Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.6, No.1, 2015

Voice Based Smart Internet Surfing for Blind using JADE Agent and HTML5 Developing Environment

Israa Akram Fadhil
Master of Science in Computer Science, UNIVERSITY OF BAGHDAD / College of Arts /Unit of Media and Information, IRAQ-Baghdad

Abstract
Internet nowadays represents the essential nerve for knowledge and information retrieval, millions of pages are published over the web every single day and valuable information is broadcasted to the public; with this expansion, and information retrieval became a hard task due to the numerous amounts of documents that have high degree of similarity when it comes to text matching schemes. Blind people are part of the communities and part of them are collapsed a way of the revolution of information due to absence of proper tools grants them the ability to surf the internet. This paper and many other efforts are targeting the objectivity of activating blind people interaction with the environment; this paper carries out an innovation scheme by deploying mutli-agent system to provide intelligent assistance services that help blind people getting into the proper page in surfing the internet. Knowledge from other sites are transferred to central unit which holds the experience and utilize it for the behalf of blind people. Multi-agent systems have been deployed and results showed knowledge developed to help blinds.
Keywords: HTML5, Web Speech API, Speech Based browser, Intelligent Agent, JADE, knowledge epistemology.

1- Introduction

The internet and web based resources are spanning huge number of knowledge domain through which users can develop their knowledge about certain subjects. Millions of valuable documents are published every day. Disable users (i.e., blind, parkensons and others who can't write or read) face a big challenge in surfing the internet and making use of its feedback due to the architecture of documents available on the internet and the mechanism used to search those documents [1, 2], anyway, search engines nowadays are providing other accessibility methods like search by image, voice. Anyway, voice interaction to a computer or a mobile machine is an ongoing project and developers are anxious to introduce new innovations in this field to eventually make electronic devices a live, knowledgeable, truthful and smart conversational port. The proliferation of voice technology, especially in voice recognition and intelligent conceptualization to the spoken words, adds new parameter to the systems driven by speech; this parameter involves knowledge development about the topic which the conversation is held about. In other words, computer or smartphone can update knowledge regarding the topic around which the conversation is held; this is accomplished by synchronizing with other computers and integrate their knowledge about the conversation [1, 2, 3].

Computer ability to recognize what we are saying and its ability to fetch knowledge about the topic we are talking about is a new interaction domain between human and machine. Brookes Talk [2] is a web browser developed in Oxford Brookes University in 90's. Brookes Talk provides function keys for accessing the web page. Brookes Talk reads out the webpage using speech synthesis in words, sentences and paragraph mode by parsing the web page content. It also uses some mechanism for searching the suitable results using search engines and supports a conceptual model of website too. It supports modeling of information on web page and summarizes the web page content.

Csurf [3] is developed by Stony Brook University. Csurf is context based browsing system. Csurf brings together content analysis, natural language processing and machine learning algorithm to help blind user to quickly identify relevant information. Csurf is composed of interface manager, context analyzer, browser object from tress processor and dialog generator. Csurf web browser uses the functionality of voice XML, JSAPI, freeTTS, Sphinx, JREXAPI, etc.

In this paper, two important factors are considered; these which affect the performance of internet explorer; they are the accuracy of speech recognition and the text matching for the retrieved documents, in matter of fact text similarity is not an easy issue to be tackled. In [1] an investigation is conducted to find out how to measure similarities in short text sentences; this issue adding a new challenge in the domain of interacting internet explorer using speech, where the speech recognition engine generates textual output with certain confidence and submit this text as query to the internet explorer and get the resultant list of links to matched documents. In this manner we have two measures that assess the quality of the interaction with the internet explorer: confidence degree in speech recognition and the methodology exploit to measure the similarity between generated text and the retrieved documents. In this paper we develop a model to promote the retrieving performance and the speech recognition quality by sharing knowledge developed at a point with other points

within the network. Multi-agent platform is deployed over the web to provide the infrastructure for knowledge integration as it will be seen in next sections.

2- Web Speech API

Before introducing the model of collaboration that used to integrate the knowledge over the web in speech based internet surfing, we need to introduce essential tool used all over the implementation of the proposed system; this tool is the Google Speech API [4].

