2013 International Conference on Advances in ICT for Emerging Regions (ICTer): 154 - 159

Document Analysis Based Automatic Concept Map Generation for Enterprises

E.L.Karannagoda[1], H.M.T.C.Herath[2], K.N.J.Fernando[3], M.W.I.D.Karunarathne[4], N.H.N.D.de Silva[5], A.S. Perera[6]

Department of Computer Science and Engineering, University of Moratuwa, Moratuwa, Sri Lanka

[1]ekarannagoda@virtusa.com,[2]thilinatnt.p@gmail.com,[3]jayaprabathfernando@gmail.com, [4]isuradilhara@gmail.com, [5]nisansadds@cse.mrt.ac.lk, [6]shehan@cse.mrt.ac.lk

***Abstract*— Ever growing knowledge bases of enterprises present the demanding challenge of proper organization of information that would enable fast retrieval of related and intended information. Document repositories of enterprises consist of large collections of documents of varying size, format and writing styles. This diversified and unstructured nature of documents restrict the possibilities of developing uniform techniques for extracting important concepts and relationships for summarization, structured representation and fast retrieval. The documented textual content is used as the input for the construction ofa concept map. Here a rule based approach is used to extract concepts and relationships among them. Sentence level breakdown enables these rules to identify those concepts and relationships. These rules are based on elements in a phase structure tree of a sentence. For improving accuracy and the relevance of the extracted concepts and relationships, the special features such as titles, bold and upper case texts are used.**

This paper discusses how to overcome the above mentioned challenges by utilizing high level natural language processing techniques, document pre-processing techniques and developing easily understandable and extractable compact representation of concept maps. Each document in the repository is converted to a concept map representation to capture concepts and relationships among concepts described in the said document. This organization would represent a summary of the document. These individual concept maps are utilized to generate concept maps that represent sections of the repository or the entire document repository. This paper discusses how statistical techniques are used to calculate certain metrics which are used to facilitate certain requirements of the solution. Principle component analysis is used in ranking the documents by importance. The concept map is visualized using force directed type graphs which represent concepts by nodes and relationships by edges.

Keywords— **Natural Language Processing, Concept Map, Concepts/Relationships Extraction**

I. Introduction

Knowledge bases of enterprises consist of large document repositories and they are growing continuously. This large volume of documents cover a vast amount of knowledge but the extraction of the relevant knowledge in a timely manner has become a challenging task. Therefore search engines and document summarization system have become prominent research areas. Some of the above systems are developed purely based on statistical analysis while others employ natural language processing techniques.

SigmaC is developed with the aim of solving the above issues by utilizing more human understandable concept maps for knowledge representation. Usage of natural language processing techniques allows SigmaC to overcome the limitations of pure statistical analysis and SigmaC also focuses on utilizing text formatting information and sematic data sources to optimize the results obtained from syntactic analysis.

978-1-4799-1276-6/13/$31.00 ©2013 IEEE

Outputs obtained from the system are useful in several different ways. Concept map representation of a document repository allows concept level browsing of the knowledge base and exploring more interesting relationships between concepts. Concept map representation of individual documents effectively produces summaries of those documents allowing users to get an understanding of the document without going through the entire document. Furthermore the system can be used to generate ontological data sources by providing a suitable set of documents for analysis.

II. Literature Review and Background

This section provides a brief description of the literature review done for SigmaC project. Furthermore candidate techniques, methodologies and tools identified, which can support in solving each of the sub problems that need to be solved for concept map generation from unstructured text are addressed in this paper.

These sub problems can be interrelated. These are only high level core problems. The subsequent section describing each of these sub problems will address further issues in each of these areas

Natural language processing, concept Extraction, relationship extraction, visual representation and browsing, and document processing are the main sub problems identified in SigmaC project.

Sentence boundary detection is initial stage of NLP which is of high importance in higher levels of linguistic processing methods such as parts of speech (POS) tagging and chunking. Simplest method of identifying the sentence boundary would be to split a text body into sentences using the period (“.”). Although this strategy works in many situations, it cannot be taken as an accurate measure due to the ambiguity of the period in a sentence. For example a period may occur in an abbreviation, end of a sentence or in mentioning the names with initials. Different methods for identifying sentence boundaries are suggested by researchers. The approach is suggested by David D. Palmer and Marti A. on their paper on Adaptive Sentence Boundary Disambiguation [2].

