I.J. Modern Education and Computer Science, **2019, 3, 16-22**
Published Online March 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2019.03.03

Improvement of Chatbots Semantics Using Wit.ai and Word Sequence Kernel: Education Chatbot as a Case Study

Alaa A. Qaffas
University of Jeddah, Jeddah, Saudi Arabia
Email: aaqaffas@uj.edu.sa

Received: 06 December 2018; Accepted: 20 January 2019; Published: 08 March 2019

***Abstract*—**Designing interactive question-response systems has become an important challenge in Artificial Intelligence which aims to build a computer program, referred to as Chatbot, able to manage an online human-computer conversation with natural language. The number of chatbots continues to increase these recent years using different languages, tools and platforms and has been used in several domains, such as marketing, education and medicine. One of the most important issues in designing chatbots is its degree of interoperability with human natural language. In this context, we propose in this work a new messenger chatbot design approach based on Wit.ai and Word Sequence kernel in order to improve semantics. Wit.ai is used to detect contexts and concepts while the Word Sequence Kernel is used as a similarity measure between textual conversations taking into account the order of appearance of words in the conversation. A testing educate chatbot has been build which aims to provide FAQBot system for university students and acts as undergraduate advisor in student information desk. The performance of the proposed chatbot was compared to conventional messenger chatbots and showed better results.

***Index Terms*—**Natural language processing, Chatbots, textual data management, textual data analysis, text similarity.

I. Introduction

Virtual assistants are helping customers with day-to-day tasks like maintaining to-do lists, setting reminders and getting daily news updates among a panoply of required tasks. Digital assistants like Alexa, Google Home and Echo are increasingly becoming popular with consumers. Seasoned brand marketers have started taking notice of the power of conversations to drive marketing activities even though it is just the early days of messenger marketing. According to Facebook, 20 million out of the 70 million brands/pages are actively responding to consumer messages [15].

This trend is not surprising as popular sources suggest that consumers love messaging and prefer interacting with brands via popular social media applications. Customers report the on-demand access to brands whenever and wherever provides a much needed and preferred convenience. It's estimated that nearly two-thirds of consumers expect to increase their messaging applications usage in the next two years. Seeking to enhance their two-way conversational capabilities, brand marketers recognize the immense built-in audience. The most glaring shortcoming of today's chatbots is that do not really understand what you are querying. They will often misinterpret typed messages and then give answers which do not much with user's queries. We propose to design and develop a "chatbot" to respond to search queries for products using Natural Language Processing techniques where semantics are improved. The proposed chatbot improves semantics by looking for entities in the user query using "wit.ai" then compares the similarity of the query of the customer to the other queries in the database and returns the response assigned to the most similar question. We propose a new method to compute the similarities of textual messages based on the word sequence kernel.

The rest of this chapter is organized as follows: Section II describes structures of chatbots and gives examples of existing works implementing chatbots. After that Section III gives steps and techniques for preparing unstructured text to numerical analysis while Section IV gives our proposed chatbot design approach which includes Wit.ai and word sequence kernel to improve semantics. The evaluation of the performance of the proposed chatbot in the field of education is then described in Section V. Finally, Section VI gives conclusion and future works.

II. Chatbots : Designing Methodologies and History

It is necessary to build a dialogue system or computer program called a Chatbot [11] in order to be able to answer phrases or keywords that have been taken from textual conversations. A chatbot is able to manage interactions between humans and computers because it can both ask and respond to questions, therefore having the ability to influence and understand user behavior.

Natural language text is input into the chatbot so that it is able to provide both questions and answers that are the most appropriate response to what has been asked. This process is repeated throughout a conversation, as illustrated in 1. In the subsequent sections, design methodologies and history of chatbots will be explored.

A. *Designing methodologies*

Designing a chatbot requires highly developed programming skills and experienced developers to achieve even a basic level of realism. There are several platforms and languages that can be used to build a chatbot. Usually, the Artificial Intelligence Markup Language (AIML), a derivative of XML, is used as a standardized language to build chatbots. This language was first used in the project A.L.I.C.E. (the Artificial Linguistic Internet Computer Entity) [16] and is able to use AIML to characterize and define various types of data objects as well as being able to describe the partial conductance of the programs it performs. AIML is effectively centered on the units of “categories” and ”topics”, with its aim being to make conversational modelling a simplified process, particularly with regard to the system-response process. In AIML, the data object class is defined as an AIML object, with the responsibility of these objects being to model conversational patterns. AIML objects have a standard structure, which is described as follows:

$$< command > parameters\ < /command >. \quad (1)$$

There exist several AIML tags, the most frequent are ”category”, ”pattern” and ”template”. The ”category” tag defines the knowledge unit of the conversation and the ”pattern” tag is able to identify input by the user, while the ”template” tag allows the chatbot to respond to particular input from the user. These tags facilitate the design of AIML chatbots, which are able to make intelligent responses to textual natural language conversations [12].

