ISSN: 2229-6956(ONLINE)
DOI: 10.21917/ijsc.2012.0046

AN INNOVATIVE WEB MINING APPLICATION ON BLOGS - A LAYOUT

S. Prakash[1], T. Chakravarthy[2] and K. Kalyani[3]
[1,2]Department of Computer Science, AVVM Sri Pushpam College, Tamil Nadu, India
E-mail: [1]prakashselvakumar@gmail.com and [2]tcvarthy@gmail.com
[3]Department of Computer Science, T.U.K Arts College, Tamil Nadu, India
E-mail: Kkalyanims@gmail.com

Abstract

Blogs and Web services agree to express user's opinions and interests, in the form of small text messages which gives abbreviated and highly personalized remarks in real-time. Recognizing emotion is really significant for a text-based communication tool such as blogs. Nowadays, user opinions in the structure of comments, reviews in blogs have been utilized by researchers for various purposes. Among them the application of sentiment analysis techniques to these opinions is an interesting one. This paper deals with a proposal of a software structural design for constructing Web mining applications in the blog world. The design includes blog crawling and data mining algorithms, to offer a full-fledged and flexible key for constructing general-purpose Web mining applications. The structural design allocates some significant customizations, such as the construction of adapters for reading text from different blogs, and the utilization of different pre-processing methods and data mining procedures. The core of this paper is on explaining the innovative software structural design of the general framework offering thorough information about the data mining sub-framework.

Keywords:
Blog Mining, Emotional Analysis, Web Crawler, Text Extraction

1. INTRODUCTION

The Internet has been quickly becoming a space for people to express their attitude, opinion, feeling and emotion. On E-commerce blogs, the assessment comments by bloggers of a product can be broadcasted at very high rate in cyberspace, and when a comment is negative that could be very harmful to a venture. Currently researchers have been concentrating much on sentiment classification. The aim is to efficiently spot the emotions of their customers so as to allow companies to respond in a suitable approach to what customers have to say.

With increasing reviews in web, it has been a immense dispute for scientists that how people successfully arrange and process document data to find the most recent information to meet with particular needs and differentiate positive and valueless information. Text sentiment extraction can automatically review the positive or negative, by mining and evaluating the subjective information in the text, such as wish, opinion, attitude, mood, experience and so on.

The software architecture of the combined solution demonstrates how blog crawling and data mining sub-frameworks are combined collectively to provide an important solution for web crawling in blogs. The main aim of this paper is on explaining the software structural design of the general framework providing thorough information about the data mining sub-framework, which uses the semantic web services skill. Instead of the conventional WSDL contract, ontology is used for explaining the service interface. The acceptance of software agents and services also provides a gain to the blog crawling sub-framework, since it allows setting away from on hand tools, such as WordPress. The rest of the paper is organized as follows. Section 2 describes the research review about the sentiment analysis and blogs. Section 3 coated the basic structures which are used in the proposed design. Section 4 launches the proposed structural design, its architecture and its implementation features. Section 5 concludes the work.

2. REVIEW OF LITERATURE

The review of blogs is analyzed first and also study about sentiment analysis also taken for discussion. Nowadays Blogs catch the attention of technological researchers and scientists. One of the researchers correlates a set of keywords with given a specific topic or event and also developed effective algorithms to spot keyword clusters in huge groups of blog posts for particular period [1]. Content–Community–Time model also used which can influence the content of entries, the community structure of the blogs, and timestamps to automatically discover story clusters [2]. Another paper deals with cooperative wisdom technique to cluster blogs thru label sets [3].One direction of blog mining is concentrating on topics of blogs [4]. Another interesting researching gave a chronological analysis on blog information and projected a method to discover trends across blogs [5]. Some had different analysis of calculating the resemblance of two blogs and also added an approach to check the friends who shared the similar topics in their blogs [6].

Sentiment analysis is the core mission of opinion mining, and the majority of the present work concentrated on determining the sentiment orientations of sentences, words and documents [7, 8]. Normally the overall emotions that the bloggers want to express is more complex so the documents are classified into positive and negative based on overall emotions [9, 10]. Sentiment analysis includes subjectivity classification [11], review rating prediction [12] and opinion summarization [13]. Traditional text classification methods with machine learning base have been implemented in sentiment classification and proved the perfectness [14, 15]. It is also exhibited [16] how these models can be topic oriented, domain oriented. In micro blogs different machine learning methods used to classify Twitter messages as positive, negative or neutral [17]. Extracting product attributes from reviews and recognizing opinions related with these attributes has been analyzed [18]. Text categorization [19] and classification [20] were done in opinion analysis. Also different variety of feature sets, like higher-order n-grams [21] [22], part-of-speech based features [22], dependency relations on words [21][22] have been utilized to improve sentiment classification output. Directed bipartite network with users and their comments related to a particular post and a weighted symmetrical posts-and-users network are implemented. In this paper color of the posts and comments indicates their emotional content [23][24]. A newly proposed Time-Weighted Page Rank (TWPR) algorithm is used for ranking on age, event and trend metrics and also compared with standard page rank algorithm. Results shows TWPR method gives proper weighting for all metrics [25]. This paper proposes a framework that contains Text extraction and data mining tasks. First one provides semantic web services and second is capable of extracting necessary information from Web substance. Through these high level tasks the above mentioned activities should be executed sequentially, so that the proposed framework can take a heterogeneous structural design based on the pipes-and-filters design, which are explained in the next section.

