

WikiEdit: A Service for Suggesting Wikipedia Pages to Edit

Gargee Anjkar

Department of Computer Science
and Electrical Engineering,
University of Maryland,
Baltimore County
Baltimore, Maryland 21250
gargee1@umbc.edu

Rakesh Deivachilai

Department of Computer Science
and Electrical Engineering,
University of Maryland,
Baltimore County
Baltimore, Maryland 21250
rakeshd1@umbc.edu

Vivek Vijayan

Department of Computer Science
and Electrical Engineering,
University of Maryland,
Baltimore County
Baltimore, Maryland 21250
vivivek1@umbc.edu

Abstract – WikiEdit is a service that recommends Wikipedia users pages to edit on Wikipedia. Wikipedia, being one of the largest repository of information, is usually the first stop for people seeking answers and knowledge on a specific subject. Hence, it is of significant importance to keep such a widely popular database of information constantly updated. WikiEdit encourages Wikipedia users to make further contributions to Wikipedia, by suggesting pages to edit based on their edit history as well as area of interest. As we also include our user's area of interest, the probability of the user in actually editing articles suggested by WikiEdit is greatly increased unlike other existing similar services. There is this possibility that pages that a person might have never thought existed in his area of interest can be suggested by WikiEdit. The various software tools and computing services used in developing our service involves the concepts taught during this course including SOAP, WSDL, MapReduce and Hadoop.

Key words–Wikipedia, Recommendations, Web Service, WikiEdit.

I.INTRODUCTION

If you are an enthusiastic Wikipedia contributor, you would often find yourself in a predicament to get appropriate pages to edit or contribute to. Being an expert in your field, there are a number of Wikipedia articles that would

benefit from your contributions. Instead of looking for different pages to edit by browsing through pages on Wikipedia, WikiEdit - our web service, will provide links to pages that a user can contribute to based on the users previous edit history and area of interest. The suggestions provided by existing services are based solely on the user's previous contributions on Wikipedia.

The scenario of being presented with suggestions based only on edit history has a few critical drawbacks. The most notable drawback can be explained with the following scenario. A user, who is an expert in the field of Computer Science has come across an error in an article related to Computational Biology. Considering that the user edited this article, he will be given suggestions related to Biology and Computational Biology along with the other Computer Science articles. This can get quite annoying in the long run if the user isn't really interested in contributing to Biology or related articles. Thus, this defeats our main purpose of making users contribute Wikipedia. WikiEdit resolves this problem by taking the area of interest of the user in account while filtering the links. By doing so, a user is ensured that even though he has edited articles that are out of his realm of expertise, he will not be entirely presented with suggestions related to these articles. WikiEdit first goes through the edit history of a user and comes up with suggestions related to these edits. Then, these suggestions are constrained with the areas of interest of the user.

The filtered suggestions are then displayed to the user.

II.MOTIVATION

The main motivation in building this service was to help constantly improve and update the vast bank of knowledge, that is, Wikipedia. The main contributors to Wikipedia are usually professionals and experts in their respective fields. By providing a service which readily makes available a list of links to articles in the field of expertise of our user, we encourage more people to make contributions to Wikipedia. This in the long run can be extremely useful as we like many others, believe the future of computers and technology lies in open source.

III. RELATED WORK

SuggestBot – Wikipedia provides a program called SuggestBot[1], which helps users by providing suggestions based on their past contributions to Wikipedia. As explained before the problem with being presented with suggestions only based on edit history can be quite tiresome if the user isn't interested in editing those articles. Another interesting feature provided by SuggestBot is that you can sign up to receive these suggestions regularly. The suggestions are made by using a list that contains all the articles edited by the user.

IV.USERS

This service can be used by enthusiastic Wikipedia users who have made contributions to Wikipedia in the past and who wish to make further contributions. We require our users to have made at least one contribution. This is similar to a scenario in which a person in the USA wishes to apply for a credit card or a loan/mortgage. The first thing that the bank or mortgage company does is to look at the credit history of the applicant. Just like how a credit history is required for the applicant, the users of our service will need to have an edit history on Wikipedia. If a user doesn't have any edit history

at the time of registering for our service, the user will not be able to register and would be asked to make contributions to Wikipedia before using our service.

V. SCENARIO

The main objective of this service is to encourage more users to contribute to Wikipedia. The scenario in which our service will be used is when a user wants to make contributions to Wikipedia but has a tough time finding which articles to edit or add to. By providing recommendations on articles to edit, by taking into consideration the edit history as well as the area of interest, this services makes the task of finding articles to edit relatively easy.