Anaphora resolution is a complicated problem in Natural Language Processing (NLP). There are various definitions in anaphora and one of them is based on the notion of cohesion. That means Anaphora can be defined as a function of cohesion which points back to some previous item. The reference is called an anaphor and the entity which it refers to is called its antecedent. The process of determining the antecedent of an anaphor is called Anaphora Resolution.

Anaphora resolution can be described as resolving what a pronoun, or a noun phrase referred to. This is also called co-reference resolution. This can be clearly explained using an example.

"Java is a programming language. It is a general-purpose, concurrent, class-based, object-oriented language."

Here the human brain can identify that 'It' means 'Java' but in computations there should be an algorithm to identify them. In that case, relevant words are tagged to resolve the anaphora.

Collocations represent multi word expressions which are conventional phrases used in communication. Three different categories were proposed by Frank Smadja[3]. Those are predicative relations, rigid noun phrases and phrasal templates.

PoS (Parts of Speech) tagging is the process of labeling words in a sentence with appropriate PoS tags. These PoS tags are possible syntactic categories that a word can belong to in its use in a particular sentence. These PoS tagged sentences can be used in a further level of parsing to generate phrase structure trees for sentences and also in concept identification.

Concept extraction from documents or individual sentences is a heavily researched area. Some of the approaches are concept identification using PoS tag signatures, frequency based key word extraction, phrase structure for concept extraction, dependency relationship based concept extraction, semantic data source based concept extraction and k-gram based method.

Relationship extraction is a task that recodes this unstructured information found in text, into a well-structured format. Common relationship extraction approaches can be categorized as supervised, bootstrapping and generic approaches.

III. METHODOLOGY

In this section, the important theoretical aspects and methodology to construct the concept map are described.

A. Pattern based Concept and Relationship Extraction

Patterns are provided to the SigmaC system as rules. This type of a rule should completely specify how to extract concepts from a sentence. The said rules are arranged and applied in the following way. The patterns are written in regular expressions that are specified for a phrase tree.

```
<relationship type="is_a">
    <pattern>
        <basePattern>
            <![CDATA["S: {NP=c1 $+ (VP=var1 <, (
            VBZ <: is & $+ (NP=c2 <, (DT=trp <: a
            <: an) & !<< NP $ > =var1)))"]]>
        </basePattern>
        <concept name="c2" ishead="false">
            <modifierPattern>
                <tregex><![CDATA[NP <, DT=trp]]></tregex>
                <tsurgeon>delete trp</tsurgeon>
            </modifierPattern>
        </concept>
        <concept name="c1" ishead="true">
            <modifierPattern>
                <tregex><![CDATA[NP <, DT=trp]]></tregex>
                <tsurgeon>delete trp</tsurgeon>
            </modifierPattern>
        </concept>
    </pattern>
</relationship>
```

Figure.1 Sample rule for the is a relationship

Each rule contains a base pattern – this pattern is applied to the sentence for matches and if there are any matching patterns are found, tree nodes which represent concepts are labeled with suitable names.

Sets of modifier patterns are defined for each label in the base pattern. These modifier patterns contain other patterns to be matched against the sub tree rooted at the node labeled as concepts. Nodes in that sub tree which require modifications are also labeled. Then a set of modification scripts are applied to these labeled nodes.

Leaves of the nodes labeled as concepts are taken after the modifications to represent concepts and are cleaned up using regular expressions, stop words and length of concepts.

B. Concept Ranking Algorithm

Concept ranking is based on the page rank algorithm. Each concept has an assigned strength value which is calculated at the optimization phase. This strength is used as the initial value of the concept in the concept map for the document.
Importance of each concept in the concept map is calculated iteratively. At each iteration the value of the concept is modified as shown in Fig. 2.

```
DO
S=sum of values of all concepts
FOR each concept C in concept map
O=total number of outgoing links from concept C
FOR each concept R related to concept C
        T=total number of links from concept C to
        concept R

        link Fraction=  T /O

        value of concept R=value of concept
        R+value of concept C*   (linkFraction /S)

        newTotalDocumentValue=sum of values of
        all concepts"
WHILE newTotalDocumentValue-S>1.0015
```

Figure.2 Concept Ranking Algorithm

C. Optimization

The optimization phase is used to optimize the results obtained at the concept and relationship extraction phases. The extracted concepts and relationships in the extractor are needed to optimize in order to get the most representative concepts of the document. A strength value is assigned for each concept and relationship in this phase. The strength is the value which measures the importance of concepts and relationships. Finally those concepts are filtered by the defined value to get the best representing concepts for the document. Then relationships between these concepts are also selected for generating the concept map for the document.