Google has introduced web speech API to enable web application developers to embed speech interaction utility into their web applications. Text-to-speech and speech-to-text are new features incorporated in modern web based application such as voice web search, speech command interface, dialog systems, speech enabled email client and many others [3, 4].

The crucial factor that embarrassing the usage of web speech API as a reliable interface and interaction utility; is its capability to accurately interpret input speech. Confidence factor is provided by the web speech API as an assessment to the interpretation process. Web Speech API defines a complex interface whose structure is given in figure (1)

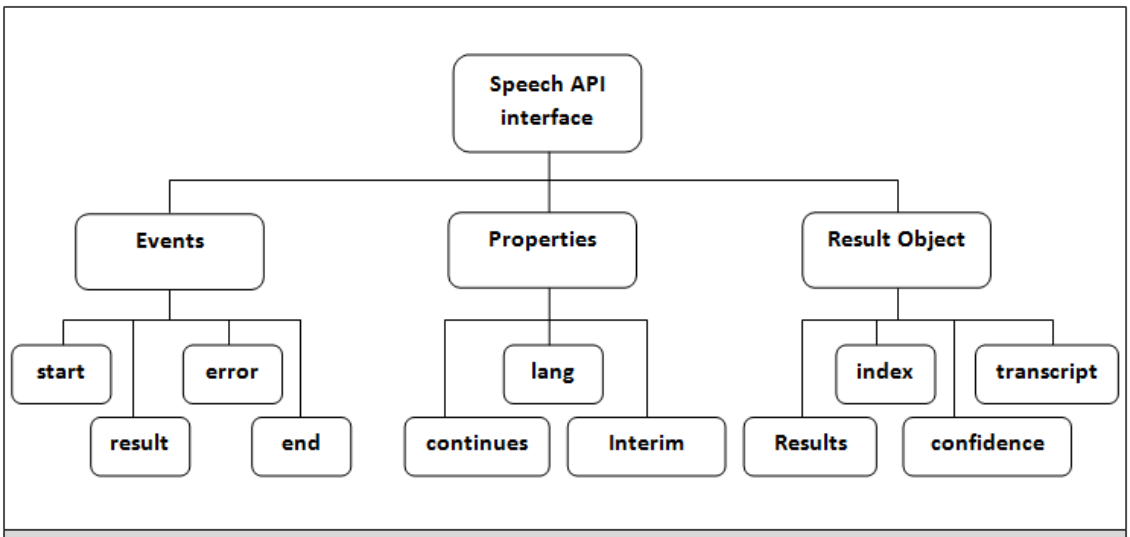

Figure 1: Google Speech API Specifications and Structure

Two important variables are shown in figure (1) which are 'interim' and 'confidence'; these variables crucial in assessing the performance of speech recognition methodology, where 'interim' refers to the intermediate results obtained from the recognition methodology [4]. When a word is in the interim, then the decision is not taken yet to announce this word as a valid recognition for certain speech with a specific 'confidence' degree which is a real number within the interval [0, 1].

3- Problem Statement and Motivations

- Most speech based browsers are built using Java applet technology which is not supported anymore by a wide range of modern internet explorers; this is due to the security constraints it violates.
- Google API develop the recognition from interim to final in stages where in the first stage many objects are resultants due to the speech recognition each with different probability for the recognition confidence. Google uses many paradigms to promote the recognition results from interim word with low confidence to final word with higher confidence.
- Knowledge about surfed documents is not transferred over the surfing session where knowledge can assist less knowledgeable people in getting the best match for their queries.

In this paper, multi-agent system is used to increase the reliability of transferring interim word to find word based on the query submitted and the selection of the user from the resultant lists.

4- Software Intelligent Agent and Developing Environment

Intelligent agent is a software module with significant features and attributes where it is defined as an entity that capable of perceiving events within the environment, and respond to perceived events through its effectors [6,7]. Software agent is autonomous, mobile, portable and platform independent; these features promote software module from component within the system to an associative and independent module that can interact the environment and enhance the overall performance of the targeted software systems. In 1996 [6,7,8] a Swiss non-

profit organization has introduced the specifications for the creation, registration, location, communication, migration and retirement of agents. In communication a dedicated language has been introduced which is ACL (Agent Communication Language) through which agents can live in communities and conduct socialization sessions [8].