3. BASIC STRUCTURES AND DESIGN

Exciting tools are on hand to interact with peoples in social media consist of forums, blogs and wikis. This work is concentrating on the blog world mining. The whole collective blogs are called in the name of Blogosphere. Blogosphere is a social experience whereas blogs are published text of feelings of persons. But web is a form of website whose configuration a allows rapid revise and is preserved by an individual with normal entries of descriptions of events, graphics or video and commentary. Entries are generally presented in reverse-chronological order, concentrating mainly on the projected theme of the blog, it can be written by a many persons, based to the policy of the blog.

Expressing the ideas and sharing knowledge or experience are the main inspiration of the bloggers and majority opinion is that the success of the blog is based on their self satisfaction. But specialized bloggers say that the success is based on number of distinct visitors. Professional or specialized bloggers are more active than those who blog for hobby. This is because the increase of work and family background. Some others do micro blogging and social networks.

3.1 WEB CRAWLER

The search engines have become vital to get appropriate information from huge volume of data that a Web provides. More collection of pages are stored by means of web crawlers. The web crawlers are in charge for moving by web through links, to assemble data. These data are nothing but the information that are indexed to the user, which runs the query efficiently. Web crawler is a source code that takes advantage of the graph formation of the Web to visit the pages. Web may be static or dynamic, if it is static, the crawler is not needed since, all the pages linked in that will be stored in a repository but when it is dynamic, changes occurs in different rates and speed. Thus there is a constant requirement for crawlers to assist applications reside current as new pages manipulated. Fig.1 shows the fundamental step by step crawler

The sequence is explained below.

- Starting process of URL Frontier: The crawler contains the list of unvisited web links. This list is processed in the basis of first in first out.
- Check for process termination : This condition is to check whether any more unvisited links are there or process can go for termination
- Taking URL From Frontier: The crawler thread begins its process by taking URL from frontier.
- Fetching web page: Crawler is fetching a webpage by means of HTTP protocol. Handling last update page and connection problems.
- Parsing: A parser is created to Extract the links and text and provide useful information to the crawler which will guide for next step.
- Preparation of adding to the URL frontier: Previously obtained URL is added to the frontier.

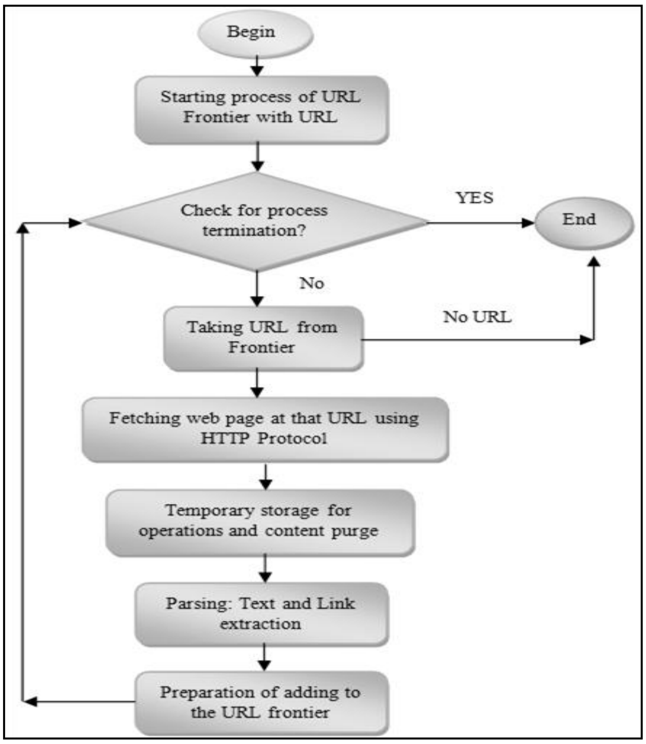
Begin
Starting process of URL Frontier with URL
Check for process termination?
YES
End
No
Taking URL from Frontier
No URL
Fetching web page at that URL using HTTP Protocol
Temporary storage for operations and content purge
Parsing: Text and Link extraction
Preparation of adding to the URL frontier

Fig.1. Sequence Crawler

3.1.1 Crawler Architecture:

In crawler architecture (Fig.2) there are two essential modules which are called the crawling application and crawling system. The duty of the crawling system is to access the Web and download pages. Further the application stores the extracted information and sends for parsing and scheduling.