Although the complexity of designed language of chatbots, their performance is usually related to the knowledge data base and the manner the program choose the right answer. The data base must be very large and must contain answers to all predicted queries. Also, the selection method of the right answer among thousands or millions of answers must be optimized in order to choose the response that best answers the user’s query. Many research works have focused on improving recognition rates of textual conversations and the technology is now approaching reality for conversation based human computer interactions. The shown improvement of chatbot is essentially related to the improvement of NLP and learning techniques. Learning phase in chatbot means saving new questions and then using them later to give appropriate answers for similar questions [2].

Chatbots keywords-pattern matching techniques can be divided into two categories. First is rather similar to human brain incremental parsing technique (Pickering et al., 2000) 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).

B. *Chatbots history*

When considering textual messages, a specific toolkit allows the text to be organized into sentences and then split into words in order to consider and provide the most suitable response. The chatbot can be ”trained” using NLP to allow it to be familiarized with the interactions it will go through. The more examples the chatbot is given, the better it will be able to provide users with the right answers and responses to their questions.

The idea of chatbots first appeared in the 1950, but it was to be more than 50 years before the world was ready to see them implemented into real life. This was facilitated by rapid progress in artificial intelligence (AI) and natural language processing (NLP) as well as the existence of text messaging applications on a global scale. A chatbot is also known by other terms, including talkbot, interactive agent, chatterbot, Bot, IM bot and Artificial Conversational Entity. Essentially, it is a computer program that conducts a conversation in either speech or written natural language and is able to both understand the user intent and then respond in a way that is appropriate to the particular organization that it has been designed for. Due to the rise of communication technology in recent years, text communication has become a socially acceptable way to interact with others and as a result, many people now prefer to chat in this way rather than making personal telephone calls or other direct contact.

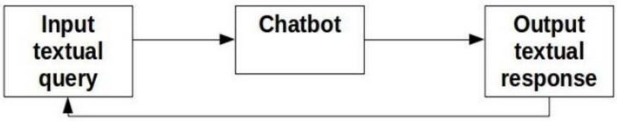


Fig.1. Example of a chatbot structure

Turing was the first to ask the question Can the machine think? [12] where thinking is a human ability of deciding the right answer for a question. The imitation game was suggested by Turing as a method to directly avoid the question and to specify a measurement evaluating the degree of similarity between the machine and a human [15]. Chatbots continue to grow in popularity with 80% of businesses expecting to be using one by 2020 [15]. There are now chatbots for multiple uses such as Customer Service (Twyla), time-sheets (Nika), Expenses (Acebot) and feedback (Wizu). We give in the following a brief history of intelligent machines which were able to make conversation with humans.

- **1966 ELIZA [17]:** was the first chatbot using pattern matching to match user prompts and scripted responses. It was able to pass the Turing test which evaluates the intelligent behavior of the machine.
- **1972 Parry [12]:** Its goal was to simulate the disease and resembles the thinking of an individual. It was more advanced when compared with ELIZA.
- **1988 Jabberwacky [4]:** It was designed to simulate a natural human conversation using a voice operated system.
- **1996 A.L.I.C.E [16]:** The name of the project was Artificial Linguistic Internet Computer Entity (A.L.I.C.E) and was a bot which uses heuristic pattern matching rules to human inputs.
- **2006 IBMs WATSON [1]:** Watson is a question-answering entity able to give answers to questions posed in natural language. WATSON uses NLP and machine learning to generate insights from small to vast amounts of data.
- **2010 Siri[2]:** is a virtual assistant of Apples iOS interface which uses natural language to answers users queries and to perform web service requests.
- **2012 Google Now [6]:** is a mobile application for android and iOS which offers predictive cards with information and daily updates developed by Google to automatically answer users' questions. Also, it can be used to make recommendations and to perform actions by submitting queries to a set of web services.
- **2015 Cortana [13]:** is a Windows Speech Platform including information to help hardware and software developers. It acts as a personal assistant recognizing NLP commands and using "Bing" to answer user questions.
- **2016 Bots for Messenger [5]:** A messenger bot is a program which uses artificial intelligence (AI) to communicate with customers. Facebook launched this platform to allow developers create bots that can interact with Facebooks users using Facebooks chat interface Messenger. Messenger bots understand users' queries and can then formulate a response in a very human way.