Anyway, to develop a software agent a development environment is to be used; this environment is called JADE (Java Agent Development Environment); this environment provides tools that can be utilized to build and run agent modules. A distributed JADE platform is composed of several run-time containers launched on one or more hosts on a network. Each hardware device typically executes one Java Virtual Machine (JVM), which can host one or more JADE instances; inter-JVM communication is based on JAVA Remote Method Invocation (RMI) [9, 10].

Agents are encapsulating their functionalities in these behaviors (i.e., behavior is a terminology of Agent programming terminologies and it refers to actions conducted by Agents and when these actions are fired) [6,10].

Behaviors run in different schemes of synchronization but they share the same triggering event, which is the arrival of ACL Message to the Message queue of the platform, in the proposed system multiple threads are required to tackle behaviors' functionality; this was a challenge to be accomplished in a client side web application (i.e., web page) due to the limitation of scripting language (i.e., jQuery and JavaScript). Javascript is a single threaded scripting language, thus, it does not fulfill the requirement.

HTML5 introduces many facilities to enforce multi-threading programming environment; Web Worker is the essential element in this issue. Web worker spawn piece of code, which is JavaScript code in this proposal, to work in different thread and does not affect the overall latency of the program.

Web worker is designed to carry out heavy weight process in the background but does not have access to the foreground components (i.e., DOM - Document Object Model); this due to the sophisticated synchronization model needed to manage accessing DOM objects from multiple Web Workers.

5- The proposed Voice based Information Broker

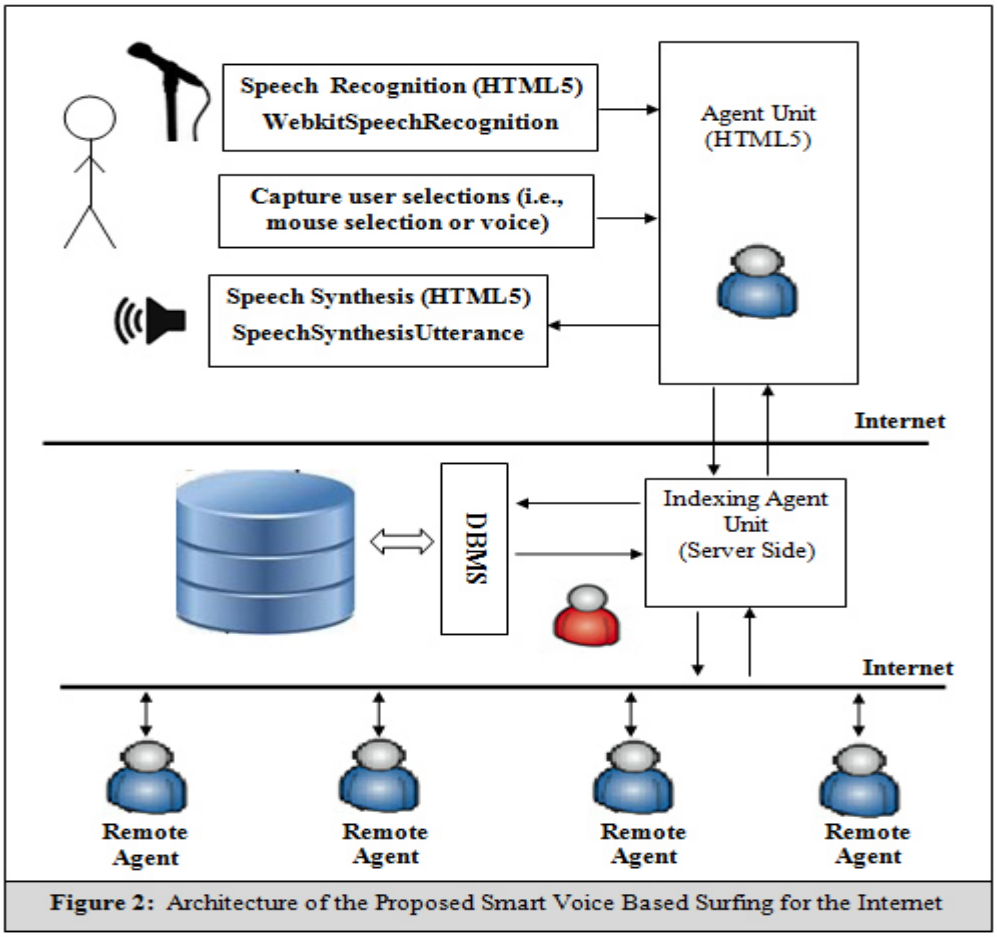

Figure 2: Architecture of the Proposed Smart Voice Based Surfing for the Internet