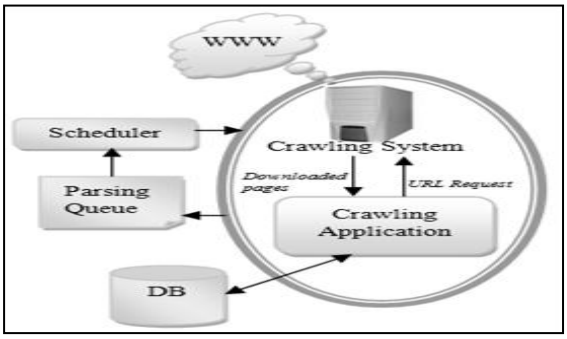

Fig.2. Crawler Architecture

3.2 THE LAYERED STRUCTURAL DESIGN

The basic idea of software development is to separate the whole system into the form of integrated jointed subsystems. Each interface in the modules defines the service they can provide. A subsystem may be a client or server. Though a well defined interface all subsystems have communications with others. But the classes in a subsystem should collaborate only with classes in the same subsystem. During the partition the parallel activity occurs, that is division in layers. Each layer of a system may include more than one subsystems. The subsystem in layer n get services from n+1(below) layer and provide services to n-1(above) layer.

This structure has some responsibilities for which three layers are used. View layer is used by the user to interact with system, control layer is to implement rules related to the user application, model layer take care of data storage. This is called model-view-control architecture or three-tier architecture.

3.3 THE REPOSITORY STRUCTURAL DESIGN

This design has two different components: (a) the storage centralization of data is done by the repository; and (b) a set of autonomous components, which revise and utilize the repository data. The contact between these systems takes place through data sharing. For example integrated development environment and CAD tools usually apply this design. CAD and IDEs, generally assimilate modeling tools, editing support, visuals and many other features based on single data set. With the use of control flow in this design two variations can be implemented: (a) processes are executed by the trigger of input system (b) processes are executed by the internal state trigger of the repository.

3.4 THE PIPES AND FILTERS STRUCTURAL DESIGN

The endpoints (filters) are connected with each other through channels (pipes) to send messages. Each filter does not know about other filter and the information they are processing. So after the process the filter send the output to the next filter (which will take the massage as input) by means of pipes. The system may can do manipulations on filters at runtime, that is add, delete or rearrange the filters to get desired output. So larger tasks are separated and allocated to different filters and in sequence the modules are processed, for example signal processing, parallel programming, compilers and also UNIX Shell.

3.5 HETEROGENEOUS STRUCTURAL DESIGN

Though it is important to understand the above said basic structures individually, the implementation of factual systems constantly entails a mixture of many of them. The mixture may be in many ways, among them hierarchy combination is the common one. A component which is inserted into a system with specific structure need not be in the same structure in which it is to be inserted. The connectors can also be different one, so that components communicate with each other thru different protocols.

3.6 SEMANTIC WEB SERVICES

Some complex tasks in the web will be performed without direct intervention of humans. To do these intelligent agents should have the capability of investigating the semantic description of the above said services. For the implementation of this situation semantic descriptions are created through ontology to the traditional web services by semantic web services. So the semantic web services have the capability of both traditional technologies and the ability to automate present features of semantic web services. This is depicted in Fig.3.

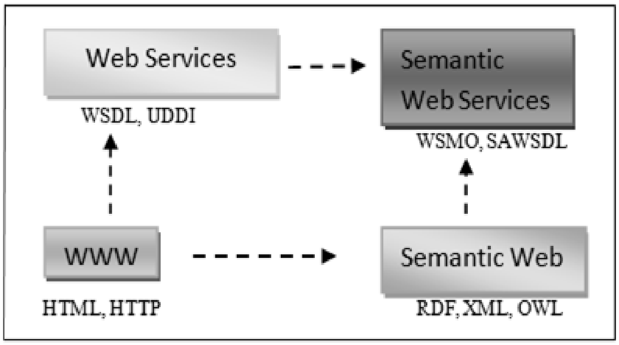

Fig.3. Semantic Web Services

While the automation process is done without human interaction, supervising and assessing is also provided to improve results. This technology can also afford mechanism to guarantee its autonomy while executing functional and non-functional requirements.

4. STRUCTURAL CONSTRUCTION OF THE PROPOSED PLAN

The main aim of the proposed framework is to have two high level tasks: (a) Text extraction and (b) data mining. First one contains semantic web services and second is capable of extracting necessary information from Web substance. The above mentioned activities should be executed sequentially, so that the proposed framework can take a heterogeneous structural design based on the pipes-and-filters design as explained above.