III. Text Analysis, Nlp And Chatbots

A. Preparing texts for numerical analysis: VSM and n-Grams

Text is normally considered to be unstructured data, for which it is impossible to conduct numerical analysis. Therefore, a Vector Space Model (VSM) representation is used to prepare the text to be numerically analyzed or calculated and in this VSM model, the text j is represented by a vector of terms:

$$\underset{t_j}{\rightarrow}= (w_{1j}, w_{2j}, \dots, w_{|T|j}) \tag{2}$$

where T is the whole set of words $T = (t_1, \dots, t_{|T|})$, T is the size of the vocabulary and w_{kj} represents the weight of the term w_k in the text t_j. When the vectors are close to each other, the text is considered to be similar in content and the VSM representation assumes that the relative position of words is of minimal importance. However, this may lead to a loss in terms of the correlation of adjacent words and information regarding the positioning of words is lost.

Meanwhile, the "*n*-Grams" representation of text [18], is an independent language text representation technique that attempts to solve the problem of the loss of information about the positions of words by considering each textual input as a sequence of *n* characters, syllables or words. Using an extraction of all the possible ordered subsequences of consecutive *n* characters, a set of "*n*-Grams" is built. The similarities between the text that has been input are then measured, according to the number of contiguous and non- contiguous sub-sequences that are shared between them. However, this representation also presents issues as a high number of dimensions is needed to represent each input text.

Several embedding techniques have been considered in order to tackle the issues of high dimensionality. One example of this is the application of Random Projections (RP) [7] to text representation as it reduces the dimensionality of the term space. It does this by projecting it to a lower-dimensional subspace, which is formed by a set of random vectors. Latent Semantic Analysis (LSA) [8] has also been used in text representation [14] and uses Singular Value Decomposition (SVD) to project the original, high dimensional vector space to a lowdimensional concept vector space. In this new, lowdimensional concept vector space, the dimensions are statistically uncorrelated, or orthogonal.

B. NLP and chatbots

The integration of NLP into chatbots allows a more human element to be added. When a chatbot is built for public use, it is common for users to ask it some difficult questions, as human nature tends to dictate that they will try to beat the bot or stump it with difficult questions. The developers can try to solve this issue or at least mitigate it by adding some default answers, although it is almost impossible to provide a suitable answer to everything or predict what may be asked because there are so many variables [12].

The chat client can include natural language processing so that users can be helped to perform data queries or access content. The chatbot should contain a set of sample phrases that have been mapped with a specific intention in mind in order to determine intents and entities. An administrator who has been specifically trained to manage this process will provide sample phrases for each of the intents. In addition, to identify existing entities in the user queries, natural language processing can be further exploited and machine- learning techniques can facilitate the establishment of similar phrases for each intent. The administrator can also review any new phrases that users provide and identify the correct entities and intents so that the platform's performance can be further developed and enhanced [12].

Given that not all interactions can be predicted, several strategies need to be implemented in order to build chatbots able to give an answer and, thus, simulate intelligence. A first possible solution is the association of a prior personality to a chatbot allowing to justify some answers to be considered inappropriate. A second solution is to direct the conversation, by asking questions inciting the user participation. This fact will keep the conversation with little contribution from the program.

IV. Proposed Approach

We propose to improve the performance of chatbots using semantics and the word sequence kernel as a similarity measure. The proposed approach can be described by the following stages as illustrated in Fig.2 : the first stage is devoted to pre-processing and preparing data, the second to improve semantics and the final one is devoted to the similarity evaluation and the selection of the best response. The main algorithm of the proposed method is described in algorithm 1.

```
Algorithm 1 Chatbot based wsk and wit.ai
Require: $m_q$: input textual message,
    $D<m_i, r_i>$: DataBase Question-Responses
Ensure: $r_i$
  1: $m^{'}q$ = pre-processing ($m_q$)
  2: $m^{''}q$ = Wit.ai ($m^{'}q$ )
  3: Evaluate similarity of $m^{''}q$ with each $m_q \in D$
  4: return $r_i$ having the max similarity based on WSK
     similarity measure
```

