over last decade and a tool to minimize heterogeneity and empower the data processing. There are various techniques to integrate the concepts from different input sources, based on the semantic or syntactic match values. In this paper, an approach is proposed to integrate concept (Ontologies or Tokens) using edit distance or n-gram match values between pair of concept and concept frequency is used to dominate the integration process. The proposed techniques performance is compared with semantic similarity based integration techniques on quality parameters like Recall, Precision, F-Measure & integration efficiency over the different size of concepts. The analysis indicates that edit distance value based interaction outperformed n-gram integration and semantic similarity techniques.

KEYWORDS

Concept Integration, Ontology Integration, Ontology Matching, N-Gram, Edit Distance, Token, Concept Mining.

1. INTRODUCTION

Data mining is a process of extract the utilizable data from divergent perspective. Data mining also called as data or knowledge discovery [1]. Data mining provides the different kind of mining techniques for gathering, grouping and extracting the information from substantial amount of data. Technically, data mining is a process of providing correlation or patterns between numbers of existing fields in relational database. Existing data is processed, the processed data is known as information. The processing of data is achieved through establishing some correlation among data items or patterns. Data mining is a special kind of data processing which established the fact that knowledge is always application-driven [8].

Data mining is an important aspect of knowledge discovery (KDD) in the database. There are various internal steps involves in KDD, e.g. Data selection, Data cleaning, Data transformation Data mining & Interpretation, as shown in figure 1.

DOI : 10.5121/ijdms.2014.6601

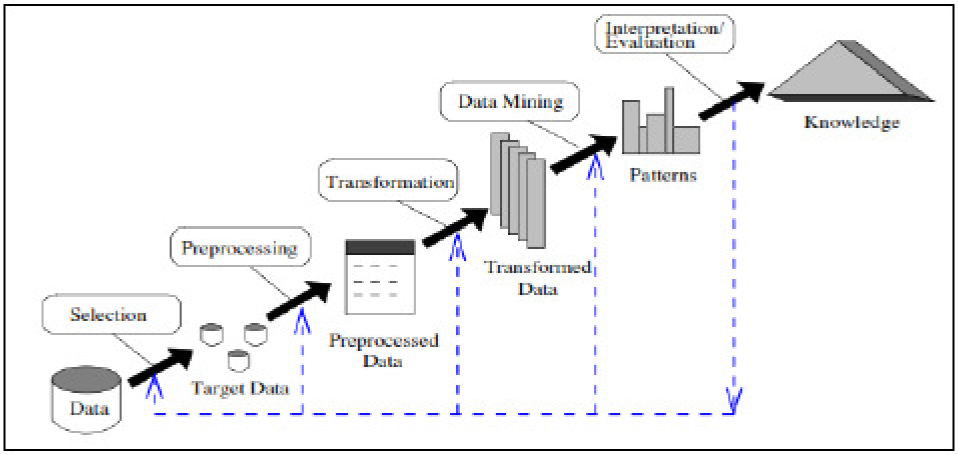


Figure 1-Data Mining in Knowledge Discovery (KDD) [1]

Ontologies are metadata schemas, providing a controlled vocabulary of concepts, each with an explicitly defined and machine process able semantics [1][7], by defining shared and common domain theories, ontology helps both people and machines to communicate precisely to support the exchange of semantics. Ontology language editors help to build semantic web [3][4][5]. Hence, the contemptible and efficient construction of domain specific ontology is crucial for the success of many data processing systems. In many data or information processing systems term ontology is refer as token or concept as well, as token & ontology refers to a term/word which represent a set of values, meaning, knowledge & both are identified based on the given input data, document, text etc[20][26]. In our approach "Token" or "Ontology" both are referred as term "Concept" for simplification on representation of proposed approach.

Concept enables the abstraction various data domains [23]. "Token/Ontology both provide the vocabulary of concepts that describe the domain of interest and a specification meaning of terms used in the vocabulary" [8]. In modern scenario, as data is growing rapidly, the main problem lies in the heterogeneity between that data and to integrate the data, so that heterogeneity can be minimize [6]. Concept integration plays an important role in minimizing heterogeneity among data items. Concept integration consists of various steps like Concept matching & Concept mapping [23]. Concept matching is a process that measures the similarity of attribute between these concepts and provides a better result for concept integration.

A complete lifecycle of a concept is shown in Figure 2, it describes step-by-step flow of activities [9], starting from concept identification to storing & sharing of concept. Matching through characterising the problem (identify the problem), selecting the existing alignment, selecting the appropriate matchers, running the matchers and select the appropriate results and correcting the choices made before (matchers, parameters), documenting and publishing good results and finally using them. The concept matching process is utilized to measures the homogeneous attribute between the two set of concepts [9].

