Journal of Education and Science (ISSN 1812-125X), Vol: 30, No: 1, 2021 (117-125)

# Adopting Text Similarity Methods and Cloud Computing to Build a College Chatbot Model

**Zaid A. Mundher [*1], Wissam K. Khater [2], Laith M. Ganeem[3]**

[1, 2, 3] Department of Computer Science, Collage of Computer Science and Mathematics, University of Mosul, Mosul, Iraq

E-mail: [1*]zaidabdulelah@uomosul.edu.iq, [2] khalfwissam8@gmail.com, [3] laithmohammad16@gmail.com

(Received June 02, 2020; Accepted September 08, 2020; Available online March 01, 2021)

DOI: 10.33899/edusj.2020.127244.1079, © 2020, College of Education for Pure Science, University of Mosul.


## Abstract

A chatbot is a computer program which is designed to interact with users and answer questions. Nowadays, chatbots are one of the most common systems that are used in many fields and by different companies to achieve different tasks. Cloud computing is gaining increasing interest. A myriad of fields and applications have been developed based on cloud computing.

In this paper, a college chatbot was developed and implemented to assist students to interact with their college and ask questions related to faculty, activities, exams, admission, amongst other tasks. Text similarity algorithms were adopted to achieve the proposed system. More specifically, cosine similarity and jaccard similarity algorithms were used to find the closest question in the dataset. Firebase real-time database, which is one of the Google cloud services, was used as a connector channel between users and the chatbot server.

Experiments were conducted to evaluate the performance of cosine similarity and jaccard similarity methods, and to compare the results of both. In addition, real-time database was also evaluated as a chatbot connecter channel.

**Keywords:** Chatbot, Text Similarity, NLP, Cloud Computing, Firebase, Mobile Programming.

# اعتماد طرق تشابه النصوص وتقنية الحوسبة السحابية لبناء نموذج دردشة خاص بالكلية

**زيد عبد الاله منذر و [2] وسام خلف خضر و[3] ليث محمد غانم [*1]**

قسم علوم الحاسوب, كلية علوم الحاسوب والرياضيات, جامعة الموصل, الموصل, العراق

## الخلاصة

برنامج الدردشة الآلي هو برنامج حاسبة يصمم للتفاعل مع المستخدمين والإجابة عن أسئلتهم. في الوقت الحاضر، تعد برامج الدردشة الآلية واحدة من أكثر الأنظمة شيوعًا التي يتم استخدامها في العديد من المجالات من قبل العديد من الشركات لتحقيق مهام مختلفة. على صعيدٍ أخر، أنتشر مفهوم الحوسبة السحابية في الأونة الأخيرة حيث تم تطوير العديد من المجالات والتطبيقات بالاعتماد على مفهوم الحوسبة السحابية.

في هذا البحث، تم تطوير وتنفيذ برنامج دردشة آلي لمساعدة الطلاب على التفاعل مع كليتهم وطرح الأسئلة المتعلقة بالكلية كالنشاطات والامتحانات والقبولات وما إلى ذلك. لقد تم اعتماد خوارزميات تشابه النصوص لتحقيق النظام المقترح في هذا العمل. تحديدًا، تم

استخدام خوارزميات الـ cosine similarity و الـ jaccard similarity . كما تم استخدام قاعدة بيانات Firebase في الوقت الحقيقي (وهي إحدى خدمات Google السحابية) كقناة رابط بين المستخدمين وخادم برنامج الدردشة الالي(Server).
أجريت التجارب لتقييم أداء الخوارزميات المستخدمة ومقارنة نتائج كل منهما. بالإضافة إلى ذلك ، تم تقييم قاعدة البيانات في الوقت الفعلي أيضًا كقناة اتصال.