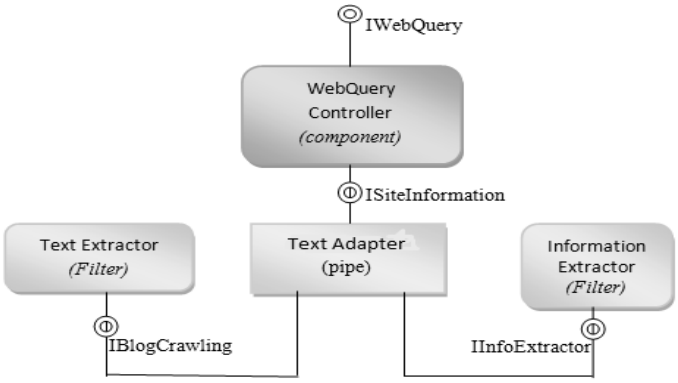

Fig.4.Top-level view of the software structured design

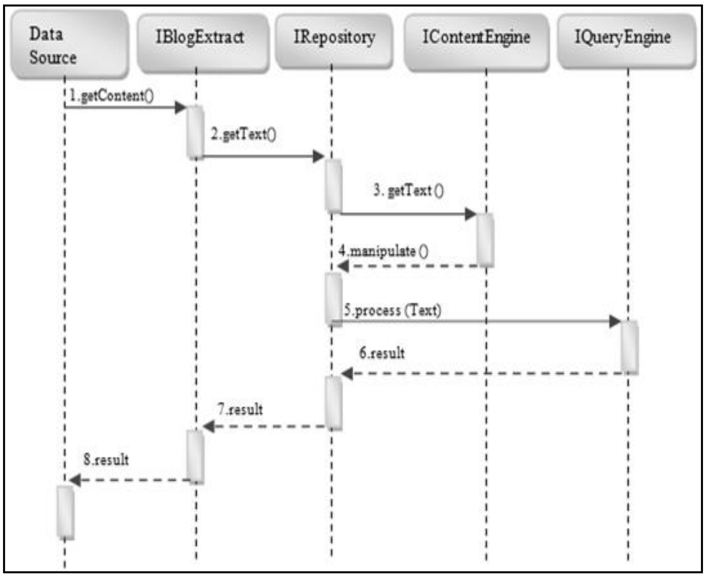

Fig.5. Sequence diagram of the top-level software design

Fig.4. depicts paired components (filters) and a connector, a pipe to connect them. Text extractor component is in charge of extracting text from website information, according to this paper from blogs. Information extractor is in charge of extracting the extra information from the content received from text extractor, which is passed by the text adaptor connector. Web mining controller will initiate the process of extracting the blog content, which is finished by the ISiteInformation interface of the text adaptor connector. IWebQuery is used by framework application for extraction.

The sequence diagram in Fig.5 explains the Extractor and Information Extractor components. From this diagram, the two level of preprocessing, that is HTML-level preprocessing, and content level preprocessing are clear. HTML processing is prepared by the Text Extractor component after evaluating the substance of an HTML page. Content-level preprocessing is completed by the Information Extractor component, through which the execution of the algorithm will be faster.

As demonstrated in Fig.5, to apply the proposed structure it is essential to utilize the services of IBlogExtractor interface of WebQueryController. This component initiates the information extraction by getting the respective service from the IRepository interface of the TextAdapter connector. This connector implements the pipes and filter protocol and so it first request the extraction of text from blogs, by means of IContentEngine interface, then information extraction, providing the extracted text as input for the IQueryEngine interface.

4.1 TEXT EXTRACTOR IN-HOUSE STRUCTURAL DESIGN

The Text Extractor component describes modules for preprocessing, indexing, extracting HTML content, and other jobs necessary for fine-tuning the quality of the text extraction from HTML pages.

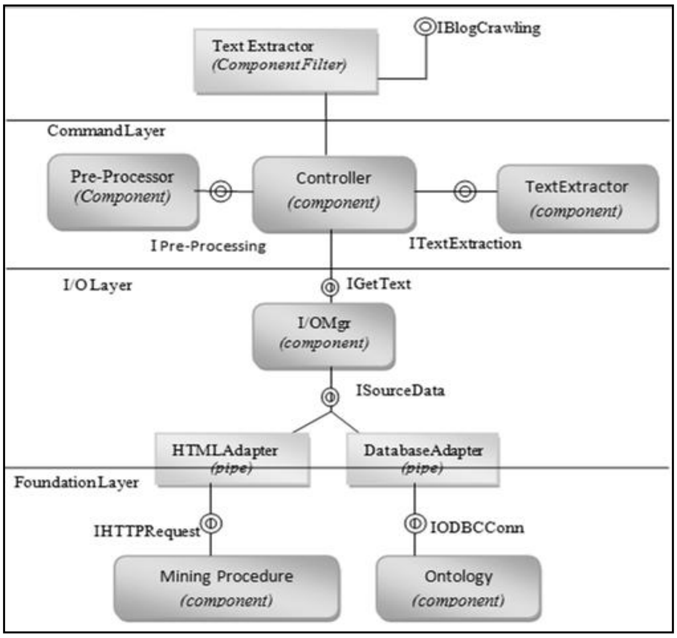

Fig.6. Internal structure of text extractor component

As shown in Fig.6, there are three layers

(a) Command layer: Includes components in-charge for implementing the initial actions for extracting text from web blogs.