Concepts and their relationships are separately optimized. In the optimization of the concept, the first step is removing redundancy. In this phase, the morphological root of each concept is considered and all concepts are reduced to their morphological roots. Here the concepts with the same morphological root are joined to obtain optimized concepts. In joining concepts, frequencies of concepts are added and all related concepts are also joined to remove any redundancies. The morphological root is taken using WordNet[1] library. The pseudo code for concept optimizing is shown below.

```
FOReach conceptsC in Document
 SETmorphRoot to morphological root of the concept
 IF entry morphRoot exists in document
   SET the frequency of the value of the morphRoot
   concept by adding the original frequency value of
   C

   MERGE the relationship Lists in both concepts.
 ELSE
  Revise the concept with new morph root
 END IF
END
```

Figure. 3 Algorithm to evaluate concepts using morphological root

In joining concept objects, a new HashMap is used to add concepts. In adding concepts, if a concept exists with same name (morphological root of the concept) their related concepts are also merged together. Related concepts are stored in anlist of Arrays, so both array lists are merged by removing redundancies. Also the related concept name is set to its morphological root.

In the optimization phase there is a strength value assigned for each concept. The strength value is used to set an importance value to a concept (importance value is the overall measurement of the importance of a concept within the document). In assigning a strength value to a concept several properties of document are taken in to account. They are,

- Frequency of a concept in a document (F_c)
- Size of the document (No of sentences) (S)
- Number of titles in the document (N)
- Scale values of titles which the concept contains (SV)
- Number of words in concept (NWC)
- Number of words in the title which the concept contains (NWT)

The strength value should be a normalized value, because it can be used in comparing concepts in different documents. In order to get the normalized frequency value, it is divided by the size of the document. If a title contains a relevant concept, title scale, that is what percentage of the title is covered by the concept and the number of titles in the document, should affectthe strength value of the concept. The following conditions should be fulfilled to get an accurate value for the concept strength.

Concept strength – CS

- By considering concepts properties, CS is directly proportional to concept frequency,

$$CS \propto \frac{F_c}{S}$$

- By considering titles properties, CS is inversely proportional to the number of titles in the document,

$$CS \propto \frac{1}{N}$$

- CS is directly proportional to the scale values of titles which the concept is containing and NWC to NWT ratio,

$$CS \propto \sum SV \times \frac{NWC}{NWT}$$

By considering the above proportionalities, the final equation for calculating the concept strength can be obtained as below.

$$CS = k \times \frac{F_c}{S} + l \times \frac{1}{N} \times \sum SV \times \frac{NWC}{NWT}$$

k = 0.5 (Constant value)

L = 1 (Constant value)

Relationship optimization is done using the WordNet library Here the 'is_a'. andthe 'part_of' relationship are optimized b asserting them using WordNet. Also 'aso' (associated relationships are modified to 'is_a'/'part_of' relationships i such relationships exist in the WordNetlexical database WordNet library can be used to get hypernyms and hyponym of concepts to asserting 'is_a' relationships.

- *Hypernyms:* Y is a hypernym of X if X is a Y (eg programming language is a hypernym of Java)
- *Hyponyms:* Y is a hyponym of X if Y is a X (Java is a hyponym of programming language)

Above hypernym results are used for the assertion method o 'is_a' relationships.

The WordNet library can be used to get holonyms an meronyms of concepts to asserting 'part_of' relationships.

- *Holonym:* Y is a holonym of X if X is a part of Y (computer is a holonym of processor)
- *Meronym:* Y is a meronym of X if Y is a part of X (processor is a meronym of computer)

Above meronym results are used for the assertion method o 'part_of' relationships.