VI. BIG DATA

The Wikipedia data dumps [2] provide a complete collection of all the pages on Wikipedia that were edited along with other information including username of the person who edited the page, the new content that was added to the page, timestamp etc. These dumps included the changes made to user pages and talk pages as well. The compressed version of this data was about 4 TB. Since our service mainly relied on the revisions made to the articles, we really did not need anything that dealt with the user and talk pages.

After a quite a lot of research, we found a source [3] that had processed the above mentioned dump and provided a data set that included only the changes made to articles along with relevant information. This data set was 400 GB in size. This data set provided the list of articles on Wikipedia along with the users who have edited the article, the content they had added or changed and other information like the date, time, category, etc.

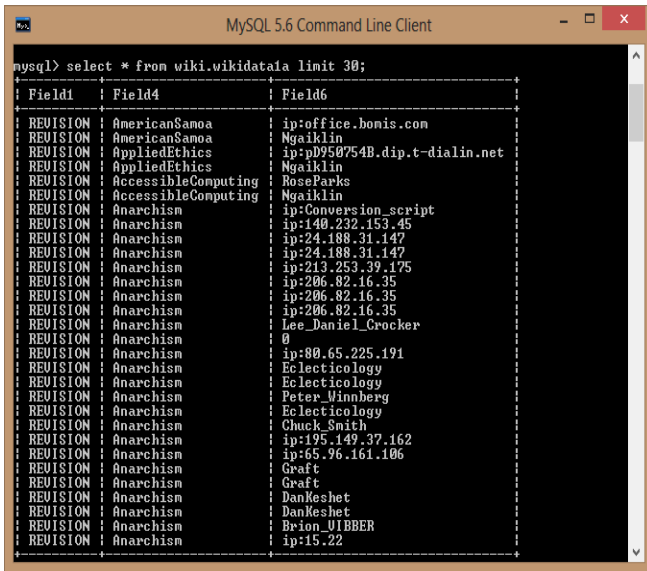
The other data sets that we have used for our service were obtained from DBpedia [4], which is a website that makes available structured Wikipedia data on the web. The first dataset is called Article Categories [5]. This dataset was in text format and included the DBpedia links to Wikipedia articles and the

category that the particular article falls under. This dataset was 2.35 GB in size.

The second dataset is called Links to Wikipedia Article [6]. This dataset consisted of the Wikipedia links to the corresponding DBpedia resource. This dataset was in text format too and was 3.8 GB in size.

VII. PREPROCESSING THE BIG DATA

Figure 1: The Processed Dataset loaded into My.



FieldId	Field4	Field6
REVISION	AmericanSanoa	ip:office.bomis.com
REVISION	AmericanSanoa	Mgaiklin
REVISION	AppliedEthics	ip:p950754B.dip.t-dialin.net
REVISION	AppliedEthics	Mgaiklin
REVISION	AccessibleComputing	RoseParks
REVISION	AccessibleComputing	Mgaiklin
REVISION	Anarchism	ip:Conversion_script
REVISION	Anarchism	ip:140.232.153.45
REVISION	Anarchism	ip:24.188.31.147
REVISION	Anarchism	ip:24.188.31.147
REVISION	Anarchism	ip:213.253.39.175
REVISION	Anarchism	ip:206.82.16.35
REVISION	Anarchism	ip:206.82.16.35
REVISION	Anarchism	ip:206.82.16.35
REVISION	Anarchism	Lee_Daniel_Crocker
REVISION	Anarchism	0
REVISION	Anarchism	ip:80.65.225.191
REVISION	Anarchism	Eclectology
REVISION	Anarchism	Eclectology
REVISION	Anarchism	Peter_Winnberg
REVISION	Anarchism	Eclectology
REVISION	Anarchism	Chuck_Smith
REVISION	Anarchism	ip:195.149.37.162
REVISION	Anarchism	ip:65.96.161.106
REVISION	Anarchism	Graft
REVISION	Anarchism	Graft
REVISION	Anarchism	Dankeshet
REVISION	Anarchism	Dankeshet
REVISION	Anarchism	Bvion_UIBER
REVISION	Anarchism	ip:15.22

The data that we obtained from the dumps have a lot of unnecessary information that we do not require for the implementation of our service.