**الكلمات المفتاحية:** الدردشة الالية، خوارزميات تشابه النصوص، معالجة اللغات الحية، الحوسبة السحابية، برمجة الموبايل.

## 1. INTRODUCTION

Chatbots can be defined as computer programs designed to automatically interact with users to answer their questions [**1**]. Using chatbots is common in companies, banks, education, amongst other tasks. Different techniques have been developed and adopted to build different kinds of chatbot systems. Basically, natural language processing (NLP) techniques and machine learning algorithms are used to develop chatbot systems.**[2][3]**
Developing a chatbot is an arduous task, therefore different platforms, such as IBM Watson and Microsoft Bot Framework, are available to help developers build chatbots. The main drawback of using these platforms is that not all languages are supported. In addition, these platforms do not produce stand-alone chatbots. Accounts on social media such as Facebook, Slack, and Wechat are needed to chat with the bot.
In general, chatbots can be divided into different types based on the approach used to develop them[**4**]. The main two types are retrieval-based approach and generative-based approach**[5][6].** In the retrieval-based approach, a dataset of predefined responses is used to answer questions. The main function of this kind of chatbots is to find the closest question in the dataset and retrieve the corresponding answer. To find the best answer, the score of similarity between the user question and the predefined questions in the dataset needs to be calculated[**7**].
In contrast, in the generative approach, machine learning algorithms and deep learning models are used to generate answers from scratch. This approach needs a large dataset with millions of examples to train the model. [**1**]
From another point of view, chatbots may be closed domain or open domain. Closed domain means that the chatbot concentrates on one specific field or topic. Open domain, however, means that there is no specific topic and the chatbot may answer any question.
In this work, the retrieval-based approach was used to build a closed domain college chatbot to answer students' queries. Using the introduced system, students can ask questions regarding the College of Computer Science of the University of Mosul. The College chatbot would help in reducing the response time and the effort which is needed to answer students' queries.

## 2. RELATED WORKS

Much research has been conducted on developing chatbots. Different methods and techniques have been introduced to build chatbots. In [**8**], a web application was developed to interact with students. Bigram was used to find the similarity between sentences. In [**9**], WordNet, which is a lexical database was used to find similarities. In [**10**] AIML (Artificial Intelligent Markup Language) was used beside the open source project "program –o" to develop the chatbot. In [**11**], cloud-based cognitive services were used. IBM OpenWhish, which is a serverless platform, was used to implement the chatbot. Microsoft Bot Builder was used in [**4**] to build an English chatbot.
Clearly, using public cloud services has limitations. Cloud services are not free, and chatbot platforms do not support all languages. For example, many chatbot platforms do not support Arabic.

In this work, a chatbot, that can support any language, was developed from scratch. In addition, this work introduced a stand-alone mobile-based chatbot, since it is not developed based on any chatbot platforms, or any social media platform.

## 3. TEXT SIMILARITY

Measuring similarity between texts is one of the most common techniques that are used to achieve various tasks which are related to data mining and information retrieval. In general, text similarity algorithms aim to find how two sentences or documents are similar relying on some mathematical concepts and equations[**12**].Text similarity methods have been used to build different kinds of systems such as translation systems, plagiarism detection systems, text clustering, and short-answer grading. Some examples of text similarity algorithms are cosine similarity, Levenshtein distance, Jaccard distance, and Euclidian distance.

### 3.1. Cosine Similarity

Cosine similarity method calculates similarity (angle) between two non-zero vectors [**12**][**13**]. Cosine similarity is calculated as:

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

Before applying the cosine similarity to measure the similarity between two sentences, text vectorization techniques should be used. Text visualization methods aim to convert text to some numerical representations [**14**]. Bag of words, TF-IDF, and word2dev are examples of text vectorization techniques. In this work, TF-IDF method was used to convert queries to numerical vectors.

### 3.2. Jaccard Distance

Jaccard Index is defined as the size of shared terms divided by the size of all unique terms of two sets.[**15**][**13**]

Jaccard similarity index and Jaccard distance are calculated using the following :

$$JaccardIndex = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

$$JaccardDistance = 1 - JaccardIndex$$

To use jaccard distance, strings/statements must convert to sets.