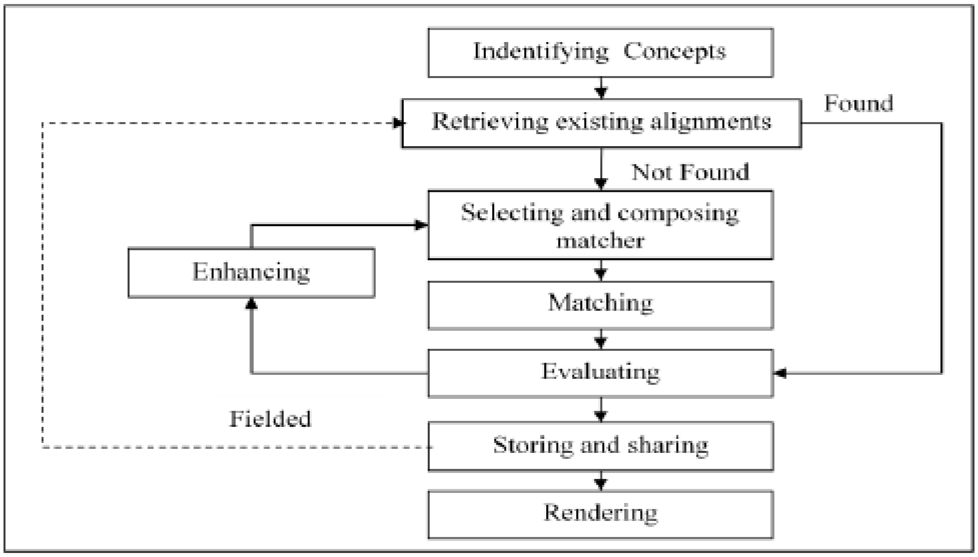


Figure 2: Concept (Token/Ontology) life cycle [9]

In the area of data and information science, concept is a formal framework for representing domain knowledge [19][20]. This framework primarily defines the semantics of data element of domain & then identifies the relationship among other. Concept identification is an important part of any token/Ontology integration system [21][22]; to identify concept preprocessing of input document/text is required [10]. Domain specific ontologies firstly identified for different sources of information/ document [11] [12]. Text and then merges into single set of concept. In concept integration, there are two activities involved like token/Ontology identification & Token/Ontology matching [6] [15]. Concepts are a vital component of most knowledge based applications, including semantic web search, intelligent information integration, and natural language processing. In particular, we need effective tools for generating in-depth ontologies that achieve comprehensive converge of specific application domains of interest, while minimizing the time and cost of this process. Therefore we cannot rely on the manual or highly supervised approaches often used in the past, since they do not scale well.

In the field of artificial intelligence, data mining, data warehousing, semantic web, systems engineering, software engineering, biomedical informatics, library science, enterprise bookmarking, and information architecture[2].

2. TOKEN/ONTOLOGY INTEGRATION

Concept identification (token/Ontology extraction, token/Ontology generation, or token/Ontology acquisition) is the automatic or semi-automatic creation of ontologies, including extracting the corresponding domain's terms and the relationships between those concepts from a corpus of natural language text, and encoding them with an token/Ontology language for easy retrieval [23]. As building ontologies manually is extremely labor-intensive and time consuming, there is great motivation to automate the process [15] [24]. Concept matching plays critical role in concept integration, each source concept is matched with each target concept based on the some matching function. In proposed approach the matching between source & target concept is based on edit-distance value or n-gram value. Finally, concept Integration, the various concepts are merged into single set of concept. By introducing concepts and their relations, ontologies provide a critical and necessary information structure that facilitates the processes of sharing, reusing, and analyzing domain knowledge in Semantic Web and other knowledge based systems [17][18].

2.1 Motivation

Information/data integration has a wide range of application through token/Ontology integration. The integration of data and integration of schema has been attracted wide interest of researcher from research area like information retrieval, data mining & warehousing, Query processing, systems engineering, software engineering, biomedical informatics, library science, enterprise bookmarking etc. Ontology integration explains the process and the operations for building ontologies from other ontologies in some Ontology development environment. Ontology integration involves various methods that are used for building the ontologies using other set of Ontology [9].

The First motivation behind the Ontology integration is to use of multiple ontologies. For example: - suppose we want to build Ontology of tourism that contains information about transportation, hotels and restaurants etc. so we can construct this Ontology from initial but this take lot of efforts when the ontologies are huge. Ontology reusing is a concept of Ontology reuse, we can utilize previously created Ontology (already exist) on topics transportation, hotels and restaurant to build desired Ontology for tourism. These ontologies may share some entities, concepts, relations and consequently. The second motivation is the use of an integrated view. Suppose a university has the various colleges affiliated to that university across the world. University needs information from the colleges about the faculty, academic etc. In this case, university can query the ontologies at various colleges through proper Ontology mappings, thus providing a unified view to the university. The third motivation is the merge of source ontologies. Suppose various ontologies are created on the same topic or concept and overlapping the information. Ontology merging is used to merge these ontologies and build a single Ontology, which consists various concepts, entities definitions from the local ontologies, for example, suppose several car companies are merged into a new car company, for which Ontology has constructed. This could be done by merging the existing ontologies of these companies.