In this proposal we aim to promote the experience of the blind or disabled user who surfs the internet looking for knowledge regarding certain subject. The experience of others is to be integrated by deploying multi-

agent platform as it is shown in figure (2)

As in figure (2), agents are implemented as javascript module at the client side to collection knowledge about user experience in surfing the internet at his/her machine. Agent is designed to Google speech API to read two important data: first the transcript generated for the speech, and second the confidence of generating that transcript, as it is shown in table(1)

Table – 1 –
Interpreted Voice and Confidence

Index	Transcript	confidence
1	Hello	0.6204143166542053
2	How to Start a Computer	0.5246079564094543
3	How are you	0.6929399371147156
4	What's going on	0.5550333857536316
5	Mar	0.5158120393753052
6	My name means	0.6032026410102844
7	What is yours	0.5905967354774475
8	Tell me	0.5533714294433594
9	I think its a bad boss the pasta song	0.6197972297668457
10	I'm talking to the computer how to use	0.6166571234532916

As we are after surfing the internet, the generated transcript is submitted to internet explorer as query as it is presented in figure (3) where results are polled from the server. The problem here is the number of results for certain query and the confidence of the transcript.

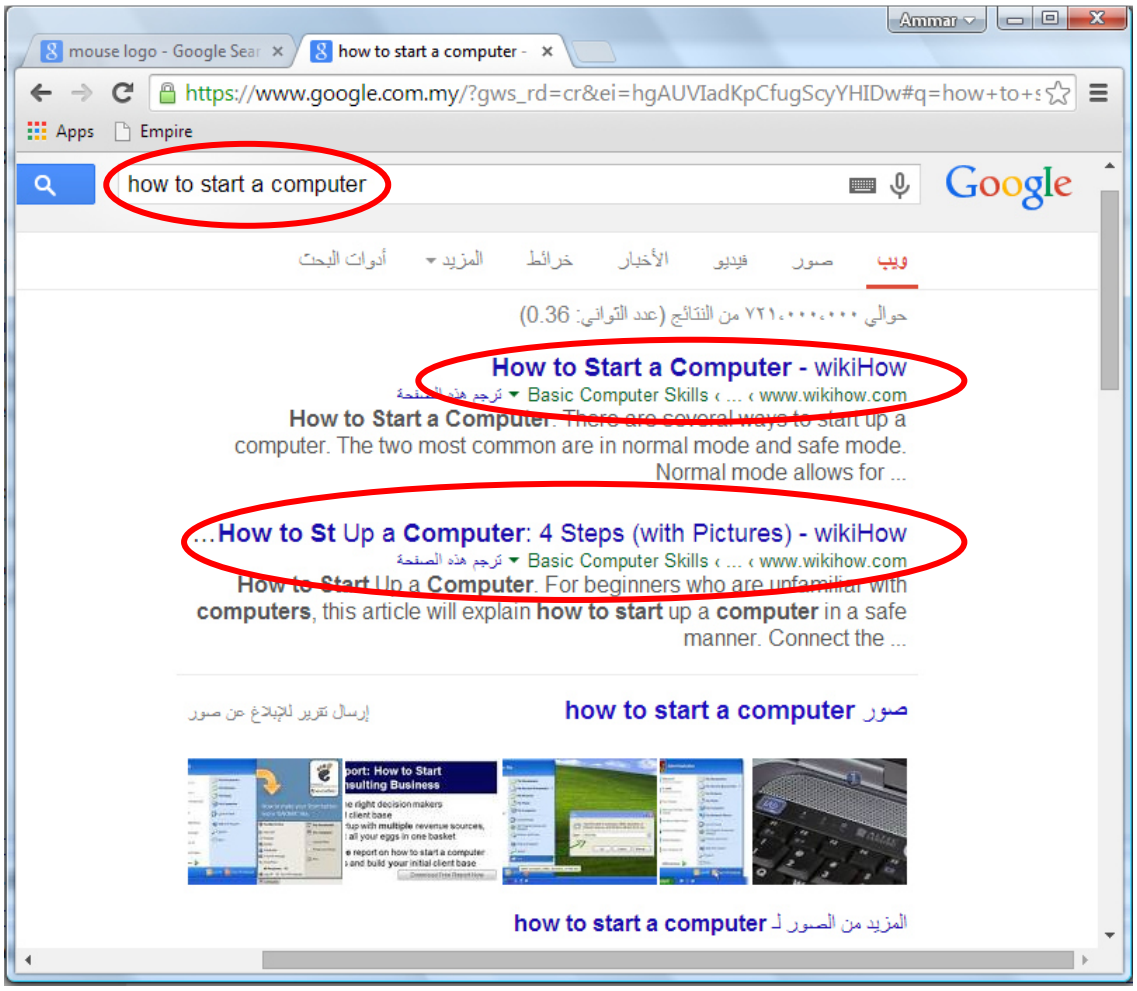

Figure 3: Resultants of Submitting Query to Google Search Engine