From the 'edit history' dataset we required only the first line from each section. This line started with the word REVISION and had the article name, the Wikipedia username of the person that edited the article as well as other information like the time stamp, the page id, etc. To extract this particular line from the dataset we wrote a MapReduce program. The Mapper checked if the line started with the word Revision and the Reducer added them to the output file. A java program was written to convert the reduced text file into a table in a MYSQL database. Several unnecessary columns were dropped and the columns containing the username and the article name that the user edited were retained.

Since the 'Article Categories' dataset had only one line of data for each article, we used the java program to directly convert it into a MySQL database. After uploading it into MySQL, we dropped the unnecessary columns in the table and only the columns with the categories and the column with the corresponding DBpedia resource link were retained. Similar processing was also done to the 'Links to Wikipedia Article' dataset, and the MySQL table consists of only two columns. One column containing the links to the DBpedia resource and the other column containing the corresponding link to the Wikipedia article.

VIII. FUNCTIONAL REQUIREMENTS

New User Registration

Input – When a new user registers with our service, the following information is required as input: Wikipedia user name, a password, email id and the user's area of interest.

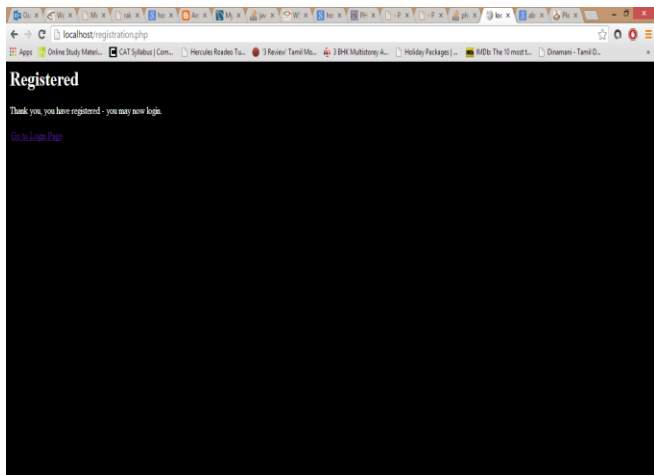
Service – The service required once the user has entered the relevant information in the registration form is to first validate the username provided in the form matches the Wikipedia username in our edit history database. After the match is successful, the user is registered with our service and if not, the user is not allowed to register until he provides a valid username.. The password entered by the user is encrypted using MD5 and stored in the user registration database.

Figure 2: New User Registration for our Service.



Results – If the user name already exists in the edit history database and not in the user registration database, then the user is displayed a message saying that the registration was successful. If the user has not made any past contributions to Wikipedia (resulting in his user name not being present in the edit history database), a message is displayed requesting the user to make at least one contribution to Wikipedia before availing our service.

Figure 3: After the user has successfully registered.



User Login

Input – The users of our service logs in by providing the Wikipedia username that was used during registration and the corresponding password.

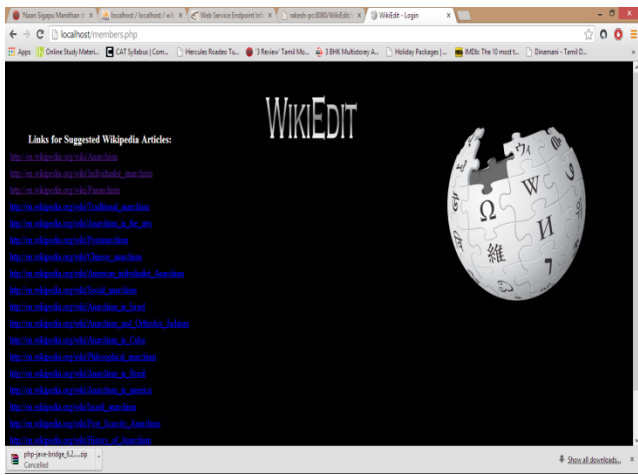
Figure 4: The Login Screen for our service



Service – When the user clicks on the login button, the first task that needs to be carried out is to validate the username and password using the registration database. Once the user is authenticated, the user is directed to the home page of the service which displays the list of suggestions.

Results – The suggestions are displayed in the form of links, which when clicked opens the Wikipedia article.

Figure 5: Successful Login.



IX. COMPUTATION

The main computation that was required of our service was to ensure that the suggestions given to the user included articles that were in the area of interest of the user.