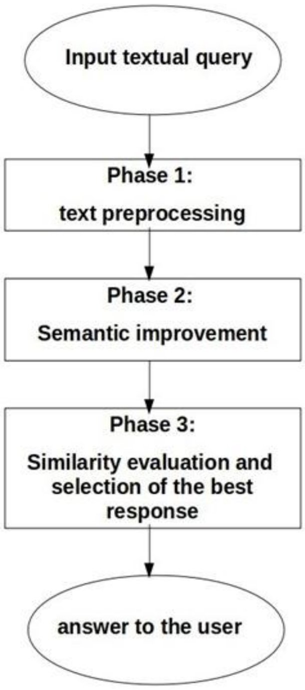


Fig.2. The different phases of the proposed approach

A. Phase 1: text preprocessing and filtering

Text preprocessing is usually considered as an important process in text mining applications. This process is used to transform input textual queries to meaningful information for the next step of processing. We present in the following the different text preprocessing techniques which will be used to prepare users textual queries for the next proposed phase:

- **Tokenization:** which consists in dividing text into linguistic units, words, phrases, terms and symbols named tokens. In this process, we use the separator space to break text into *k* words.
- **Stop words elimination :** Given the *k* obtained words after tokenization, we remove common terms which appears frequently in the text and do not help to find the right meanings. This process decreases the size of data and makes the system more efficient. We propose to combine three stop words lists, the first is ”SMART” [9][1] built by Gerard Salton and Chris Buckley for the experimental SMART information retrieval system at Cornell University which contains 571 English words, the second list is ”SNOWBALL”[2] containing 175 stop words for multiple languages. Most of these have been ported from the Quanteda stop word list. The last list is ”ONIX”[3] which contains 429 words. As a result, our stop words list contains 1175 multi-languages stop words.

B. Phase 2: Semantic improvement

To improve semantics, we propose to use a new NLP platform, referred to as ”Wit.ai” adopted by Facebook since January 2015 and is behind the NLP capabilities of Facebook Messenger platform. This platform has rolled out major updates. One of the best capabilities of Wit.ai is the sophisticated tool that can be used to train the platform in new conversation models as well as monitoring the interactions between users and the platform. Wit.ai allows using entities, intents, contexts, and actions, and it incorporates natural language processing (NLP) techniques. It supports about 50 free languages and offers for developers an easy interface to build applications and to connect devices in or- der to change textual or voice messages.

Wit.ai contains two major components: Understanding and Inbox. The Understanding is the page where you can teach your bot sentences and words using intents and entities while the Inbox component contain the user’s messages sent to the chat bot. The bot works with confidence. If the value of confidence is less than a threshold, it sends the messages to the inbox. The developer should look at the phrases and validate them or gives answers.

[1] http://snowball.tartarus.org/dist/
[2] http://snowball.tartarus.org/dist/
[3] https://github.com/igorbrigadir/stopwords/blob/master/en/onix.txt

Our goal of using wit.ai is to find entity from users question. For example, to ask for the temperature a user can formulate his question with different syntax terms:

- What's the current temperature?
- Is it me or it is hot here?
- could you give me the current temperature please?
- temperature please ...

Using wit.ai we can detect that all these questions are about the entity "Temperature". We note also that one question can contain several entities.

C. Phase 3: Similarity calculation and selection of the best answer

In order to improve semantics when calculating textual similarities of users queries and existing questions in the data base, we propose to use the word sequence kernel which keeps order of words as they appear in the message. Many kernels known as String Kernel are proposed in the literature to solve the problem of high dimensional features in n-Grams representation of Text. The n-Grams are not computed for each input, but only dot products between n- Grams of each pair of inputs are computed. An example of these kernels is WSK (Word Sequence Kernel) [3]. The advantage of using words as atomic unit is to keep information regarding words positions in order to maintain linguistic meaning of terms. For example, the terms son-in-law have a special meaning that can be lost if they are broken. The WSK has also the advantage of reducing the number of features per text because it uses sequences of words rather than sequences of characters. The time complexity of computing WSK similarity between two messages m_1 and m_2 is evaluated to $O(n\ m_1\ m_2\ )$ [3] where n is the length of the used subsequence and m_i is the number of words in message m_i.