As it is shown in figure (3), results are obtained due to submitting query which is a transcription for the speech spilled out by the user. The resultant text is priorities based on the text similarities between the query from a side and the obtained documents from the other side; this is done at each client side. When multiple agents collects experience and send it to the platform administrator a list of records is generated as it shown in table (2)

Table – 2 –
Results for 'How to Start a Computer ' Query

I	URL	confidence	similarity	visited	Visit-ratio
1	http://www.wikihow.com/Start-a-Computer	0.62041160790318	0.8901101	5	0.357143
2	http://www.wikihow.com/Start-Up-a-Computer	0.53038211835612	0.7773924	4	0.285714
3	http://www.tldp.org/HOWTO/Unix-and-Internet-Fundamentals-HOWTO/bootup.html	0.52460795640945	0.6938434	1	0.071429
4	http://computer.howstuffworks.com/pc3.htm	0.58110305326122	0.6656345	1	0.071429
5	http://www.answers.com/Q/What_are_the_steps_to_start_a_computer	0.49881031191038	0.8883432	2	0.142857
6	http://support.kaspersky.com/general/various/493	0.58816324324866	0.6594394	1	0.071429

In vector space, the vector that represents the state for retrieved documents and its relation to the transcript generated for the speech is given by the following:

$$DocState = Confidence\ X + Similarity\ Y + Visit\_Ratio\ Z \quad \text{eq.1}$$

$$Weight = |DocState| = \sqrt{Confidence^2 + Similarity^2 + Visit\_Ratio^2} \quad \text{eq.2}$$

The results of the calculations is shown in figure (4) where URLs are weighted based on the knowledge exchanged among agents

From figure (4), it is obvious that " http://www.wikihow.com/Start-a-Computer " is the most candidate document to be read due to submitting "how to start a computer" query. Other documents are prioritized based on their state vector norm.