When the user logs in to our service, the first thing that the service does is to authenticate the user. This is explained in section VIII under *User Login*. Once the user is authenticated, our service looks up the edit history of the user. This can be done with the help of our preprocessed data in the 'edit history' table of our database.

Once we have the edit history of the user we can look up the categories of these articles that the user has previously edited. This can be done with the 'Article Categories' table in our database.

Now that we have the categories of the articles that are previously edited by the user, we can eliminate the articles that do not fall under the area of interest of the user. In this way we get a collection of categories that the user is actually interested in.

The Article Categories table is again used to get the DBpedia resource links that fall under the filtered category list. Then, querying the Links to obtain the Wikipedia Articles using this link, we get the actual Wikipedia link to each of the articles. This links are then displayed to the user, so that when the user clicks on the link, he will be directed to the Wikipedia article.

X. SOFTWARE AND COMPUTING SERVICES USED

We have used a wide variety of Software and Computing services in implementing our project. The complete list along with their functionalities is given below-

1. Web Service Description Language (WSDL) - It is compiled in XML and is used to describe the web services. WSDL provides the location of the web service and the operations (or methods) that the service exposes. It acts as the contract for other applications to consume our web service.
2. Simple Object Access Protocol (SOAP) - SOAP is an XML based protocol for accessing web services. It is based on XML which is used for exchanging structured information between the various modules.
3. Oracle Glassfish - We have used the Glassfish platform for delivering the server side java applications and web-services. The WSDL is produced by glassfish which in turn is consumed by our Application. We have used Oracle Glassfish V4.0 in our project.
4. Java SD - We have used Java for implementing our Web Service. It mainly consists of 3 modules - a) Looking up the Wikipedia Username. b) Retrieving the previous edit history of the user. c) Filtering the Wikipedia Suggestions based on the User's Area of Interest and then displaying the final result. We have used Java SD Version 7.0.
5. Bluegrit - We have used the grid computer for doing parallel data processing by running map reduce programs. The Hadoop Map - Reduce program is used to retrieve only the part with the edit history from the Wikipedia dumps.
6. Wikipedia Dumps - The Wikipedia dumps are used to retrieve the edit history of the Wikipedia users. The DBpedia datasets are used to find the category of the Wikipedia Article and also to provide the suggestions for the users of our applications in the form of Wikipedia links.
7. MySQL - The Relational database scheme used in our Project is MySQL. We have used 3 databases - Wikidata: This database contains the edit history of all the users, Category: This database contains all the categories of the Wikipedia articles, Wikiuser: This contains the list of all the registered users of our service.
8. Cloudera QuickStart Virtual Machine - The Cloudera QuickStart Virtual Machine provides us with the Linux Platform to execute our application and it also has Hadoop Map Reduce installed and enables us to do data processing. We have used version 4.4.0-1.
9. VMware Virtual Player - VMware Virtual Player is used to start the Cloudera Quickstart Virtual Machine. The version that we have used is VMware Player v11.

XI. SECURITY

Our Service is equipped with sound security features that will prevent bots and other malevolent users from registering or attacking our service. The first security measure that we take is to ensure that the user exists in our edit history table in the database. This confirms two necessary details which are: the username is a valid Wikipedia username and the user has made previous contributions to Wikipedia.

Once it is verified that the username exists in the edit history table of our database, we can proceed to register the user with our service by adding the user in our user registration table of our database. While doing this, we store the username of the user and the corresponding password entered by the user. The encrypting of the password is done using MD5 [7]. MD5 is a hashing encryption technique that produces a 128 bit hash key. The usual representation of this key is a 32 digit hexadecimal value and this value is stored in the user registration database. The MD5 encryption scheme prevents any unauthorized access.

XII. CONCLUSION

WikiEdit is a secure and user friendly service that encourages enthusiastic Wikipedia users and contributors to share more information and knowledge on Wikipedia, possibly the world's largest bank of knowledge.

The main characteristic of our project is that we have considered the area of interest of our users as well in filtering the Wikipedia suggestions list. This will greatly increase the possibility of the user in actually contributing to Wikipedia.

XIII. FUTURE WORK

Possible future work can include keeping track of the frequency with which a user edits a particular category of pages. This could mean that the user is more likely to contribute to

similar pages and we can increase the number of suggestions relating to this category in our recommendation list.

We can also include forums for user with similar area of interests to interact with each other and this will lead to quality information being uploaded to Wikipedia.