Let Σ the alphabet consists of the set of words that exist in the text. Let $S = m_1\ m_2\ m_3 ... m_{|S|}$ a sequence of words where S is the length of S. Let $u = s[i]$ a subsequence in S with $s[i] = t_i1\ ...tj\ ...t_in$ where t_i1 and tj in this subsequence are not necessarily contiguous in S and n is the length of the subsequence u. The feature mapping φ for the sentences S in the feature space is provided by defining φ_u for each $u \in \Sigma^n$ as:

$$\emptyset_u(\mathrm{S}) = \sum_{i:u=s[i]} \lambda^{l(i)} \tag{3}$$

where the decay factor λ is used to penalize non-contiguous sub-sequences and $l(i)$ is the length of sub-sequence $s[i]$ in S and defined by :

$$l(i) = Index\ (t_{in}) - Index\ (t_{i1}) + 1 \tag{4}$$

These features are used to calculate the number of times that sub-sequence u in the sentences S occurs and these are then weighted according to their length. Therefore, when there are two strings S_1 and S_2, the feature vectors' inner product is gathered by the sum being computed across all the common sub-sequences.

$$K_n(S_1, S_2) = \sum_{u\in\sum^n} \emptyset_u(S_1)\, \emptyset_u\,(S_2)$$
$$= \sum_{u\in\sum^n} \sum_{i:u=s1\,[i]} \sum_{j:u=s2\,[j]} \lambda^{l(i)+l(j)} \tag{5}$$

V. Experiments and Evaluations Results: Education Chabot

We choose education chatbots as a challenging domain to evaluate the performance of the proposed chatbot's designing approach. Recently, messenger chatbots has been designed for revolutionary universities such as Georgia State University, the University of Memphis, West Texas and Arizona State University. Education chatbot aims to improve communication, minimize ambiguity from interactions and increase productivity by facilitating information access and retrieval. Students can ask questions about general topics such as financial aid or exam issues more specific to the college they are inquiring about. We present in the following two challenging use cases demonstrating the motivation of education chatbots to create a better cam- pus experience to students:

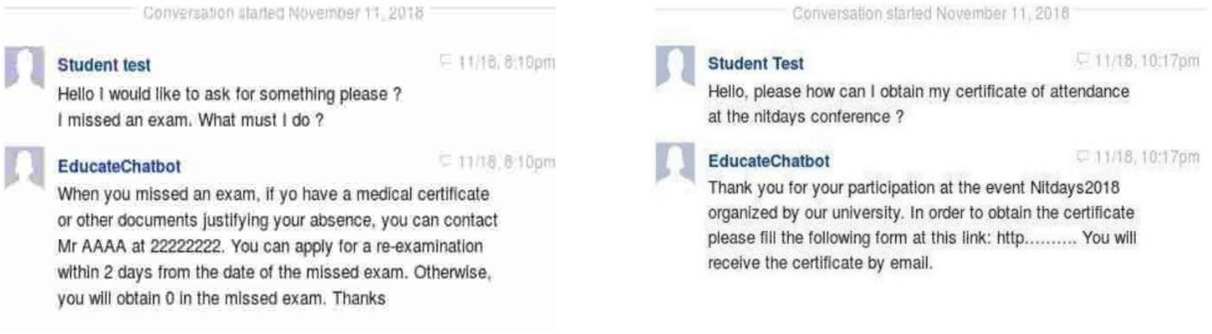
- Given a new student having no idea about scholarships that can potentially reduce cost of his course. He can visit the university website and looks for details when he finds time. This may take several minutes since the student is not familiar with the university website and may not find response to his query. Education chatbot can accelerate and facilitate this process by simply typing the question "*Where can I find information on your scholarships?*" and the chatbot gives an instantly answer *Here's a link to all our current scholarships.* In addition, the chatbot can go further by making a conversation with the student in order to find only scholarships that can be applied to his situation. The chatbot can ask for example "*Which department are you in? (Please select one.)* " or which "*Which course are you studying? (Please select one.)*"
- Given a student that has missed an exam and would like to know what he must to do ? is there any possibility that he pass the exam later ? This is an extremely use case where the student may not find the response in the website and time looking for the response is also important since he can have the right to do the exam only after a short period. The education chatbot may facilitate the action for this student by simply asking "*What do I do if I miss an exam?*" and the chatbot gives him options or responses like *You can apply for a re-examination within* 2 *days from the date of the missed exam*".