2.2 Proposed Procedure & Example

For a given text/document, a document heap is created based on the tokens/ontologies frequency within input documents. A heap is a specialized tree-based data structure that satisfies the heap property: If A is a parent node of B then the key of node A is ordered with respect to the key of node B with the same ordering applying across the heap. Proposed algorithm consists of three activities for token/ontology integration mentioned below and schematic diagram is shown in figure 3.

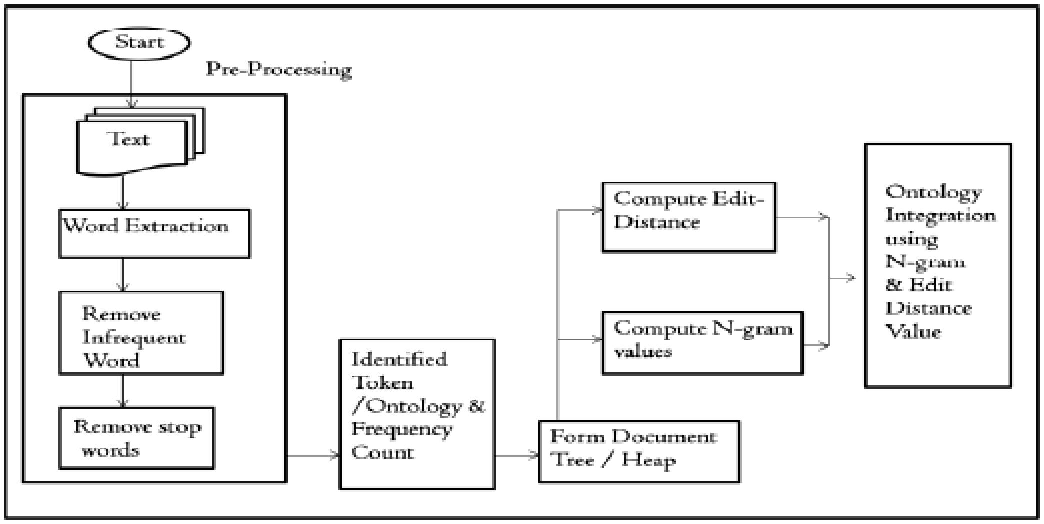

Start
Pre-Processing
Text
Word Extraction
Remove Infrequent Word
Remove stop words
Identified Token /Ontology & Frequency Count
Form Document Tree / Heap
Compute Edit-Distance
Compute N-gram values
Ontology Integration using N-gram & Edit Distance Value

Figure 3: Schematic diagram of proposed method

Step 1: Token/Ontology Identification and Construction of Token/Ontology Heap:-First step involves two important activities, firstly pre-processing of input document/text is done for the purpose of token/Ontology detection, in this step word extraction, stop word removal & stemming are applied to indentify all possible Ontology in input document/ text. Another activity is to compute the frequency of each of the Ontology within document, frequency simply represent the number of appearance of the Ontology in the document or paragraph. The term frequency is used to construct the heap (max heap) for respective document, in which the Ontology with highest frequency appears on the top of heap. Similarly heaps are constructed for each of the document or the text document.

Step 2: Computation of Edit Distance and N-Gram match values [19]: For each pair of Concepts, edit-distance and n-gram matching values are to be calculated. The constructed heap's in step 1 are the input for this step and for each pair of concepts from participating heaps edit distance and n– grams value is been computed. The computed matching values are stored in 2-dimentional array and used in next step during the integration of the heaps.

The edit distance between pair of token/Ontology determines the number of character modifications (additions, deletions, insertions) that one has to perform on one string to convert it to the second string. The n-grams of the two element name strings are compared. An n-gram is a substring of length n and the similarity is higher if the two strings have more n-grams in common.

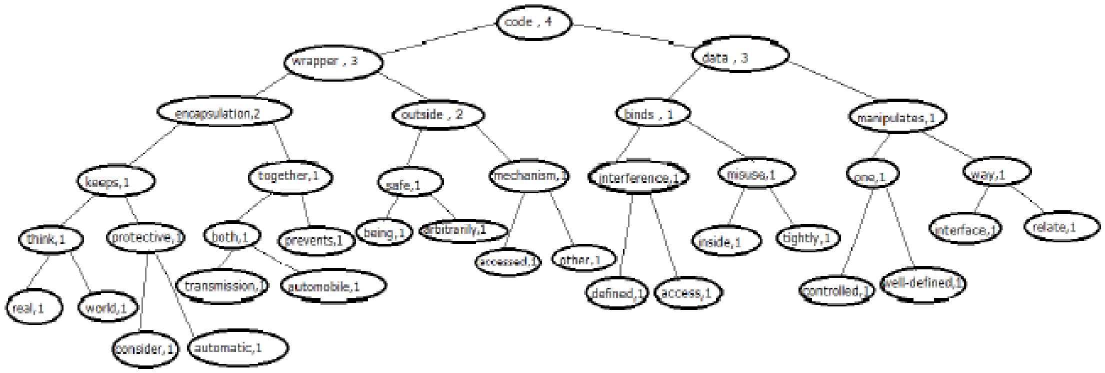
Table 1: Edit Distance & N-Gram value of Ontology Pair