(b) I/O layer: Includes components in-charge for reading text content.

(c) Foundation layer: Includes components in-charge for the target repositories to be mined. This may be either Web-based repositories that use HTTP protocol or database supported repositories which use ODBC protocol.

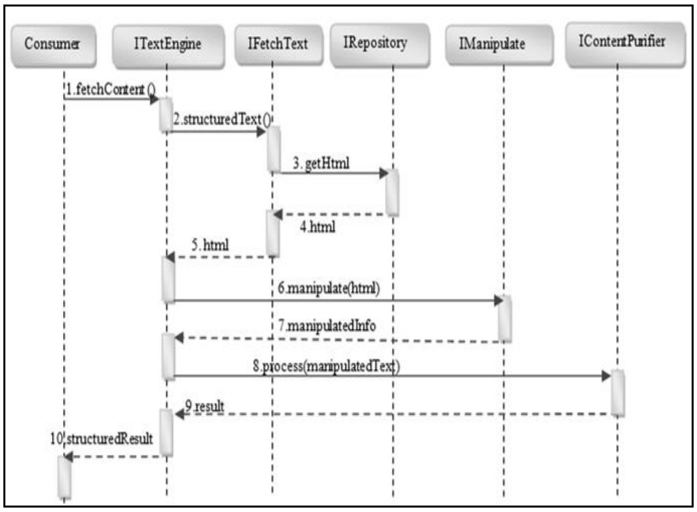

Fig.7. Sequence diagram of text extractor module

The presence of repositories assists the assimilation of the proposed structure with other existing applications which produce text content. This attribute can be seen as the performance of the repository structural design.

As depicted in Fig.7. The ITextEngine interface offers services of text extraction from Web substance. Then, the

Controller component gets the text content with the help of IFetchText interface of the I/OMgr component. By means of IRepository, I/OMgr can obtain either HTML pages, using the HTMLAdapter or get database content by using the DatabaseAdapter.

The preprocessing activity starts now, which includes a set of configurable actions, such as removing white spaces, cleaning HTML, stemming the substance, sorting information, and other preprocessing procedures employed by the user. The controller requests this preprocessing through the IManipulate interface. Thus, after the preprocessing, the text content can be extracted with the help of IContentPurifier interface of the content purifier component. This extraction includes process of the textual content in order to develop the semantic quality of the text by means of specific techniques. The user can also describe new methods when instantiating the structure.

4.2 INFORMATION EXTRACTOR IN-HOUSE STRUCTURAL DESIGN

The Content Extractor module includes a design for extracting information from texts with the help of query engine. This design pursues a service based method in order to provide the addition and integration with diverse applications over the Web. To deal with queries, the design describes components for preprocessing the text, and selects diverse mining procedures, based on the intention of the application. This proposed framework was employed for developing an e-commerce application.

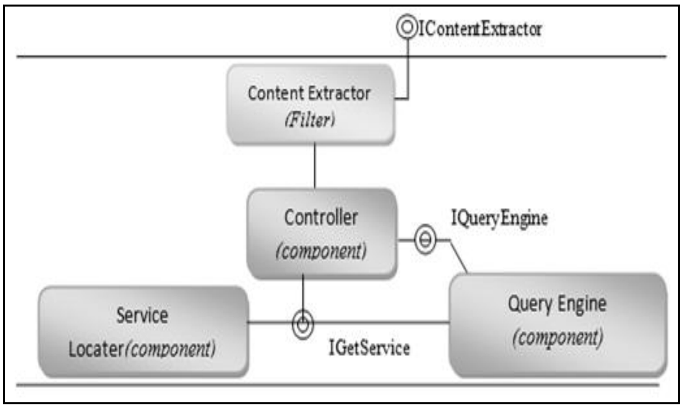

Fig.8. In-house structure of information extractor structural design

Fig.8 depicts the internal structural design of the Content Extractor component. Here a heterogeneous structural design is followed, which merge the layered architectural style with the Service Oriented Architecture. A special component, Service Locator is responsible for explaining and locating all the services described by the application. The presence of this component is the main difference between the software architecture actually defined and the layered architectural style. Another component is Query Engine component, which accepts a text as input and is in charge of mining it, to extract extra information. The extra information may be like emotional information and feedback about the sentiment connected to a particular subject.