We build an ensemble of 200 questions-responses related to the field of education extracted from a university Facebook page. Question and responses are saved in the database after removing Facebook emoticons and punctuation. Examples of these questions' responses are as following:

1. < ”what list documents required inscription”, ”The list of documents required for inscription are: Recent Photo and all copies of diploma ” >
2. <”How apply certificate Attendance”, ”To request a certificate of attendance you must go to the university website in this link:’... ’ and click on the certificate section and then fill in your details”>
3. ...
4. *etc.*

An example of conversations between our educate messenger chatbot and a test user is shown in Fig. 3. We show the ability of the proposed chatbot to give answers for students which are querying about missed exams and steps to obtain a certificate of attendance for an event organized in the university. This may help students to rapidly find responses to their questions without losing time to search this information in the website or contacting professors and university administrators.

In order to objectively evaluate the proposed chatbot to give correct answers for students, we prepared 100 users queries related to 50 questions in the simulated dataset. For each query, we evaluate if the response of the chatboat corresponds to the expected answer. We calculate the percent- age of correctly identified responses by dividing the number of correctly identified responses by the total number of queries. We compare the obtained results of the proposed chatboat with and without using Wit.ai to those obtained using Jaccard and Cosine textual similarities. Table 1 reports the percentage of correctly identified responses for each evaluated method. We show that the proposed approach gives better results than those obtained with Jaccard and Cosine. We obtained 89% of correctly answers from a set of 100 queries on 50 questions for the proposed chatbot while only 60% using Jaccard textual similarity. Another interesting result reported in Table 1 is the important improvement of quality when using Wit.ai which increases quality of correct answers about 24%.



Conversation started November 11, 2018

Student test 11/18, 8:10pm

Hello I would like to ask for something please ?
I missed an exam. What must I do ?

EducateChatbot 11/18, 8:10pm

When you missed an exam, if yo have a medical certificate or other documents justifying your absence, you can contact Mr AAAA at 22222222. You can apply for a re-examination within 2 days from the date of the missed exam. Otherwise, you will obtain 0 in the missed exam. Thanks

Conversation started November 11, 2018

Student Test 11/18, 10:17pm

Hello, please how can I obtain my certificate of attendance at the nitdays conference ?

EducateChatbot 11/18, 10:17pm

Thank you for your participation at the event Nitdays2018 organized by our university. In order to obtain the certificate please fill the following form at this link: http.......... You will receive the certificate by email.

Fig.3. Examples of messages between the educate chatbot and a user

Table 1. Comparison of the proposed chatbot with and without using Wit.ai and WSK to standard textual similarity method

Method	percentage of CIR	number of queries	number of used questions
Jaccard textual similarity	60 %	100	50
Cosine textual similarity	62 %	100	50
Proposed chatbot without Wit.ai	65 %	100	50
Proposed chatbot	89 %	100	50

Table 2. Comparison of the semantic performance of the proposed chatbot with and without using Wit.ai and WSK to Jacarrd and Cosine textual similarity methods

Method	Percentage of CIR	number of queries	number of used questions
Jaccard textual similarity	20%	50	10
Cosine textual similarity	25 %	50	10
Proposed chatbot without Wit.ai	32 %	50	10
Proposed chatbot	70 %	50	10

In order to better demonstrate the high semantic improvement of Wit.ai compared to classic messenger chat- bots, we prepared 50 questions related to only 10 responses. We formulated these questions in such a way that the same question is asked using different syntactic terms. Table 2 reports obtained scores. We show the high improvement of correctly identified responses equally to 50% compared to Jaccard and Cosine.

From the scores given by our evaluations the following inferences could be taken:

- The proposed building approach of the chatbot outperforms those designed based on standard textual matching such as Jaccard and Cosine.
- The proposed chatbot has a good running time as Jac- card and Cosine and gives instantly responses.
- Wit.ai largely improves quality of responses when question are syntactically different to those in the Knowledge database, which is the case for users in real life application of chatbots.

VI. Conclusion and Future Works

We proposed in this work a new approach of building Facebook messenger chatbots integrating Wit.ai and word sequence kernel to improve semantics. Wit.ai is used to efficiently detect contexts and concepts while the Word Sequence Kernel is used as a similarity measure between textual conversations taking into account the order of appearance of words in the conversation. A testing educate chat- bot has been build which aims to provide FAQBot system for university students and acts as undergraduate advisor in student information desk. The performance of the proposed chatbot were compared to conventional messenger chatbots and showed better results.