Ontology Pair	**Edit Distance**	**N-Gram**
(RESP, RESPONSIBILITY)	The number of editing changes that needs to convert one of these strings to the other is 10 either add the characters 'O', 'N', 'S', 'I', 'B', 'I', 'L', 'I', 'T', 'Y', to RESP or delete the same characters from RESPONSIBILITY. Thus the ratio of the required changes is 10/14, edit distance between these two strings; **1-(10/14) = 4/14 = 0.29**	Let n = 3, means 3-grams. The 3-grams of RESP are 'RES' and 'ESP'. Similarly, there are twelve 3-grams of RESPONSIBILITY: 'RES', 'ESP', 'SPO', 'PON', 'ONS', 'NSI', 'SIB', 'IBI', 'BIP', 'ILI', 'LIT', and 'ITY'. There are two matching 3-grams out of twelve, giving a 3-gram similarity of **2/12 = 0.17**

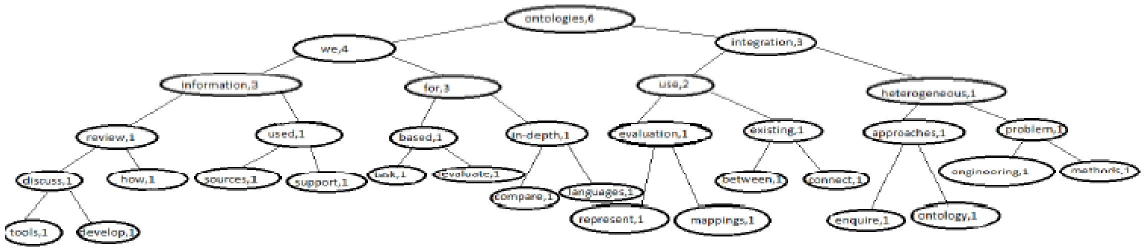
Step 3: Concept Heap Merging/Integration: next step to integrate the various heaps-for integration/merging, firstly algorithm decides the dominating heap from the participating heaps. The heap with highest values of concept frequency become the dominating among the pair of heaps and will play as the basis for the integration process, other participating heaps are merged into the dominating heap during the integration/merging process. Integration of the merged nodes position heaps base on the edit distance of n-gram matching value between pair of ontologies from pair different heaps, eg. O_{ii} of H_i is integrated with O_{jk} of H_j, which has highest edit distance or highest n grams matching values. The resultant heap will retain both Ontology in the node and position of the node is determined on basis of best position among participated ontologies (O_{ii}, O_{jk}). The integration results into creation of merged node and best position for newly created will be based on highest values of frequency among participating concept.

Example

Input 1: *"Encapsulation is the mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse. One way to think about encapsulation is as a protective wrapper that prevents the code and data from being arbitrarily accessed by other code defined outside the wrapper. Access to the code and data inside the wrapper is tightly controlled through a well-defined interface. To relate this to the real world, consider the automatic transmission on an automobile. It encapsulates hundreds of bits of information about your engine, such as how much you are accelerating, the pitch of the surface you are on , and the position of the shift lever"* .



Input 2: *"We review the use on ontologies for the integration of heterogeneous information sources. Based on an in-depth evaluation of existing approaches to this problem we discuss how ontologies are used to support the integration task. We evaluate and compare the languages used to represent the ontologies and the use of mappings between ontologies as well as to connect ontologies with information sources. We also enquire into ontology engineering methods and tools used to develop ontologies for information integration"*.