## 4. The Proposed System

The proposed chatbot system consists of three different parts that are showen in Figure 1, and explained below.

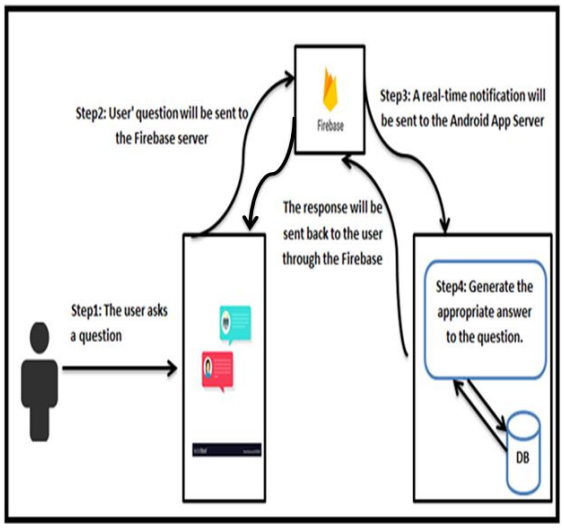


**Figure 1: Overall Proposed System Architecture**

1. Android-based client app which acts as a chatbot channel. Students can use the client app (on their smartphone) to interact with the chatbot and ask questions. The algorithm of the client app is explained in the following:

*var firebaseRef = FirebaseGetReference()*
*var phoneNumber = GetClientPhoneNumber()*
*var msgInput ← ReadUserInput()*
*firebaseRef.child(phoneNum).child("msg") ← msgInput*
*firebaseRef.child(phoneNum).child("response") ←*
*AddListener*
*var response = ReadResponse*
*Display response*

2. Android-based server app which acts as a chatbot. The server app analyses the students' queries first, and then provides an appropriate answer. Cosine similarity and jaccard distance methods are used to find the similarity among questions. The answer to the question that has the highest score of similarity will be retrieved. Figure 2 and Figure 3 illustrate the steps of this part.