A strength value is assigned for each relationship to get a idea about the importance of the relationship in the relevan document. In setting strength to a relationship the frequency o the relationship (only for association relationships) in document (F_r) is considered.

Relationship strength – RS

$$RS = e^{-\left[\frac{1}{F_r}\right]}$$

- For 'aso' relationships
  - If 'aso' relationship modified to 'is_a' or 'part_of' i optimizing phase then RS=1.
  - Otherwise RS is directly proportional to concep frequency, - Here RS should be less than 1 ('is_a' an 'part_of' relationships are more important)

- For 'is_a' and 'part_of' relationships – 'is_a' and 'part_o relationships of two given concept does not get repeated in single document. Therefore the frequency of occurrence o these relationships in a single document is not considered a a measure of strength.
  - RS value is set to 1 if the relationship identified i concepts/relationships extraction phase – RS=1
  - If the relationship asserted by the optimization phas (using WordNet) the RS value is increased by 1 RS=2.

D. Document Ranking

In the repository there can be many related documents whic are related to a particular concept. It will be convenient to th user to choose a document if the documents were sorte according to the relevance with the concept. So the documer ranking algorithm is used to fulfill that task. Here th documents are sorted according to the calculated relevance t the concept. Here the parameters are only the informatio gathered from the textual content of the document. Sigma also allows theusers to sort the top 10 relevant document according to the document metadata like last modified date.

First a correlation matrix is generated by referring column as documents and rows as concepts. The values in the matri are the importance values of concepts in particular document. I the concept is not appearing in the document the value i

assigned to zero. Here the summation of the importance value of concepts in particular document is always equal to one. So these values are normalized to compare across the repository.

Here the importance value is calculated using the term frequency and the PageRank algorithm.

TABLE I
SAMPLE CORRELATION MATRIX

	Document 1	Document2	Document3	..
Concept1	0.8989	0.0000	0.1300	..
Concept2	0.0602	0.4423	0.8934	..
Concept1	0.0045	0.0022	0.0000	..
....	...	...	...	

After constructing, each row will be sorted by the importance value along with the corresponding document ID. Any sorting algorithm can be used but for this implementation, the sorting algorithm used is quick sort. Quick sort algorithm is used because of the relative low time complexity (n log (n)).

Here the top 10 documents that related to the concept are stored in the relational database along with the rank. Also another ranking value is assigned to the concept-document tuple which is calculated from the last modified date of the document. This enables us to show the most relevant documents in the latest document order.

IV. Results and Discussion

A. *Result comparison before/after resolving Anaphora.*

Resolving anaphora significantly improves the accuracy of the output. Since the input document is preprocessed and fed content, sentence by sentence to the extractor, it is unaware of the pronouns such as 'it, that, those' etc. This affects the calculations done based on the statistical analysis.

The Appendix A test sample is used to test the above scenario. Frequency values of the main concepts before resolving anaphora are shown in Table 2.

TABLE 2
INITIAL CONCEPT FREQUENCIES

Concept	Frequency
java	3
programming language	2
java virtual machine	1
java application	1
sun microsystems	1

Result after resolving anaphora- Appendix B

TABLE 3
CONCEPT FEQUENCIESAFTER RESOLVING ANAPHORA

Concept	Frequency
java	4
programming language	3
java application	2
java virtual machine	1
sun microsystems	1

According to the result shown in the tables, there is a considerable amount of improvement in the frequency and the final importance value also increased.

B. *Result comparison before/after optimizing result*

Optimizing module has two main sub modules; they are concept optimizer and relationship optimizer. In the concept optimizer, first we remove the redundancies by converting all the concepts using morphological root.

Sample Results:

- Before optimizing - 'object-oriented programming languages' and 'object-oriented programming language' (plural/singular) are considered as different concepts.
- After optimizing - only 'object-oriented programming language' exists and relationships of both the above concepts are joined

In the optimization phase a strength value is assigned to each concept.

- Before optimizing - only frequency value is considered.
- After optimizing - frequency value and title strength value are considered (according to the existence of concept in titles).

Relationship optimization is performed in reducing concepts; here if the same relationship between the same concepts exist then they are joined by considering the type and strength of those relationships. Also relationships are modified with the strength value using some assertion methods.