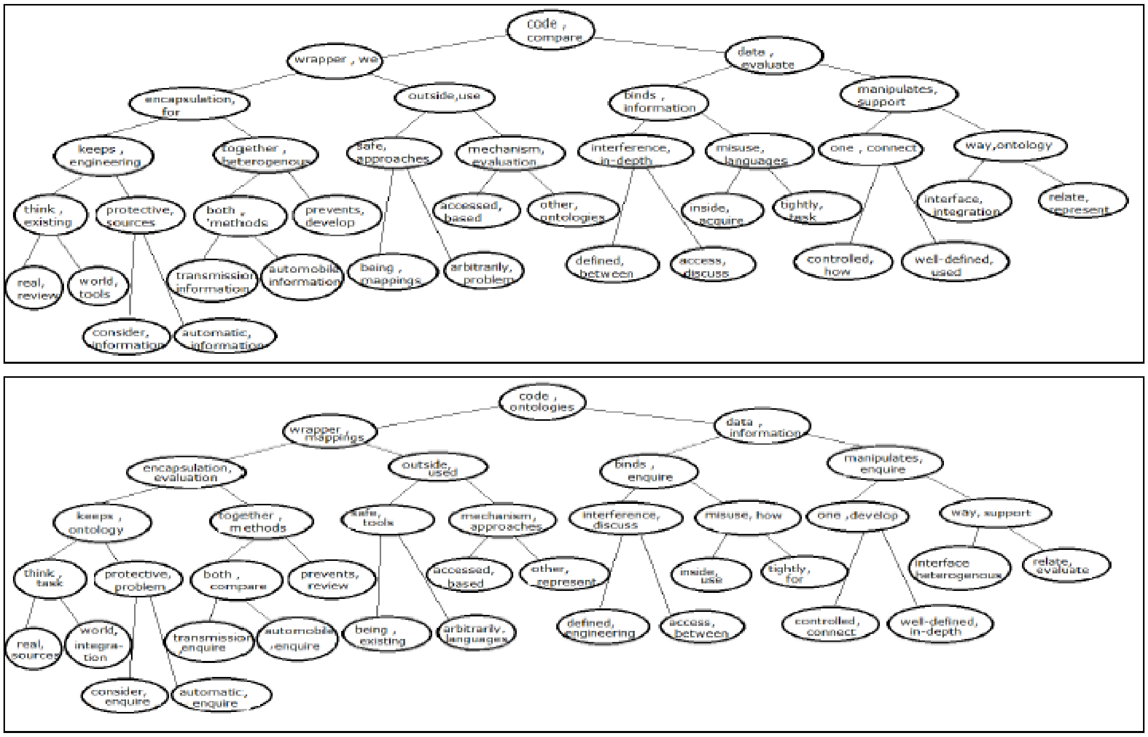
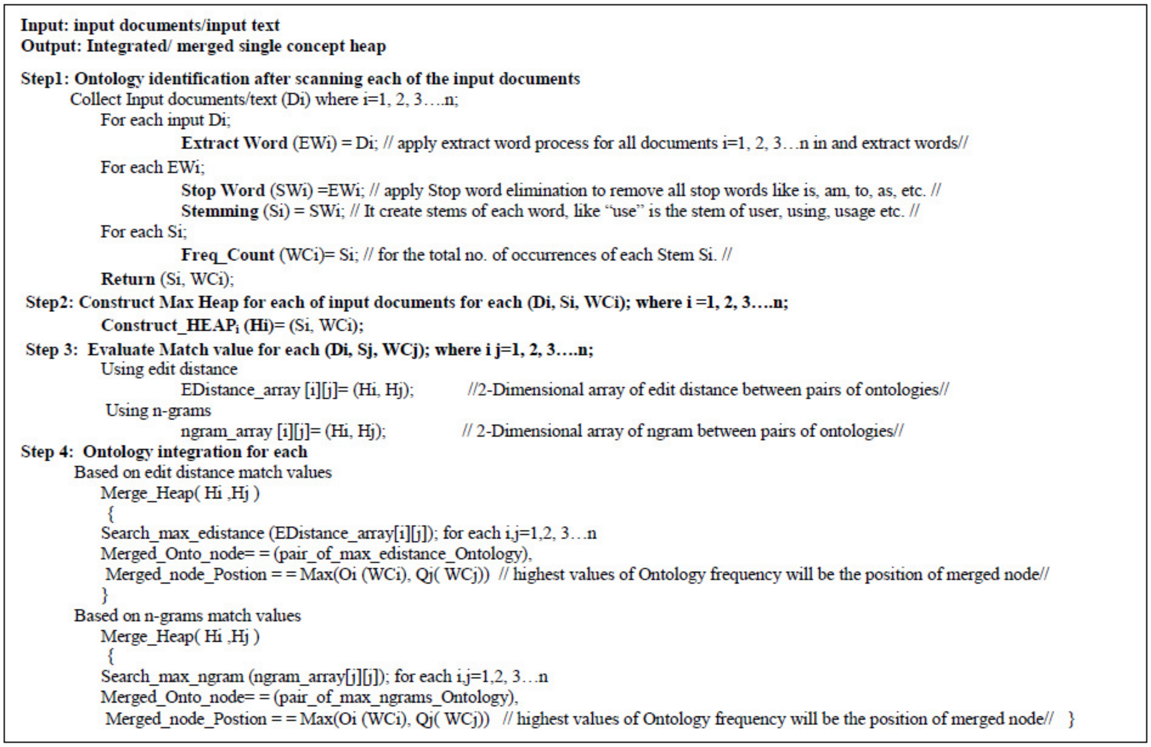


Figure 4: Integrated/Merged Concept of Input 1 & Input 2 using (a) edit distance technique (b) n grams technique

As shown in example, the pair of ontologies form different document heap, the matching values calculated and integrated trees are formed. In the paper Ontology integration based on the edit distance and n-gram values has been done. The performance analysis for both approach are based on the parameters like Precision, Recall, F-measure and efficiency of the approach. Precision, recall & f-measure indicate the quality of match and quality of integrated Ontology and efficiency parameters represent the execution efficiency to generate & integrate the ontologies form various source ontologies.