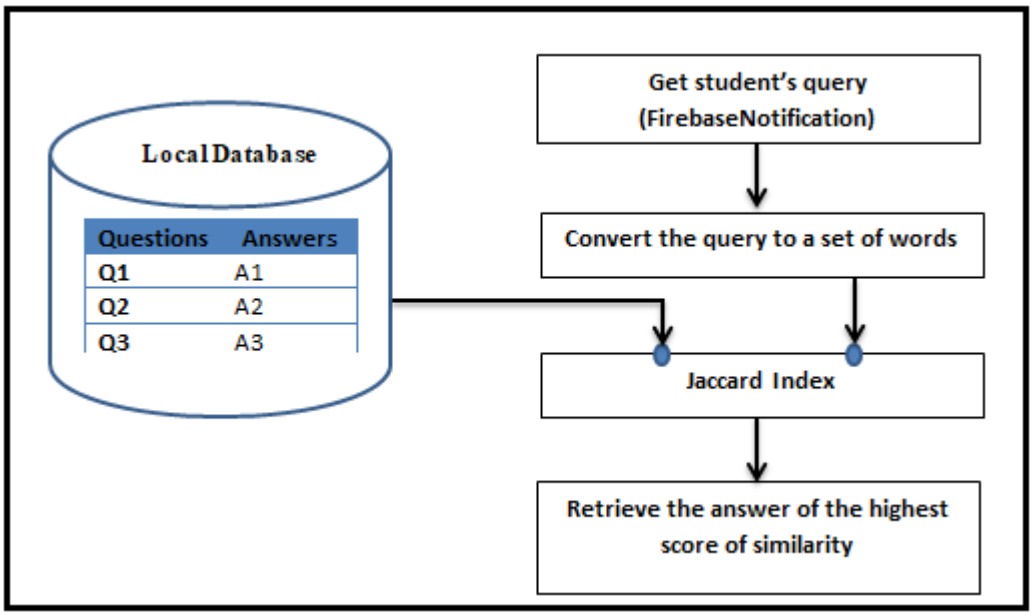


**Figure 2: Server's steps using jaccard distance method**

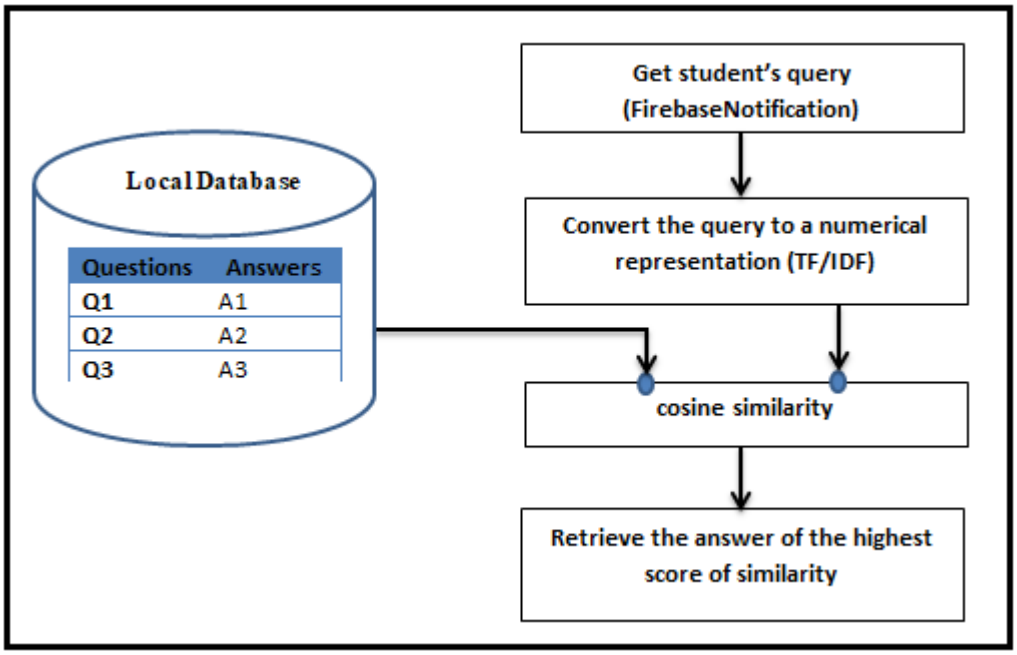


**Figure 3: Server's steps using cosine similarity method**

.

3. Real-time cloud database which acts as a chatbot connector. Each user (client) is represented in the Firebase as a node based on the phone numbers, and his/her questions are stored as a sub-node. Figure 4 shows the structure of the Firebase database.

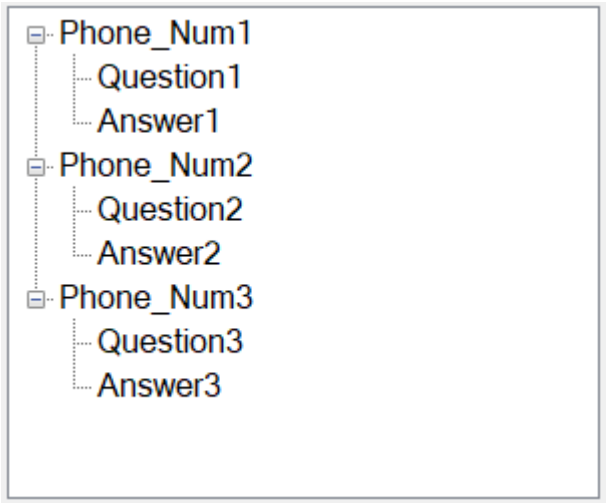


**Figure 4: Firebase database structure**

In addition to the above parts, the proposed system has a local database to store questions and answers. The database may be updated and new questions can be added by the administrator of the database.