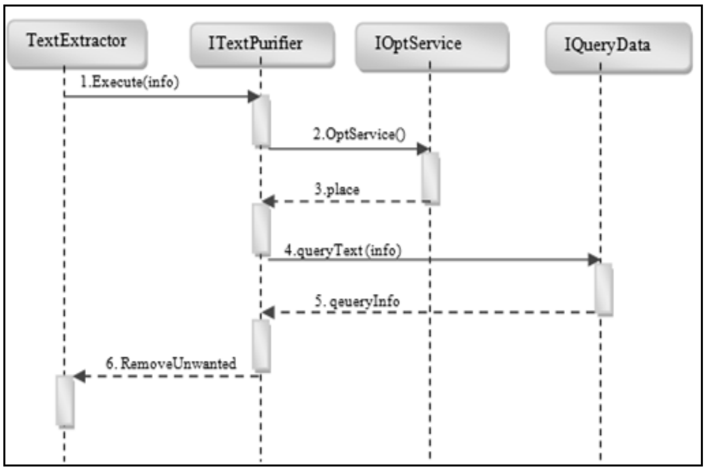

Fig.9. Sequence diagram of content extractor

As depicted in Fig.9 the information extraction initiates when a request and the text should be mined are obtained by the ITextPurifier interface of the Controller component. The internal structure of the Data Query Process component follows a heterogeneous architectural style which combines layered structural design (Section 3.2) and the pipes-and-filters structural design (Section 3.4).

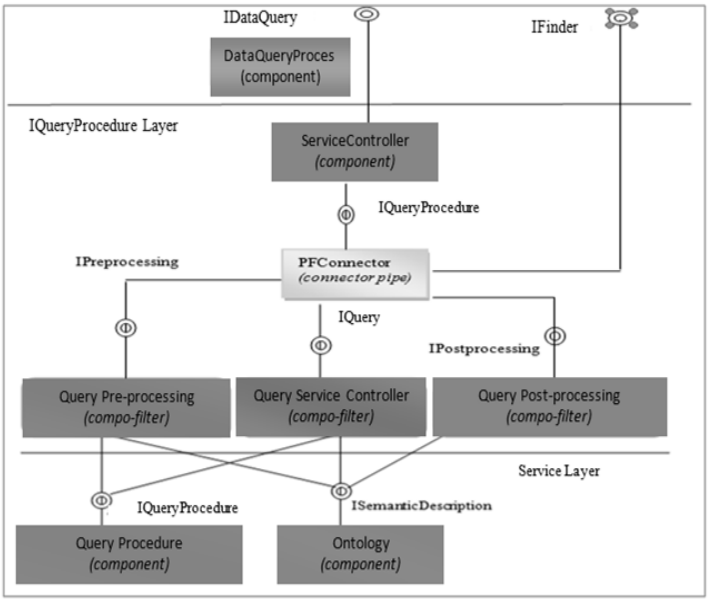

Fig.10. Internal structure of data mining layer

Fig.10 depicts data mining process, which is compiled by three basic functions: preprocessing, data mining and post processing. These functions are incorporated as Query Service Controller.

After the receipt it locates the data mining service which is to be executed. The location process is fulfilled through IFind interface of service Identification module. After this, a request is broadcasted to the particular service, which employs interface IDataQueryProcess of the DataQuery component. This module is responsible for extracting information from text content.