XVI. LESSONS LEARNED

This course was very effective in helping us to understand the details of developing an effective web service. At first, we learnt how to deal with large amounts of data. The means of analyzing and understanding this data with the help of different tools available was useful in understanding the way data is represented and used in today's era. Importing the data into MySql is something fascinating that we have learnt during the course of this project and we are positive that it will help us in future projects too. Creating and running Hadoop MapReduce programs was quite challenging and we are glad to have learned how to use this handy tool. Being in our first semester, this project also helped us to learn how to efficiently manage time and resources.

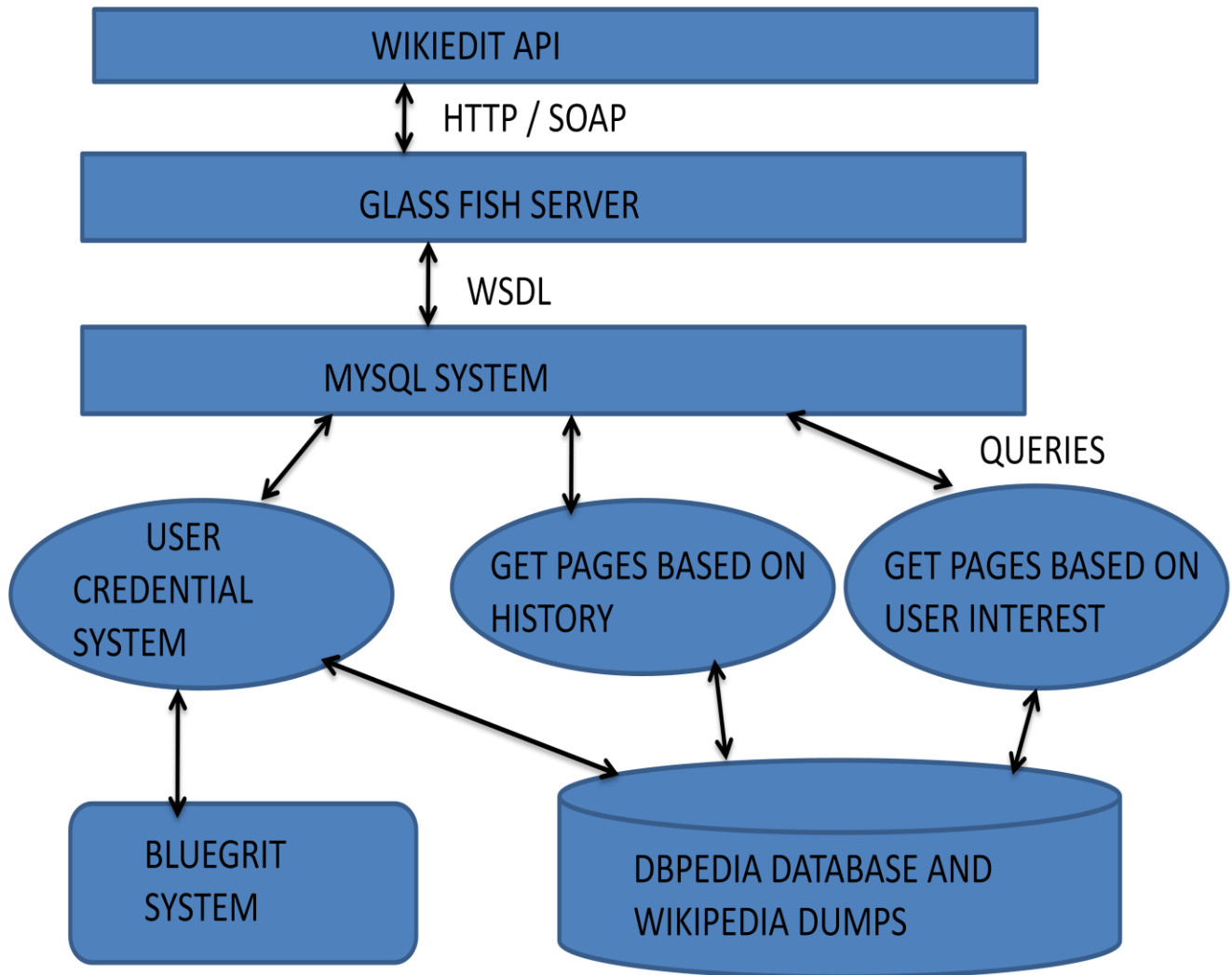
XV. REFERENCES

- [1] *User:SuggestBot - Wikipedia, the free encyclopedia.* (n.d.). Retrieved from <http://en.wikipedia.org/wiki/User%3ASuggestBot>
- [2] *Index of /enwiki/.* (n.d.). Retrieved December 20, 2013, from <http://dumps.wikimedia.org/enwiki/>
- [3] *SNAP: Network datasets: Wikipedia edit history.* (n.d.). Retrieved from <http://snap.stanford.edu/data/wiki-meta.html>
- [4] *wiki.dbpedia.org : About.* (n.d.). Retrieved from <http://dbpedia.org/About>
- [5] *wiki.dbpedia.org : Downloads 39.* (n.d.). Retrieved from <http://wiki.dbpedia.org/Downloads39#articles-categories>