As future works, it would be interesting to evaluate the proposed approach in other interesting domains in order to confirm its showed performance in the education domain. We also plan to improve the computational complexity when looking for the best answer. For example, we can generalize the context of conversations by clustering similar topics and grouping similar answers in order to reduce the search space when looking for the best response.

References

[1] IBM 2018. Overview of Watson assistant. https://console.bluemix. net/docs/services/conversation/index.html, 2018.

[2] Jerome R Bellegarda. Spoken language understanding for natural interaction: The siri experience.,. In Natural Interaction with Robots, Knowbots and Smartphones, pages 3–14, 2005.

[3] N. Cancedda, E. Gaussier, C. Goutte, and J.M. Renders. Word-sequence kernels. Journal of Machine Learning Research, 3:1059–1082, 2003.

[4] R. Carpenter and J. Freeman. Computing machinery and the individual: the personal turing test, http://www.jabberwacky.com, 2005.

[5] J. Constine. Facebook launches messenger plat- form with chatbots.,. [online] Available at: https://Itechcrunch.com/2016/04/12/agents on messenger/, 2016.

[6] Brooke Crothers. Google now reporting self-driving car accidents: Her, it's not the car's fault. forbes.com, Jun 8, 2015.

[7] Kumar R. Sarls T. Dasgupta, A. Fa sparse johnson: Linden strauss transform. In in Proceedings of the forty-second acm symposium on theory of computing, pages 341–350, 2010.

[8] Dumais S. T. Furnas G. W. Landauer T. K. Harsh- man R. Deerwester, S. Indexing by latent semantic analysis, Journal of the American society for information science, 41(6), 1990.

[9] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. J. Mach. Learn. Res., 5:361– 397, 2004.

[10] H. Lodhi, N. Cristianini, J. Shawe-Taylor, and C. Watkins. Text classification using string kernel. The Journal of Machine Learning Research, 2:419–444, 2001.

[11] Michael L. Mauldin. Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. In AAAI, 1994.

[12] Ayse Pinar Saygin, Ilyas Cicekli, and Varol Akman. Turing test: 50 years later. Minds and Machines, 10(4):463–518, 2000.

[13] Bhupendra Singh and Upasna Singh. A forensic in- sight into windows 10 cortana search. Computers and Security, 66:142–154, 2017.

[14] Park S. C. Song, W. A novel document clustering model based on latent semantic analysis. In in Proceedings of the Semantics, knowledge and grid, third international conference, pages 539–542, 2007.

[15] D. R. Vukovic and I. M. Dujlovic. Facebook messenger bots and their application for business. In 2016 24th Telecommunications Forum (TELFOR), pages 1–4, 2016.

[16] Richard S. Wallace. The Anatomy of A.L.I.C.E., pages 181–210. Springer Netherlands, 2009.

[17] Joseph Weizenbaum. Eliza &mdash; a computer program for the study of natural language communication between man and machine. Common. ACM, 26(1):23–28, January 1983.

[18] E.J. Yannakoudakis, I. Tsomokos, and P.J. Hutton. N-grams and their implication to natural language understanding. Pattern Recognition, 23(5):509 – 528, 1990.

Authors' Profiles

A laa A. Qaffas received the *B.S.* degree in Computer Science from King Abdul-Aziz University, Jeddah, Saudi Arabia, in 2007 and the *M.S.* degree in Computer Science from Hertfordshire University, Hatfield, United Kingdom, in 2010.

Dr. Qaffas completed the *Ph.D.* degree in Computer Science from the University of Warwick, Coventry, United Kingdom, in 2016. Currently, he is an assistant professor in the Department of Management Information Systems (MIS), University of Jeddah. His research interests include e-advertising, adaptive advertising, adaptation model, personalization, user modeling, authoring and delivering personalized content and social networks analysis.

How to cite this paper: Alaa A. Qaffas, " Improvement of Chatbots Semantics Using Wit.ai and Word Sequence Kernel: Education Chatbot as a Case Study", International Journal of Modern Education and Computer Science(IJMECS), Vol.11, No.3, pp. 16-22, 2019.DOI: 10.5815/ijmecs.2019.03.03