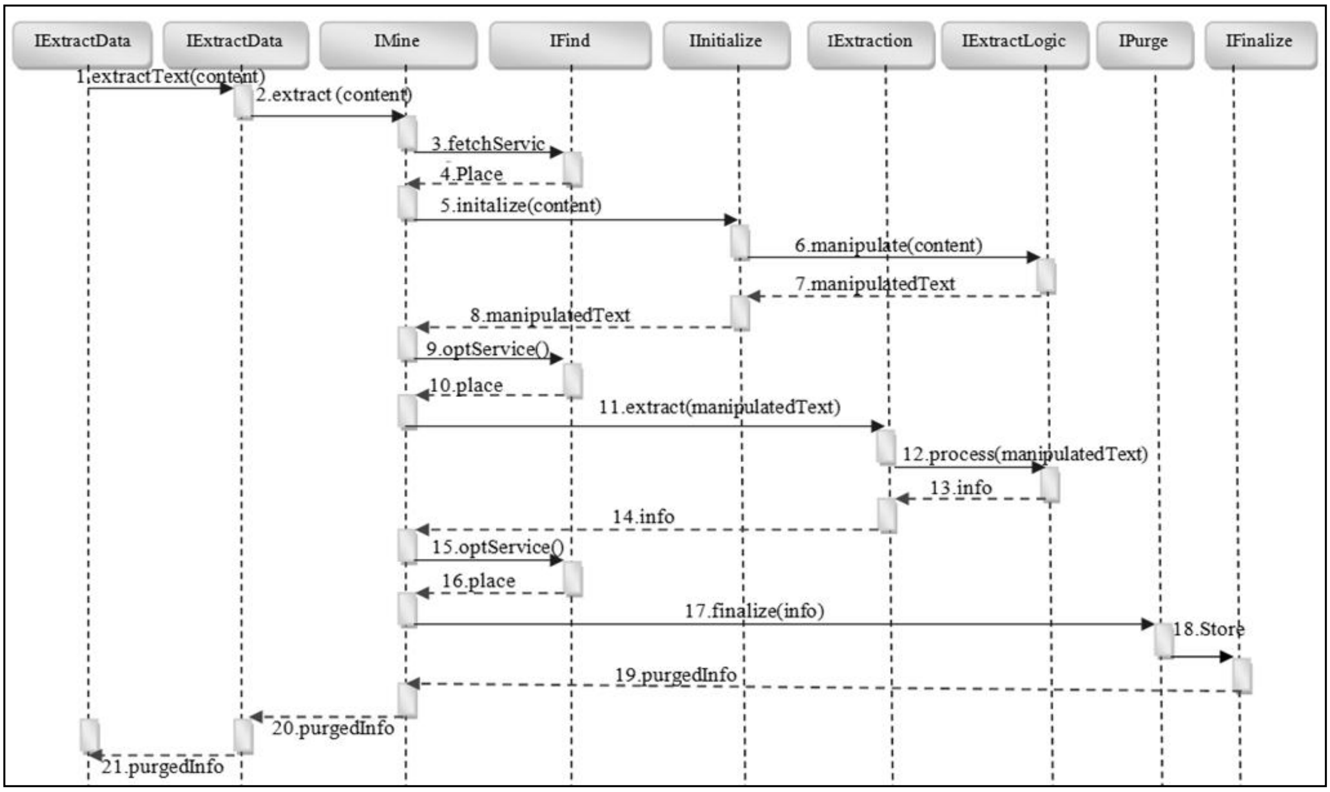

Fig.11. Sequence diagram of data mining module for information extractor

The Query Preprocessing component includes services that organize the text data before it can be mined. The Query Service Controller component includes services that summarize mining procedures and tools as Web services. The Query postprocessing component includes services that employ postprocessing procedures relevant to the specific application in which the structure is being applied. The arrangement of preprocessing, data mining and postprocessing is completed by the Query Service Controller component. This web service is in-charge of determining, arranging and inducing the available semantic web services at the Data Query component. Above and beyond, it supports the execution of the data mining process. For this two valuable components are also described: (a) Query Procedure component (b) Ontology component. Fig.11 provides the activities of the Data Query component, after receiving a request in the IDataQuery interface (Fig.10).

Query Service Controller component receives service request and text to analyze, and then it ask for the execution of the data mining process by the IQueryProcedure interface of the PFConnector. This connector executes the three following components in sequence (pipes-filter design), by arranging their services. First, a service of Query Preprocessing is established by IFindService interface and executed. Preprocessing services can employ procedures provided by the Mining Algorithms component by IDataQueryProcedure interface.

After preprocessing the text, a mining service is situated and executed passing the preprocessed text as an argument to the IDataQueryProcedure interface. Lastly, the mining outcomes should be postprocessed based on specific purpose. All the data mining services are semantically indicated using ontology i.e., IModelDescription (semantic) interface.

5. CONCLUSION

This paper has offered a software design for constructing blog mining applications in e-commerce set-up, also provides a set of services to lighten the effort of application developers. The proposed design has offerings which makes it from other existing design. The use of semantic web services is considered very significant for attaining this benefit, since it progresses the system autonomy by permitting automatic discovering, composition and completing of services. In the background of software engineering, the most vital advantage of using semantic web services is the substantial cutback of the coupling between services, which has a positive contact into the system understandability and maintainability. The low-coupling is attained by using individual and self-governing service agreement definition by means of ontology, instead of a single WSDL file. Moreover, web services are also used for summarizing existing tools and maximize reprocess. A limitation of this work is the need to manual interpretation of blogs in order to improve the accuracy of the crawling task. Also for blogs which have multiple content we need NLP.

Regardless of these boundaries, the proposed approach is very useful in perspective of blogs of a single subject, such as those used for promoting specific products on the web and forums. The extension of this work may be with NLP to develop a tool for automatically annotating blogs.

REFERENCES

[1] N. Bansal, F. Chiang, N. Koudas, F. Tompa, “Seeking stable clusters in the blogosphere”, *Proceedings of the 33rd*

International Conference on Very Large Data Bases, University of Vienna, Austria, pp. 806–817, 2007.