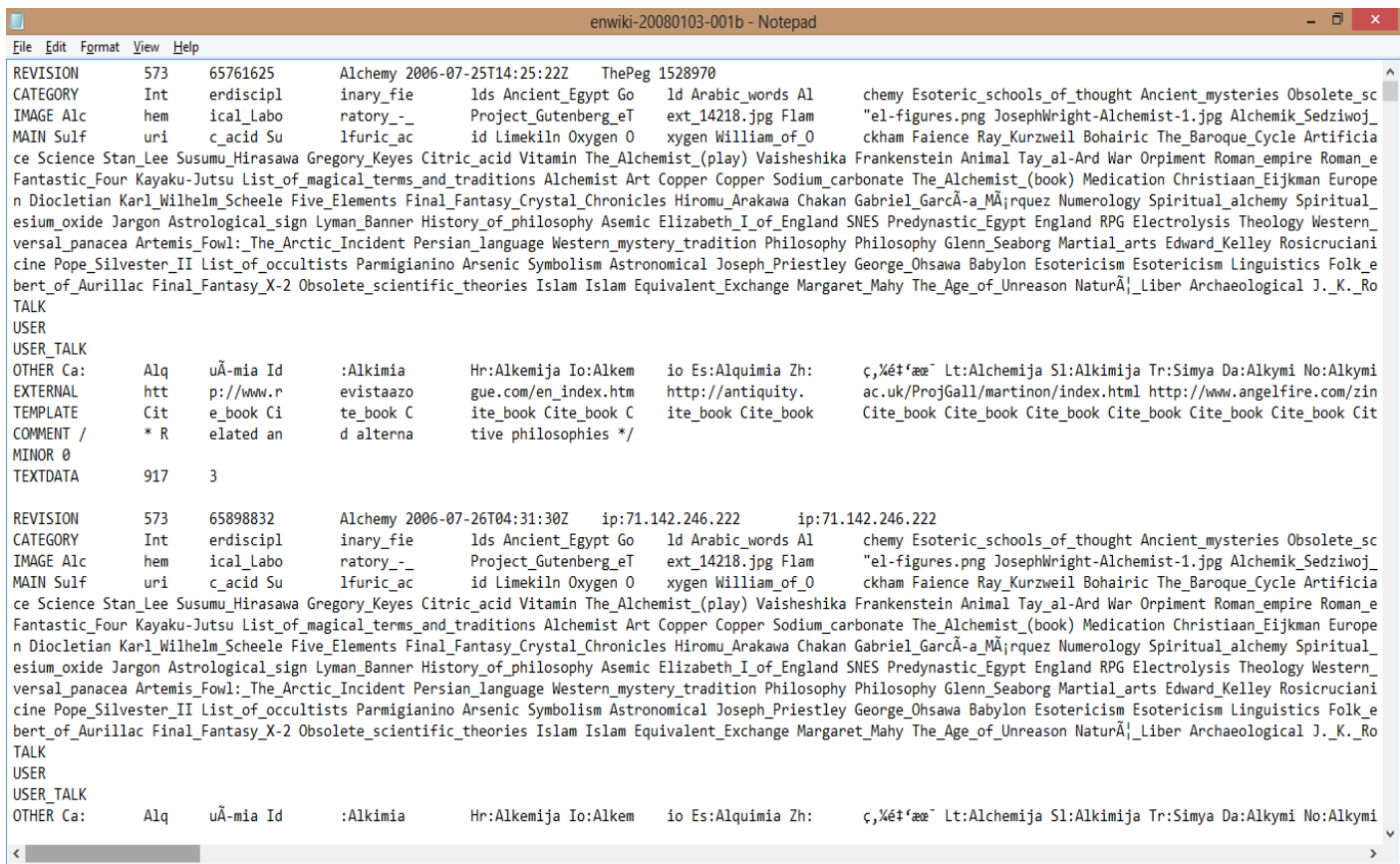
- [6] *wiki.dbpedia.org : Downloads 39*. (n.d.). Retrieved from <http://wiki.dbpedia.org/Downloads39#links-to-wikipedia-article>
- [7] P. Mendes, M. Jakob, A. García-Silva and C. Bizer, "DBpedia Spotlight: Shedding Light on the Web of Documents," *7th Int. Conf. on Semantic Systems*, Graz, Austria, Sept. 7-9, 2011.
- [8] GlassFish Server Open Source Edition, Release Notes, Release 4.0, May 2013.
- [9] D. Thiébaud, Y. Li, D. Jaunzeikare, *et. al*, "Processing Wikipedia Dumps: A Case-Study comparing the XGrid and Map Reduce Approaches," *Department of Computer Science, Smith College Northampton, MA, USA*, February 2011.
- [10] R. Engelen, "Pushing the SOAP Envelope With Web Services for Scientific Computing," *Department of Computer Science and School of Computational Science and Information Technology Florida State University*, Tallahassee.
- [11] J. Dean, S. Ghemawat "MapReduce: Simplified Data Processing on Large Clusters," *USENIX Association OSDI '04: 6th Symposium on Operating Systems Design and Implementation*.
- [12] *HadoopMap : Object Oriented Framework* (n.d.). Retrieved from <http://http://www.cs.colorado.edu/~kena/classes/5448/s11/presentations/hadoop.pdf>
- [13] Tyson Condie, Niel Conway, Peter Alvaro, Joseph M. Hellerstein, Khaled Elmeleegy, Russell Sears "*MapReduce Online*" Retrieved from <http://db.cs.berkeley.edu/papers/nsdi10-hop.pdf>