Precision is a value in the range [0, 1]; the higher the value, the fewer wrong merging computed [5][11]. Precision is determines as the ration of number of correct found alignment/ matching with total number of alignment found. Recall is a value in the range [0, 1]; the higher this value, the smaller the set of correct mappings which are not found. The ratio of number of correct found alignment with total number of alignment. The F-measure is a value in the range [0, 1], which is global measure of matching quality. F-Measure used the mean of precision and recall [11]. The values for F-measure is computed by '2*Precision*Recall' with ratio of 'Precision*Recall'. The comparison graph between three methods, e.g Semantic Similarity based, Edit distance based & N-Gram based integration techniques are shown over the range of different values of Precision, Recall & F-Measure, in figure 5. The graph depicts edit distance based techniques as the winner among three, as integration of concept are having better recall, precision & f-measure values.

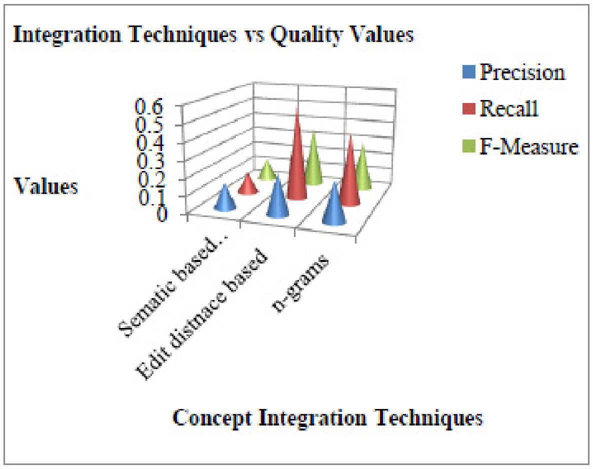
2.3 Algorithm



```
Input: input documents/input text
Output: Integrated/ merged single concept heap

Step1: Ontology identification after scanning each of the input documents
    Collect Input documents/text (Di) where i=1, 2, 3….n;
        For each input Di;
                Extract Word (EWi) = Di; // apply extract word process for all documents i=1, 2, 3…n in and extract words//
        For each EWi;
                Stop Word (SWi) =EWi; // apply Stop word elimination to remove all stop words like is, am, to, as, etc. //
                Stemming (Si) = SWi; // It create stems of each word, like "use" is the stem of user, using, usage etc. //
        For each Si;
                Freq_Count (WCi)= Si; // for the total no. of occurrences of each Stem Si. //
        Return (Si, WCi);
 Step2: Construct Max Heap for each of input documents for each (Di, Si, WCi); where i =1, 2, 3….n;
        Construct_HEAPi (Hi)= (Si, WCi);
 Step 3: Evaluate Match value for each (Di, Sj, WCj); where i j=1, 2, 3….n;
        Using edit distance
                EDistance_array [i][j]= (Hi, Hj);        //2-Dimensional array of edit distance between pairs of ontologies//
         Using n-grams
                ngram_array [i][j]= (Hi, Hj);            // 2-Dimensional array of ngram between pairs of ontologies//
Step 4: Ontology integration for each
    Based on edit distance match values
        Merge_Heap( Hi ,Hj )
         {
        Search_max_edistance (EDistance_array[i][j]); for each i,j=1,2, 3…n
        Merged_Onto_node= = (pair_of_max_edistance_Ontology),
         Merged_node_Postion = = Max(Oi (WCi), Qj( WCj))  // highest values of Ontology frequency will be the position of merged node//
        }
    Based on n-grams match values
        Merge_Heap( Hi ,Hj )
         {
        Search_max_ngram (ngram_array[j][j]); for each i,j=1,2, 3…n
        Merged_Onto_node= = (pair_of_max_ngrams_Ontology),
         Merged_node_Postion = = Max(Oi (WCi), Qj( WCj))   // highest values of Ontology frequency will be the position of merged node//   }
```

In figure 6, effect of ontology length over the overall efficiency of integration techniques are depicted. For ontology length 7 & 8, both edit distance and n-gram based integration method are close on their efficiency while semantic similarity based techniques is outperformed by the both techniques. Finally, in figure 7 is for comparative analysis is depicting the performance (quality values delivered) comparison between edit distance techniques & n-gram techniques while integrating ontology length. The comparison is performed under range of ontology length and quality parameters values are kept in the observation. The overall performance of edit distance techniques is consistent and significant performance is delivered on ontology of length 7 or 8 while for n-gram integration techniques the better result delivered for the ontology of length 6 or 5. Few conclusion from the experimental analysis is drawn like, edit distance perform better than n-gram & semantic similarity based integration techniques for different size of ontology. The edit distance technique performs better and shows potential to carry good values of all quality parameters, which affects the quality of results during processing.