[2] Qamra, B. Tseng, E. Chang, "Mining blog stories using community based and temporal clustering", *Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM, USA*, pp. 58–67, 2004.

[3] N. Agarwal, M. Oliveras, H. Liu, S. Subramanya, "Clustering blogs with collective wisdom", *Proceedings of the Eighth International Conference on Web Engineering, Yorktown Heights, New York, USA*, pp. 336–339, 2008.

[4] J. Bar-Ilan, "An outsider's view on topic-oriented blogging", *Proceedings of the Alternate Papers Track of the 13th International World Wide Web Conference, New York, USA*, pp. 28-34, 2004.

[5] N. Glance, M. Hurst, T. Tornkiyo, "Blogpulse automated trend discovery for weblogs", *Proceedings of WWW Workshop on the Weblogging Ecosystem, NewYork, USA*, 2004.

[6] D. Shen, J. Sun, Q. Yang, Z. Chen, "Latent friend mining from blog data", *Proceedings of ICDM, 6th International Conference on Data Mining, Hong Kong, China*, pp. 552–561, 2006.

[7] M. Efron, "Using cocitation information to estimate political orientation in web documents", *Knowledge Information System*, Vol. 9, No.4, pp. 492–511, 2006.

[8] T. Fan, C. Chang, "Sentiment-oriented contextual advertising", *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pp. 202-215, 2009.

[9] B. Pang, L. Lee, S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques", *Proceedings of the ACL-02 Conference on Empirical methods in natural language processing, Philadelphia, PA, USA*, Vol. 10, pp. 79–86, 2002.

[10] P. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews", *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistic(ACL), Philadelphia, PA, USA*, pp. 417–424, 2002.

[11] J. Wiebe, E. Riloff, "Creating subjective and objective sentence classifiers from annotated texts", *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing'05). Mexico City, Mexico*, pp. 486–497, 2005.

[12] Z. Zhang, B. Varadarajan, "Utility scoring of product reviews", *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM'06), ACM, New York, NY, USA*, pp. 51–57, 2006.

[13] M. Hu, B. Liu, "Mining and summarizing customer reviews", *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'04), ACM, New York, USA*, pp. 168–177, 2004.

[14] T. Mullen, N. Collier, "Sentiment analysis using support vector machines with diverse information sources", *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pp. 412–418, 2004.

[15] B. Pang, L. Lee, S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques", *Proceedings of the ACL-02 conference on Empirical methods in natural language processing (EMNLP'02), Association for Computational Linguistics, Morristown, NJ, USA*, pp. 79–86, 2002.

[16] J. Read, "Using emoticons to reduce dependency in machine learning techniques for sentiment classification", *Proceedings of the ACL Student Research Workshop, ACL'05. Association for Computational Linguistics, Morristown, NJ, USA*, pp. 43–48, http://portal.acm.org/citation.cfm?id= 1628960.1628969>, 2005.

[17] V. Pandey, C. Iyer, "Sentiment analysis of microblogs". <www.stanford.edu/class/ cs229/proj2009/PandeyIyer.pdf>, (accessed 11.10), 2009.

[18] A.-M. Popescu, O. Etzioni, "Extracting product features and opinions from reviews", *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT'05), Association for Computational Linguistics, Morristown, NJ, USA*, pp. 339–346, 2005.

[19] Y.-S. Dong, K.-S. Han, "A comparison of several ensemble methods for text categorization", *The 2004 IEEE International Conference on Services Computing (SCC)*, pp. 419–422, 2004.

[20] S. Li, C. Zong, X. Wang, "Sentiment classification through combining classifiers with multiple feature sets", *Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE 07)*, pp. 135–140, 2007.

[21] K. Dave, S. Lawrence, D.M. Pennock, "Mining the peanut gallery: opinion extraction and semantic classification of product reviews", *Proceedings of the International World Wide Web Conference (WWW)*, pp. 519–528, 2003.

[22] V. Hatzivassiloglou, J. Wiebe, "Effects of adjective orientation and gradability on sentence subjectivity", *Proceedings of the International Conference on Computational Linguistics (COLING)*, pp. 299–305, 2000.

[23] V. Subrahmanian, D. Reforgiato, "AVA: adjective–verb–adverb combinations for sentiment analysis", *IEEE Intelligent Systems*, Vol. 23, No. 4, pp. 43–50, 2008.

[24] M. Mitrovi´c, G. Paltoglou, and B. Tadic, "Networks and emotion-driven user communities at popular blogs", *The European Physical Journal B*, Vol. 77, No. 4, pp. 597–609, 2010.

[25] Bundit Manaskasemsak, Arnon Rungsawang., and Hayato Yamana, "Time-weighted web authoritative ranking", *Information Retrieval - Springer*, Vol. 14, No.2, pp. 133-157, 2011.