XVI. APPENDIX:

1. ARCHITECTURE DIAGRAM:



2. BIG DATASET:



2. WSDL:

```
<!--
Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Metro/2.3 (tags/2.3-7528; 2013-04-29T19:34:10+0000) JAXWS-RI/2.2.8 JAXWS/2.2 svn-revision#unknown.
-->
<!--
Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Metro/2.3 (tags/2.3-7528; 2013-04-29T19:34:10+0000) JAXWS-RI/2.2.8 JAXWS/2.2 svn-revision#unknown.
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://Wikipedia.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://Wikipedia.org/" name="wikiSearchService">
```

```

<types>
  <xsd:schema>
    <xsd:import namespace="http://Wikipedia.org/" schemaLocation="http://rakesh-
      pc:8080/WikiEdit/wikiSearchService?xsd=1"/>
  </xsd:schema>
</types>
<message name="wikiFind">
  <part name="parameters" element="tns:wikiFind"/>
</message>
<message name="wikiFindResponse">
  <part name="parameters" element="tns:wikiFindResponse"/>
</message>
<portType name="wikiSearch">
  <operation name="wikiFind">
    <input wsam:Action="http://Wikipedia.org/wikiSearch/wikiFindRequest"
      message="tns:wikiFind"/>
    <output wsam:Action="http://Wikipedia.org/wikiSearch/wikiFindResponse"
      message="tns:wikiFindResponse"/>
  </operation>
</portType>
<binding name="wikiSearchPortBinding" type="tns:wikiSearch">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
  <operation name="wikiFind">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
<service name="wikiSearchService">
  <port name="wikiSearchPort" binding="tns:wikiSearchPortBinding">
    <soap:address location="http://rakesh-pc:8080/WikiEdit/wikiSearchService"/>
  </port>
</service>
</definitions>

```

3. SOAP REQUEST:

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:wikiFind xmlns:ns2="http://Wikipedia.org/">
      <arg0>Mykenism</arg0>
    </ns2:wikiFind>
  </S:Body>
</S:Envelope>
```

4. SOAP RESPONSE:

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:wikiFindResponse xmlns:ns2="http://Wikipedia.org/">
      <return>http://en.wikipedia.org/wiki/Individualist_anarchism</return>
      <return>http://en.wikipedia.org/wiki/Panarchism</return>
      <return>http://en.wikipedia.org/wiki/Traditional_anarchism</return>
    </ns2:wikiFindResponse>
  </S:Body>
</S:Envelope>
```