By processing text in Appendix C with and without optimization, the results can be examined as follows.

Without optimization

TABLE 4
CONCEPTS AND FREQUENCIES BEFORE OPTIMIZING

ConceptName	Frequency
java	4
programming language	1
programming languages	1

Here both 'programming language' and 'programming languages' exist. Only frequency value exists for filtering concepts.

TABLE 5
RELATED CONCEPTS BEFORE OPTIMIZING

ConceptName	Frequency	Type	RelatedConcept
programming language	1	aso	java
programming languages	1	aso	java

Also both 'programming language' and 'programming languages' has relationships with 'java'.

TABLE 6
CONCEPTS, AND RELATED PARAMETERS AFTER OPTIMIZING

ConceptName	Frequency	TitleStrength	Strength

java	4	4	0.571429
programming language	2	0	0.285714

Here the redundancies are removed. Both 'programming language' and 'programming languages' are joined and frequencies added together. 'Java' concept is in a title so it has highest importance value. 'Programming language' has the next highest importance. The reason for this is because it has a frequency of 2.Other concepts have even lower values as they have lower frequencies.

TABLE 7
RELATED CONCEPTS TO JAVA AFTER OPTIMIZING

ConceptName	frequency	type	Related Concept
programming language	2	aso	java

Here relationships of 'programming language' and 'programming languages' with 'java' are joined and frequencies are added together.

C. *Results of Concept Ranking*

A Document containing the following three sentences is used for the illustration of the results obtained by the concept ranking.

- Java is a programming language.
- Java was developed by Sun.
- Sun developed a programming language.

The concepts found in the document and their statistics and calculated importance are given in the following table.

TABLE 8
CONCEPT RANKING RESULTS

Concept	Concept Frequency	Related concepts	importance	Rank
Java	2	Programmingl anguage ,Sun	0.41532397	1
Program ming language	2	Java, Sun	0.4017329	2
Sun	2	Java, programming language	0.1829432	3

Here all the concepts have the same frequency and each has relationships with the other two concepts but "Java" concept gets a higher importance due to its relationship with "programming language" being an "is-a" relationship and being the head of that relationship. The concept "programming language" has a lower importance than java due to the reason of being the tail of a "is-a" relationship with "java". The concept "sun" has the lowest importance because it only participates in associations with the other two concepts.

D. *Visualization of the Concept Map*

A force directed graph was chosen for visualize the outcomes of the concept map generation process (Fig. 4). Here the nodes represent the concepts and edges represent the relationship among the two concepts. The strength of the relationship between two concepts is represented by the thickness of the edge and the type by the color. Prominence of a particular concept is shown by the size of the node. Related list of documents shorted by the relevance can be obtained by clicking on those nodes.