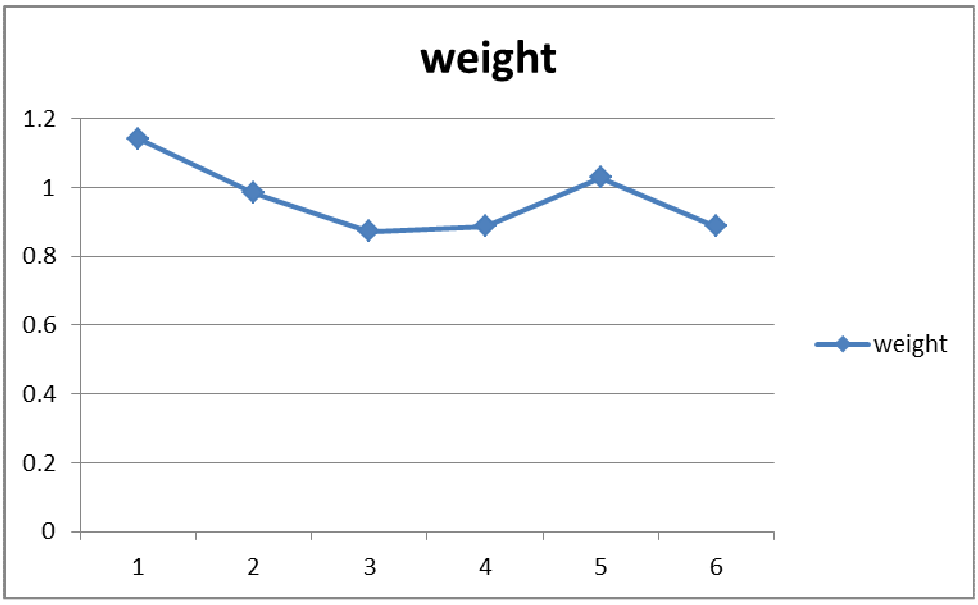

Figure 4: URLs are weighted according to Stat Vector Norms

Conclusions

Surfing the internet using speech has many complications due to the accuracy of the speech recognition tools; this is from a side and the huge number of matching text from the other side. These complications are getting

harder to resolved when it comes to blind people where results are hard to be visited, thus knowledge epistemology can provide a great help as it has been concluded in this paper. Multi-agent system helps very much the transformation of knowledge from people who can see to people who can't see; this approach also introduces new trend for increasing the performance of speech recognition where text is manipulated in the interim stage before reaching the final; in this manipulation knowledge is integrated from other sources reach out higher confidence values, as an extension to this work a proposal can be put together to promote the quality and performance of speech based internet surfing.

References

1- Prabhat Verma, Raghuraj Singh and Avinash Kumar Singh, " A framework to Integrate Speech Based Interface for Blind Web users on the Websites of Public Interest", Springer, 2013, accessed on: http://www.hcis-journal.com/content/3/1/21

2- Zajicek M, Powel C, Reeves C (1999) Web search and orientation with brookestalk. In: Proceedings of tech. and Persons with disabilities Conf. 1999. Los Angeles: (CSUN'99).

3- Mahmud J, Borodin Y, Ramakrishnan IV (2007) Csurf: A context-driven non-visual web-browser. Proceedings of the International Conference on the World Wide Web (WWW'07), Banff, Alberta, Canada, May 08-12, 2007.

4- Web Speech API Specification, W3C Community Group, reached at https://dvcs.w3.org/hg/speech-api/raw-file/tip/speechapi.html , 2012.

5- Donald Metzler, Susan Dumais, and Christopher Meek, "Similarity Measures for Short Segments of Text", Microsoft Research Center, 2008

6- Amani Salah Eldin Abdalaziz, Prof. Mohamed Shouman, Prof. Ibrahim Elhenawy, (2009), "A Multi Agent-Based Framework for Network Intelligence and Intrusion Prevention", Cairo. [Accessed on 12 May 2012], Available at: <http://www.dtic.mil/dtic/tr/fulltext/u2/a417590.pdf>

7- J. Lind, (2000) "Issues in Agent-Oriented Software Engineering". *The First International Workshop on Agent-Oriented Software Engineering*, Limerick/Ireland

8- P.Bresciani, A. Perini, P. Giorgini, F. Giunchiglia and J. Mylopoulos, (2006) "A knowledge level software engineering methodology for agent oriented programming," *ACM Proceedings of the fifth international conference on Autonomous agents,* vol. 12, pp. 648 – 655

9- R. Bordini, L. Braubach, M. Dastani, , J. Sanz, J. Leite, G. O'Hare, A. Pokahr, and A. Ricci, (2006) "A survey of programming languages and platforms for multiagent systems", *Informatica*, vol. 30, pp. 33-44.

10-William John Teahan, " Artificial Intelligence – Agents and Environments", Ventus Publshing ApS, ISBN: 978-87-7681-528-8, 2010.

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:
http://www.iiste.org

CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

Prospective authors of journals can find the submission instruction on the following page: http://www.iiste.org/journals/ All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: http://www.iiste.org/book/

Academic conference: http://www.iiste.org/conference/upcoming-conferences-call-for-paper/

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar