American Journal of Applied Sciences 7 (10): 1406-1411, 2010
ISSN 1546-9239

One-Match and All-Match Categories for Keywords Matching in Chatbot

Abbas Saliimi Lokman and Jasni Mohamad Zain
Faculty of Computer Systems and Software Engineering,
University Malaysia Pahang, Lebuhraya Tun Razak, 26300 Kuantan, Pahang, Malaysia

Abstract: Problem statement: Artificial intelligence chatbot is a technology that makes interactions between men and machines using natural language possible. From literature of chatbot's keywords/pattern matching techniques, potential issues for improvement had been discovered. The discovered issues are in the context of keywords arrangement for matching precedence and keywords variety for matching flexibility. **Approach:** Combining previous techniques/mechanisms with some additional adjustment, new technique to be used for keywords matching process is proposed. Using newly developed chatbot named ViDi (abbreviation for Virtual Diabetes physician which is a chatbot for diabetes education activity) as a testing medium, the proposed technique named One-Match and All-Match Categories (OMAMC) is being used to test the creation of possible keywords surrounding one sample input sentence. The result for possible keywords created by this technique then being compared to possible keywords created by previous chatbot's techniques surrounding the same sample sentence in matching precedence and matching flexibility context. **Results:** OMAMC technique is found to be improving previous matching techniques in matching precedence and flexibility context. This improvement is seen to be useful for shortening matching time and widening matching flexibility within the chatbot's keywords matching process. **Conclusion:** OMAMC for keywords matching in chatbot is shown to be an improvement over previous techniques in the context of keywords arrangement for matching precedence and keywords variety for matching flexibility.

Key words: Chatbot, artificial linguistic internet computer entity, artificial intelligence markup language, VPbot, database management system, hypertext preprocessor, relational database model, hypothetically

INTRODUCTION

In 1950, mathematician Alan Turing proposed the question "Can machines think?" (Turing, 2008). Since then, a number of attempt to encounter that particular question have been emerged in computer science field that later formed the field of Artificial Intelligence. One of many attempts to visualize an intelligence machine is chatbot or chatter robot. Chatbot is a technology that makes interaction between man and machine using natural language possible. First introduced by Weizenbaum (an MIT professor) in 1966 (Weizenbaum, 1966), the first chatbot named ELIZA then famously became an inspiration for computer science and linguistic researchers in creating a computer application that can hypothetically understand and response to natural human language. The huge breakthrough in chatbot technology came in 1995 where Dr. Richard Wallace, an ex-Professor of Carnegie Mellon University combine his background in computer science with his interest in the internet and natural language processing to produce Artificial Linguistic Internet Computer Entity (ALICE) (Wallace, 2008). ALICE that later being described as a modern ELIZA is a three times winner of Loebner's annual instantiation of Turing's Test for machine intelligence (Shah, 2006). When computer science evolves, so does the chatbot technology. As for a chatbot that need to have a wide data storage for its knowledge-based (some call it "chatbot's brain"), managing data is really a crucial issue. Reviewing the evolving of chatbot technology surrounding the evolving of computer science technology, ELIZA stored its knowledge-based data by directly embedding it into the application's code while later chatbot ALICE uses more advance Artificial Intelligence Markup Language (AIML) which is a derivative of Extensible Markup Language or XML to stored the knowledge-based data (Shawar and Atwell, 2007; Wallace, 2008). Then with the emerging of Relational Database Model together with Database

Corresponding Author: Abbas Saliimi Lokman, Faculty of Computer Systems and Software Engineering,
University Malaysia Pahang, Lebuhraya Tun Razak, 26300 Kuantan, Pahang, Malaysia

Management System (DBMS) technology, came more advance chatbots that taking advantages of it. One of an example is VPbot and SQL-Based chatbot for medical application (Ohno-Machado and Webber, 2005). Developed by Dr. Webber from Harvard University, VPbot is a chatbot that takes advantage of a Relational Database Model to stored, manage and even used the SQL language (database scripting language) to perform the chatbot main process which is keywords/pattern matching.

Reviewing ELIZA's keywords matching technique, an input sentence is analyzed from left to right. Each word is looked up in a dictionary of keywords for a match and if word/s is identified as keywords, then decomposition rule will apply (Weizenbaum, 1966) (note that decomposition rule is a method used by ELIZA in the process of reassembly rule or response generation). For ALICE, its knowledge about English conversation is stored using a mechanism called Graphmaster (written using AIML). The Graphmaster consists of collection of nodes called Nodemappers. These Nodemappers map the branches from each node. The branches are either single words or wildcards. A convenient metaphor for AIML patterns is the file system stored in computers that are organized hierarchically (tree structure). The file system has a root, such as "c:\" and the root have some branches that are files and some that are folders. The folders, in turn, have branches that are both folders and files. The leaf nodes of the whole tree structure are files. Every file has a "path name" that spells out its exact position within the tree. The Graphmaster is organized in exactly the same way. AIML that stored a pattern like "I LIKE TO *" is metaphorically are "g:/I/LIKE/TO/star". All of the other patterns that begin with "I" also go into the "g:/I/" folder. All of the patterns that begin with "I LIKE" go in the "g:/I/LIKE/" subfolder. So it's like the folder "g:/I/LIKE/TO/star" has a single file called "template.txt" that contains the template (Shawar and Atwell, 2007; Wallace, 2008).

Following Graphmaster rules, A.L.I.C.E pattern matching process can be described as follows (let say the input utterance first word is "yesterday" and the AIML is described as file system architecture with folders and files):

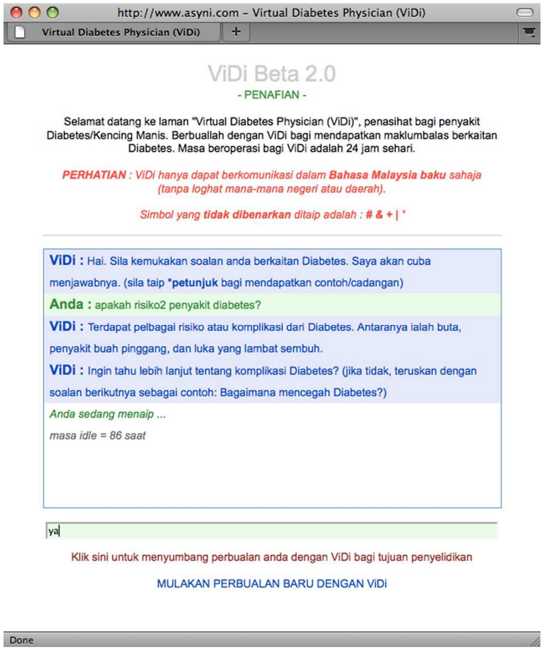
- From template file in the root folder, find a match pattern. If no match was found, try
- Find the subfolder "_". If found, try matching all remaining suffixes from the input utterance following the first word "yesterday" (the whole input utterance). If no match was found, try
- Find the subfolder "yesterday". If found, try matching all remaining suffixes minus "yesterday". If no match was found, try
- Find the subfolder "*". If found, try matching all remaining suffixes from the input utterance following the first word "yesterday". If no match found, change directory to the parent of this folder and put back "yesterday" on the head of the input

These processes will run recursively until the input is null (all words in the input utterance have been processes), or until the match is found, making the process to stop.