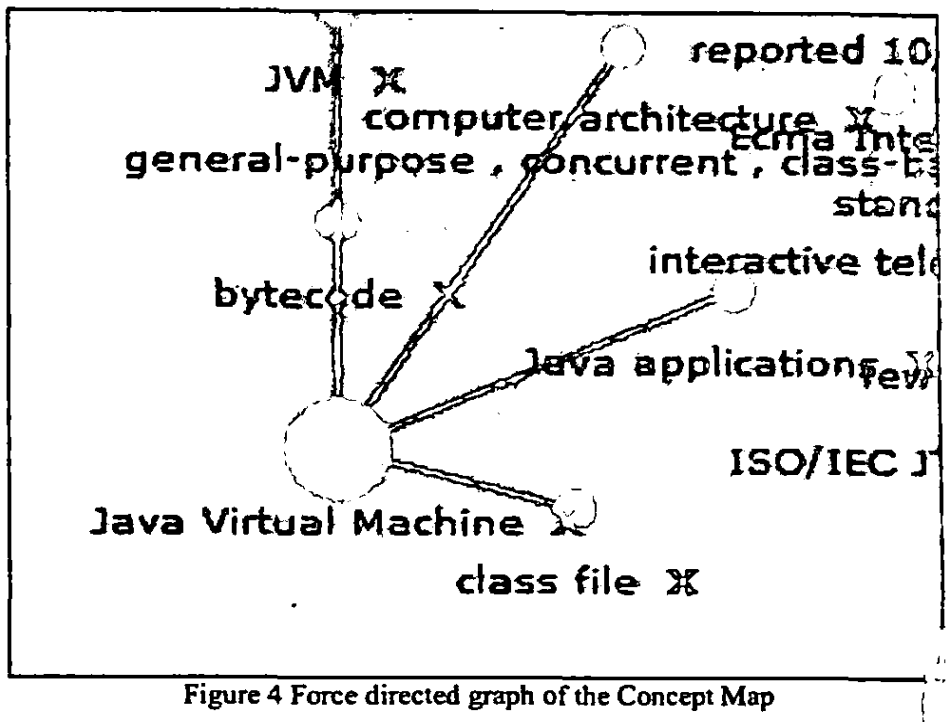

Figure 4 Force directed graph of the Concept Map

V. CONCLUSION

The SigmaC project suggests an elegant and novel solution for enterprise knowledge retrieval by using concept maps which are constructed using advanced natural language processing methods and related technologies. Furthermore the project also provides an automated solution for the creation of ontological data sources saving a vast amount of human labor. Concept map based document summaries provided by SigmaC provide a machine understandable and human friendly representation of knowledge embedded in documents. Patterns which are based on the type of relationships described by the sentence results in a more accurate identification of concepts than those that can be identified using patterns which are not based on the type of relationship described by the sentence

REFERENCES

[1] G. A. Miller, "WordNet: a lexical database for English." Communications of the ACM, vol. 38, no. 11, pp. 39–41, 1995.
[2] D. D. Palmer and M. A. Hearst, "Adaptive sentence boundary disambiguation," 1994, p. 78.
[3] F. Smadja, "Xtract: An overview," Computers and the Humanities, vol. 26, no. 5, pp. 399–413, Dec. 1992.
[4] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, Stroudsburg, PA, USA, 2003, pp. 173–180.
[5] P. Buitelaar and T. Eigner, Topic Extraction from Scientific Literature for Competency Management. .
[6] V. Seretan, "Induction of syntactic collocation patterns from generic syntactic relations," in Proceedings of the 19th international joint conference on Artificial intelligence, San Francisco, CA, USA, 2005, pp. 1698–1699.
[7] C. D. Manning and H. Schuetze, Foundations of Statistical Natural Language Processing, 1st ed. The MIT Press, 1999.
[8] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," Comput. Linguist., vol. 28, no. 3, pp. 245–288, Sep. 2002. .
[9] B. Gelfand, M. Wulfekuler, and W. F. Punch, "Automated concept extraction from plain text," in AAAI 1998 Workshop on Text Categorization, 1998, pp. 13–17.
[10] B. Hachey, "Towards generic relation extraction," 2009.
[11] J. G. Conrad and M. H. Utt, "A system for discovering relationships by feature extraction from text databases," in Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, 1994, pp. 260–270.

[12] T. Hasegawa, S. Sekine, and R. Grishman, "Discovering relations among named entities from large corpora," in Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, 2004, p. 415.

[13] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in Proceedings of the 14th conference on Computational linguistics-Volume 2, 1992, pp. 539–545.

[14] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, Stroudsburg, PA, USA, 2003, pp. 423–430.

Appendix A

Java is a programming language originally developed by James Gosling at Sun Microsystems released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities than either C or C++. Java application is typically compiled to bytecode. It can run on any Java Virtual Machine regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers 'write once, run anywhere', meaning that code that runs on one platform does not need to be recompiled to run on another. Java is as of 2012 one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users.

Appendix B

Java is a programming language originally developed by James Gosling at Sun Microsystems released in 1995 as a core component of Sun Microsystems' Java platform. a programming language which derives much of its syntax from C and C++++ but has a simpler object model and fewer low-level facilities than either C or C++. Java application is typically compiled to bytecode. A Java application can run on any Java Virtual Machine regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. Java is intended to let application developers 'write once, run anywhere', meaning that code that runs on one platform does not need to be recompiled to run on another. Java is as of 2012 one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users.

Appendix C

Java
Java is a programming language. PHP, C++ and C# are some of other famous programming languages. Java application can run on any Java Virtual Machine regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. Java is intended to let application developers 'write once, run anywhere', meaning that code that runs on one platform does not need to be recompiled to run on another. Java is as of 2012 one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users.