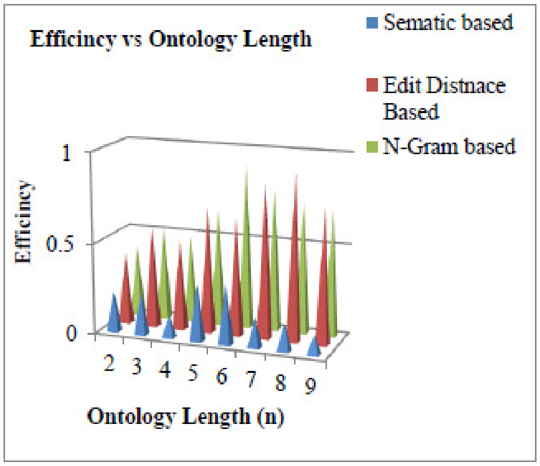


Figure 5: Concept Integration technique vs Quality values

Figure 6: Concept Integration Technique efficiency vs Ontology Length

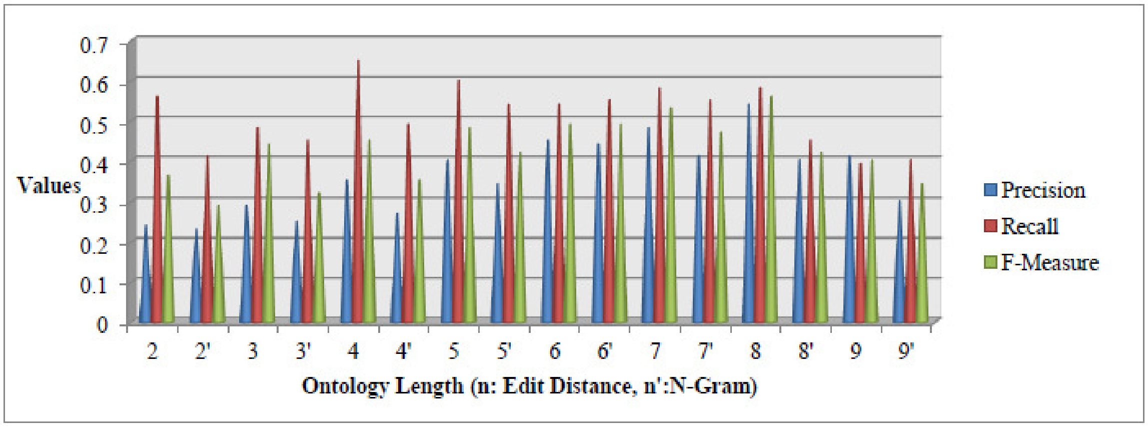


Figure 7: Quality values vs Ontology length on edit distance & n-gram technique

3. CONCLUSION

In the area of data and information science, token/Ontology is a formal framework for representing domain knowledge. This framework primarily defines the semantics of data element of domain & then identifies the relationship among other. Information/data integration has a wide range of application through token/Ontology integration. The integration of data and integration of schema has been attracted wide interest of researcher from research area like information retrieval, data mining & warehousing, Query processing, systems engineering, software engineering, biomedical informatics, library science, enterprise bookmarking, schema integration, E-R diagram integration, Graph integration (web semantic based graph) etc. Ontology integration explains the process and the operations for building ontologies from other ontologies in some Ontology development environment. There are various existing techniques for ontology integration, in this paper an approach is proposed for the ontology integration using match values based on the edit distance & n-gram. Edit distance determines match values among pair of concepts based on the changes required in participating concepts, in order to align them. In case of n-gram method, the matches are the count of n-length substrings of participating ontology are matching. Concept integration using both methods are implemented on wide range of input document/text. The performance comparison is through over the existing method, Semantic similarity based with edit distance and n-grams method is done. The ontology length has proportional effect on overall efficiency of the techniques, as ontology of length 6 to 8 edit distances outperform all other integration techniques while for smaller length of ontology n-gram & semantic similarity perform better. Few conclusion from the experimental analysis is drawn like, edit distance perform better than n-gram & semantic similarity based integration techniques for different size of ontology. The edit distance technique performs better and shows potential to carry good values of all quality parameters, which affects the quality of results during processing.

REFERENCES

[1] J. Han, M.Kamber, and J. Pai, "Data Mining Concepts and Techniques", Morgan Kaufmann Publishers, USA, Third Edition, Elsevier, 2011.

[2] D. Kumar, D. Bhardwaj, " Rise of Data Mining: Current and Future Application Areas", International Journal of Computer Science Issues", vol. 8, issue 5, p.p 256-260, 2011

[3] Berners-Lee T., Handler J., and Lassila O, "The Semantic Web", Scientific American, May 2001.

[4] Jiang Huiping, " Information Retrieval and the Semantic Web", IEEE International Conference on Educational and Information Technology (ICEIT), Chongqing, China, Vol No. 3, p.p 461-463, 2010.

[5] Vikram Singh, Balwinder Saini, An Effective Tokenization Algorithm For Information Retrieval Systems", in the 1st international Conference on data Mining, DMIN-2014, Banglore, 2014, pp. 109–119.

[6] F. Giunchiglia, M. Yatskevich, and P. Shvaiko, "Semantic matching: Algorithms and implementation," Journal on Data Semantics, Vol. No. 9, pp. 1–38, 2007.

[7] Ontology Components: http://en.wikipedia.org/wiki/Ontology_components.

[8] Pavel Shvaiko and Jerome Euzenat, "Ontology matching: state of the art and future challenges", IEEE Transactions on Knowledge and Data Engineering, Vol. No. 25, Issue: 1 , p.p 158-176, 2013.

[9] J. Euzenat and P. Shvaiko, "Ontology matching", Springer, 2013.

[10] C.Ramasubramanian and R.Ramya, Effective Pre-Processing Activities in Text Mining using Improved Porter's Stemming Algorithm, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 12, December 2013.

[11] AnHai D, Jayant M, Pedro D et al., "Learning to map between ontologies on the semantic web", Eleventh International World Wide Web Conference, Honolulu Hawaii, USA, 2005.

[12] Tordai, A, "On combining alignment techniques" PhD thesis, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands , pp. 65, 193, 2012.

[13] A Rodrfguez, M Egenhofer, "Determining Semantic Similarity Among Entity Classes from Diferent Ontologies" , IEEE Transactions on Knowledge and Data Engineering, Vol. No.15 (2), 442-456, 2003.

[14] D. Aum, H.-H. Do, S. Mabmann, and E. Rahm, "Schema and Ontology matching with COMA++," in Proceeding 24th International Conference on Management of Data (SIGMOD), Demo track, pp. 906–908, 2005.

[15] Bin Ye, Hongyu Chai, Weiping He , Xiaoting Wang , Guangwei Song, "Semantic similarity calculation method in Ontology mapping", 2nd International Conference on Cloud Computing and Intelligent Systems (CCIS), Hangzhou, Vol No. 3, p.p 1259-1262, 2012.

[16] Hartung, M., Grob, A., Rahm, E, " COntoDiff: generation of complex evolution mappings for life science ontologies", J. Biomed. Inform. 46(1), p.p 15–32, 2013.

[17] Kirsten, T., Groß, A., Hartung, M., Rahm, E, " GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution", J. Biomed. Semant, 2011.

[18] Doan, A., Halevy, A., Ives, Z, " Principles of Data Integration", Morgan Kaufmann San Mateo. 497 pp. 5, 73, 2012.

[19] M. Ozasu, P.Valduriez, "Principles of Distributed Database Systems", Prentice Hall, 1991

[20] S. Shehata, F. Karray, and M. S. Kamel, "An Efficient Concept-Based Mining Model for Enhancing Text Clustering", IEEE Transactions On Knowledge And Data Engineering, Vol. 22, No. 10, October 2010.

[21] Algergawy, A., Nayak, R., Siegmund, N., Köppen, V., Saake, "Combining schema and level based matching for web service discovery",10th International Conference on WebEngineering (ICWE), Vienna, Austria, pp. 114–128, 2010.

[22] Wache, H., Voegele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, "Ontology-based integration of information—a survey of existing approaches", 17th International Joint Conference on Artificial Intelligence (IJCAI)", Seattle, WA, USA, pp. 108–117, 2001.

[23] Giuseppe Fenza, Vincenzo Loia, and Sabrina Senatore, "Concept Mining of Semantic Web Services By Means Of Extended Fuzzy Formal Concept Analysis (FFCA)," IEEE, Feb. 2008.

[24] Yang Zhe., "Semantic similarity match of Ontology concept based on heuristic rules". Computer Applications, Vol. No. 12, Dec. 2007.

[25] Li Chun-miao, Sun jing-bo, "The Research of Ontology Mapping Method Based on Computing Similarity ". Science & Technology Information, Vol. No.1, p.p 552-554, 2010.

[26] Shalini Puri, "A Fuzzy Similarity Based Concept Mining Model for Text Classification", International Journal of Advanced Computer Science and Applications, Vol. 2, No. 11, 2011