As a recap, chatbot's keywords/pattern matching techniques can be divided into two categories. First is rather similar to human brain incremental parsing technique (Crocker *et al*., 1999) where an input sentence is being analyzed in a word-by-word basis from left to right by sequence. Keywords can be one-word keywords or many-words keywords but each word in many-words keywords must be attached to one another, forming a long keywords pattern (cannot be separated as e.g., one word in prefix and one word in suffix separated by several words in the middle). Second is a direct match process where input sentence is being analyzed for an appearance of keywords anywhere in the input sentence. Whole input sentence is being treated as a one variable and available keywords in the database will scan this variable for match. The principal difference between first and second technique is first being input centered (words from input sentence is being matched against keywords in knowledge-based) and second being keywords centered (keywords in knowledge-based is being matched against an input sentence). Despite the difference, both categories suggested the same paradigm for matching process in which only one keywords is needed in order to trigger the respective response. One keywords in this context means one word, phrase or even sentence for one keywords set (not a collection of word, phrase or sentence). However, there is an augment regarding this matter by VPbot's keywords architecture/design. In VPbot, author can assign several keywords (maximum of three) in the same keywords set. All keywords within the same set must be matched in order to trigger the respective response (Ohno-Machado and Webber, 2005). Using the second category of keywords matching technique, all keywords can be located anywhere in the input sentence and as long as the keywords is in the same set, VPbot will matched it. For the issue of precedence over which keywords is more accurate, longer keywords appear to have the top priority justified by long keywords set will only match a very specific phrase, while short keywords set will match a larger range of possible input queries (Ohno-Machado and Webber, 2005).

MATERIALS AND METHODS

To test the proposed technique of One-Match and All-Match Categories (OMAMC), we had designed and developed a new chatbot named Virtual Diabetes physician (ViDi), a web-based chatbot that functions in the specific domain of Diabetes education. Taking advantage of Relational Database Model approach, we redesign the whole architecture of chatbot's keywords by incorporating the proposed technique into it. In technical details, ViDi is being coded using Hypertext Preprocessor (PHP) programming language together with Asynchronous Javascript + XML (AJAX) technology which contains Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), XMLHttp Request (XHR) and Document Object Model (DOM) being accessed via JavaScript. Figure 1-3 shows ViDi's UI (User Interface) design. Fig. 1 is a chatting UI for users while Fig. 2 and 3 are knowledge-based (responses and keywords) management UI known as vBrain (developed for authors). Note that ViDi is a Bahasa Malaysia human language chatbot (Lokman and Zain, 2010a) and that being the case, contents presented in each UI are mostly originated from this language.



http://www.asyni.com – Virtual Diabetes Physician (ViDi)

Virtual Diabetes Physician (ViDi)

ViDi Beta 2.0

- PENAFIAN -

Selamat datang ke laman "Virtual Diabetes Physician (ViDi)", penasihat bagi penyakit Diabetes/Kencing Manis. Berbuallah dengan ViDi bagi mendapatkan maklumbalas berkaitan Diabetes. Masa beroperasi bagi ViDi adalah 24 jam sehari.

*PERHATIAN : ViDi hanya dapat berkomunikasi dalam **Bahasa Malaysia baku** sahaja (tanpa loghat mana-mana negeri atau daerah).*

*Simbol yang **tidak dibenarkan** ditaip adalah : **# & + | '***

ViDi : Hai. Sila kemukakan soalan anda berkaitan Diabetes. Saya akan cuba menjawabnya. (sila taip ***petunjuk** bagi mendapatkan contoh/cadangan)

Anda : apakah risiko2 penyakit diabetes?

ViDi : Terdapat pelbagai risiko atau komplikasi dari Diabetes. Antaranya ialah buta, penyakit buah pinggang, dan luka yang lambat sembuh.

ViDi : Ingin tahu lebih lanjut tentang komplikasi Diabetes? (jika tidak, teruskan dengan soalan berikutnya sebagai contoh: Bagaimana mencegah Diabetes?)

Anda sedang menaip ...

masa idle = 86 saat

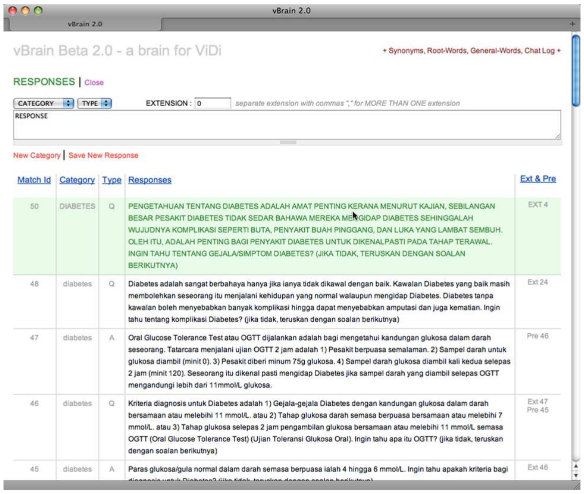
ya

Klik sini untuk menyumbang perbualan anda dengan ViDi bagi tujuan penyelidikan

MULAKAN PERBUALAN BARU DENGAN ViDi

Fig. 1: ViDi chatting interface

OMAMC technique comprises of two components that correlated with each other. The two components are (1) keywords arrangement for matching precedence and (2) keywords variety for matching flexibility. Describing the fundamental idea of OMAMC, each response in ViDi's knowledge-based is designed to have an infinite number of keywords sets associated with either One-match or All-match category. Each keywords set in One-match category contains single keywords that can be in a form of one-word or many-words keywords (a single word or a phrase) while each keywords set in All-match category contains more than single keywords as in VPbot's keywords design.



vBrain 2.0

vBrain Beta 2.0 - a brain for ViDi

+ Synonyms, Root-Words, General-Words, Chat Log +

RESPONSES | Close

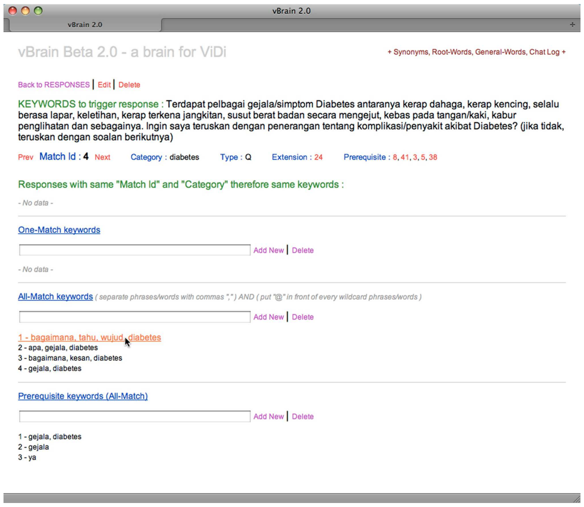
CATEGORY TYPE EXTENSION : 0 *separate extension with commas "," for MORE THAN ONE extension*

RESPONSE

New Category | Save New Response

Match Id	Category	Type	Responses	Ext & Pre
50	DIABETES	Q	PENGETAHUAN TENTANG DIABETES ADALAH AMAT PENTING KERANA MENURUT KAJIAN, SEBILANGAN BESAR PESAKIT DIABETES TIDAK SEDAR BAHAWA MEREKA MENGIDAP DIABETES SEHINGGALAH WUJUDNYA KOMPLIKASI SEPERTI BUTA, PENYAKIT BUAH PINGGANG, DAN LUKA YANG LAMBAT SEMBUH. OLEH ITU, ADALAH PENTING BAGI PENYAKIT DIABETES UNTUK DIKENALPASTI PADA TAHAP TERAWAL. INGIN TAHU TENTANG GEJALA/SIMPTOM DIABETES? (JIKA TIDAK, TERUSKAN DENGAN SOALAN BERIKUTNYA)	EXT 4
48	diabetes	Q	Diabetes adalah sangat berbahaya hanya jika ianya tidak dikawal dengan baik. Kawalan Diabetes yang baik masih membolehkan seseorang itu menjalani kehidupan yang normal walaupun mengidap Diabetes. Diabetes tanpa kawalan boleh menyebabkan banyak komplikasi hingga dapat menyebabkan amputasi dan juga kematian. Ingin tahu tentang komplikasi Diabetes? (jika tidak, teruskan dengan soalan berikutnya)	Ext 24
47	diabetes	A	Oral Glucose Tolerance Test atau OGTT dijalankan adalah bagi mengetahui kandungan glukosa dalam darah seseorang. Tatarcara menjalani ujian OGTT 2 jam adalah 1) Pesakit berpuasa semalaman. 2) Sampel darah untuk glukosa diambil (minit 0). 3) Pesakit diberi minum 75g glukosa. 4) Sampel darah glukosa diambil kali kedua selepas 2 jam (minit 120). Seseorang itu dikenal pasti mengidap Diabetes jika sampel darah yang diambil selepas OGTT mengandungi lebih dari 11mmol/L glukosa.	Pre 46
46	diabetes	Q	Kriteria diagnosis untuk Diabetes adalah 1) Gejala-gejala Diabetes dengan kandungan glukosa dalam darah bersamaan atau melebihi 11 mmol/L. atau 2) Tahap glukosa darah semasa berpuasa bersamaan atau melebihi 7 mmol/L. atau 3) Tahap glukosa selepas 2 jam pengambilan glukosa bersamaan atau melebihi 11 mmol/L semasa OGTT (Oral Glucose Tolerance Test) (Ujian Toleransi Glukosa Oral). Ingin tahu apa itu OGTT? (jika tidak, teruskan dengan soalan berikutnya)	Ext 47 Pre 45
45	diabetes	A	Paras glukosa/gula normal dalam darah semasa berpuasa ialah 4 hingga 6 mmol/L. Ingin tahu apakah kriteria bagi diagnosis untuk Diabetes? (jika tidak, teruskan dengan soalan berikutnya)	Ext 46

Fig. 2: vBrain-managing ViDi's response



vBrain 2.0

vBrain Beta 2.0 - a brain for ViDi

+ Synonyms, Root-Words, General-Words, Chat Log +

Back to RESPONSES | Edit | Delete

KEYWORDS to trigger response : Terdapat pelbagai gejala/simptom Diabetes antaranya kerap dahaga, kerap kencing, selalu berasa lapar, keletihan, kerap terkena jangkitan, susut berat badan secara mengejut, kebas pada tangan/kaki, kabur penglihatan dan sebagainya. Ingin saya teruskan dengan penerangan tentang komplikasi/penyakit akibat Diabetes? (jika tidak, teruskan dengan soalan berikutnya)

Prev Match Id : 4 Next Category : diabetes Type : Q Extension : 24 Prerequisite : 8, 41, 3, 5, 38

Responses with same "Match Id" and "Category" therefore same keywords :

- No data -

One-Match keywords

Add New | Delete

- No data -

All-Match keywords *(separate phrases/words with commas ",") AND (put "@" in front of every wildcard phrases/words)*

Add New | Delete

1 - bagaimana, tahu, wujud, diabetes
2 - apa, gejala, diabetes
3 - bagaimana, kesan, diabetes
4 - gejala, diabetes

Prerequisite keywords (All-Match)

Add New | Delete

1 - gejala, diabetes
2 - gejala
3 - ya

Fig. 3: vBrain-managing ViDi's keywords

The different is that ViDi’s All-match keywords had no limit over how many keywords can a single set have (VPbot limitation is three keywords for each single set). Therefore, All-match keywords can be in a form of combination between a single word and a phrase, producing either multiple one-words keywords, multiple phrases keywords or both one-word/s and phrase/s keywords in the same single keywords set. For both OMAMC, each keywords set will be stored as a single variable. Therefore for All-match category that can have multiple keywords within the same set, author need to put a symbol of commas (“,”) to separate each keywords. For matching process, One-match is considered to be an exact-match process where word/s and its location must be the same as in the input sentence, while All-match is considered to be a flexible-match where word’s location is a flexible factor. Same as VPbot’s keywords matching technique, if each All-match keywords within the same set is matched, it will then trigger the response. The sequence location of the keywords can be different between the set and the input sentence. As example, first and second keywords in the set do not have to be in the same sequence location as in the input sentence (in the input sentence, the second keywords can came first before the first keywords).

Looking back to the two components of OMAMC (keywords arrangement for matching precedence and keywords variety for matching flexibility), keywords arrangement for this technique is designed based on keywords precedence that is as in literature, long keywords over short keywords (note that the length of keywords is defined by total count of words within the set) and exact-match over flexible-match (generic keywords) that is One-match over All-match. For keywords variety, OMAMC technique had expanse VPbot’s technique on generic keywords by making no limitation on the number of keywords that can be associated with a single set.

RESULTS

Table 1-3 will demonstrate results for the same sample of input sentence being converted into several possible keywords sets using Graphmaster-AIML technique, VPbot technique and OMAMC technique presented by three respective tables. Note that keywords variations for each technique can be more than as presented but given the purpose of analyzing the capability and limitation for each technique, such variations is considered to be not essential. The sample input sentence is “Yesterday, my chest hurt badly”.

Table 1: AIML result

<pattern>	YESTERDAY MY CHEST HURT BADLY * MY CHEST HURT BADLY MY CHEST HURT BADLY MY CHEST HURT * MY CHEST HURT * CHEST HURT BADLY CHEST HURT BADLY CHEST HURT	<pattern>

Table 2: VPbot result

Keyword 1	Keyword 2	Keyword 3
YESTERDAY MY CHEST HURT BADLY		
YESTERDAY	MY CHEST	HURT BADLY
YESTERDAY	MY CHEST	HURT
MY CHEST HURT BADLY		
MY CHEST	HURT	
MY CHESTHURT	BADLY	
MY CHEST	HURT	
MY	CHEST	HURT
CHEST HURT BADLY		
CHEST HURT		
CHEST	HURT	BADLY
CHEST	HURT	

Table 3: OMAMC result

One-match

YESTERDAY MY CHEST HURT BADLY
MY CHEST HURT BADLY
MY CHEST HURT

All-match

Keyword 1	Keyword 2	Keyword 3	Keyword 4	Keyword n
YESTERDAY	MY CHEST	HURT	BADLY	
YESTERDAY	MY	CHEST	HURT	BADLY
MY CHEST	HURT	BADLY		
MY	CHEST	HURT	BADLY	
MY	CHEST	HURT		
CHEST HURT BADLY				
CHEST HURT				
CHEST	HURT	BADLY		
CHEST	HURT			

DISCUSSION

Presented results are possible keywords database for three separated keywords storing technique. The first issue to be analyzed is precedence. For AIML with Graphmaster component, precedence for keywords goes by atomic categories (exact-match), then default categories (pattern with wildcard/s) and later recursive categories (symbolic reduction, synonyms replacement). To be noted that in AIML, longer keywords will not affect the precedence level. For VPbot, all keywords will be matched first before precedence analysis is being done.

VPbot precedence goes by specific instance over generic response (exact-match over flexible-match), variation with low total weighs over high total weights (symbolic reduction, synonyms replacement) and later total string length (longer string over shorter string). For both techniques, exact-match is considered to be the highest precedence over all keywords. As such, in OMAMC technique, exact-match keywords is treated in a totally different category from generic keywords (flexible-match) with One-match being the exact-match and All-match being the flexible-match. Being in different group, if algorithm finds a match in One-match category, then All-match category will not be processed. This scenario will result on the elimination of redundant matching time for less precedence keywords if more precedence keywords had already matched. Next if no match is found within One-match category, then algorithm will proceed to generic keywords category, which is All-match category. With strong argument by VPbot that longer string length have more precedence over short string length, One-match and All-match keywords had built in attached variable name "wordCount" to encounter this issue. In each category according to precedence (One-match then All-match), wordCount will be among the first to be analyzed in order to avoid unnecessary matching process. That is if a match is found, wordCount for that keywords will be hold as a benchmark for string length. Therefore, algorithm will not process keywords with less count of words than already matched keywords, eliminating the need for unnecessary matching process for keywords that eventually will not be used.

The second issue to be analyzed is matching flexibility, which is created by generic keywords technique. AIML did not have the support for generic keywords while VPbot had the limit of maximum three keywords for each set (keywords 1, 2 and 3). For All-match category, generic keywords had no limit in quota (keywords 1 to n). Same rule as VPbot is applied where all keywords within the same set must be matched in order to trigger the response. As shown in Results section, more quotas on generic keywords can produced more keywords variety for matching flexibility.

CONCLUSION

ViDi is designed and developed to functions as virtual diabetes physician for diabetic patients and public to learn about diabetes disease. Several additional techniques and/or algorithms had been proposed in attempt to enhance ViDi's productivity in becoming the virtual helpdesk for diabetes education domain (Lokman and Zain, 2010b). OMAMC technique is proposed to enhanced ViDi's keywords matching technique in the context of keywords arrangement for matching precedence and keywords variety for matching flexibility. Presented results and discussion had demonstrated the result in using this technique against previous techniques, showing improvement in keywords matching precedence and its flexibility in the process.

Other area in which OMAMC technique can be implemented is in Information Extraction (IE) application. As proposed by Christy and Thambidurai (2008), additional algorithms can be useful in performing IE process. Using OMAMC technique logic, input keywords from user can be transformed into several keywords varieties (in respect to OMAMC format) in order to make retrieval process results have the value of precedence (based on the matched keywords categories). This value later can be used for results representation. Computer hardware processing algorithms also had involved in string matching process algorithms (Raju and Babu, 2007). In this area, further research can be done into making the two categories of OMAMC being process in two different string maching algorithms with One-match category being directly match without preprocessing phase and All-match category being match with preprocessing phase (because the flexible matching process of generic keywords). Differentiating these two processes could result in (1) faster processing time by the logic that All-match category did not have to be matched if One-match category already found a match and (2) maintaining matching flexibility for generic keywords category (All-match category) while still concerning the processing time for exact match keywords category (One-match category). From interconnectivity between OMAMC and other areas of computing, it can be said that OMAMC technique is also and could be useful in many areas despite the original design purposed that is for the used of keywords matching process in chatbot technology.

REFERENCES

Crocker, M.W., M. Pickering and C. Clifton, Jr., 1999. Architectures and Mechanism for Language Processing. 1st Edn., Cambridge University Press, Cambridge, ISBN: 0521631211, pp: 357.

Christy, A. and P. Thambidurai, 2008. CTSS: A tool for efficient information extraction with soft matching rules for text mining. J. Comput. Sci., 4: 375-381. DOI: 10.3844/jcssp.2008.375.381

Lokman, A.S. and J.M. Zain, 2010a. Chatbot enhanced algorithms: A case study on implementation in Bahasa Malaysia human language. Network. Digit. Technol., 87: 31-44. DOI: 10.1007/978-3-642-14292-5_5

Lokman, A.S. and J.M. Zain, 2010b. Extension and prerequisite: An algorithm to enable relations between responses in chatbot technology. J. Comput. Sci., 6: 1212-1218. http://www.scipub.org/fulltext/jcs/jcs6101212-1218.pdf

Ohno-Machado, L. and G.M. Webber, 2005. Data representation and algorithms for biomedical informatics applications. Ph.D. Thesis, Harvard University. http://portal.acm.org/citation.cfm?id=1104329

Raju, S.V. and A.V. Babu, 2007. Parallel algorithms for string matching problem on single and two dimensional reconfigurable pipelined bus systems. J. Comput. Sci., 3: 754-759. DOI: 10.3844/jcssp.2007.754.759

Shah, H., 2006. ALICE: An ACE in digitaland. tripleC, 4: 284-292.

Shawar, A.B. and E. Atwell, 2007. Chatbots: Are they really useful? LDV-Forum Band, 22: 31-50.

Turing, A.M., 2008. Computing Machinery and Intelligence. In: Parsing the Turing Test, Epstein R., G. Roberts and G. Beber (Eds.). Springer, USA., ISBN: 978-1402096242, pp: 23-65.

Wallace, R.S., 2008. The Anatomy of ALICE. In: Parsing the Turing Test, Epstein R., G. Roberts and G. Beber (Eds.). Springer, USA., ISBN: 978-1402096242, pp: 181-210.

Weizenbaum, J., 1966. ELIZA-a computer program for the study of natural language communication between man and machine. Commun. ACM, 9: 36-45. DOI: 10.1145/365153.365168