The main steps of the proposed system can be summarized as followed:

- Obtain student's question.
- Send the question to the firebase
- Firebase sends notification to the chatbot server
- The server listens to the firebase notification to read the student's question.
- Calculate the TF-IDF for the user question.
- Calculate the cosine similarity/jaccard distance between the user question and questions in the database.
- Retrieve the answer to the closest question.

## 5. Implementation

The proposed system was implemented using Kotlin programming language with Android Studio. The proposed chatbot system has two applications that were developed to act as a chatbot client and a chatbot server. Students interact with the client-side application to ask questions and obtain answers. Figure 5 provides an example of the client screen.


**Figure 5: Sample of Conversation Flow**

The server application has more features that allow the admin to administrator more questions and modify answers. Figure 6 shows the main screen of the server.


**Figure 6: The Main Screen of the Server**

Conversations of the users (students) are stored on the cloud (Firebase) with a special node for each student. Students are identified by their phone number.

## 6. Experiments and Results

Experiments were conducted to evaluate the performance of the proposed system. In general, cosine similarity and jaccard distance give acceptable and reasonable results.
Table 1 shows the results of using the cosine and jaccard similarity methods with some sample data. Noting that in cosine similarity and jaccard similarity methods, 1 means that the sentences are similar, while 0 means that the sentences are completely dissimilar. The threshold value was set to 0.5.

**Table 1: Samples of questions matching using different measures**

| Stored. in the dataset | Question by a user | Cosine similarity | Jaccard similarity |
|---|---|---|---|
| How can I use your product? | How may I use your make? | 0.55 | 0.73 |
| What is your favorite programming language? | Do you have a favorite programming language? | 0.63 | 0.85 |
| Is it true that you are a computer program? | Is it right that you are a software program? | 0.41 | 0.83 |
| | | | |
| موعد مناقشة المشاريع | متى موعد مناقشة المشاريع | 0.83 | 0.87 |
| موعد التقديم للدراسات العليا | متى يبدأ التقديم للدراسات العليا | 0.5 | 0.73 |

Results have shown that the jaccard similarity method is more efficient than the cosine similarity method especially when the sentiment analysis is not considered. In addition, the results of the performance evaluation indicate the effective use of real-time database notification as a chatbot channel.Phone numbers, questions, and all other information are successfully stored on the cloud database. Figure 7 shows a sample of the users conversations that are stored on the cloud.

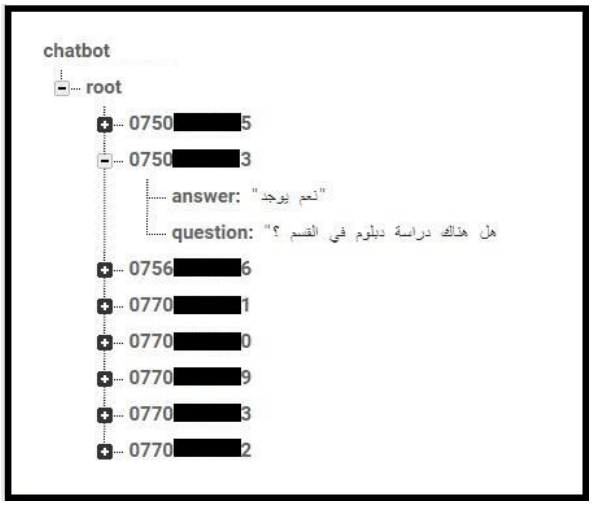


**Figure 7: Cloud Database**

## 7. Conclusion and Future Work

Chatbots have been one of the most common systems which are used to achieve different tasks. Different techniques could be used to develop different types of chatbots. In this work, text similarity methods are adopted to develop a college chatbot. Firebase real-time notification was also used as a chatbot channel between users and the chatbot. Experiments have shown acceptable and reasonable results. In addition, results have also shown the efficiency of using real-time notification as a way to connect clients and the server.

Features can be added to the proposed system. For example, other text similarity algorithms can be used to find the highest score of similarity. In addition, using other word vectorization methods may be used to find how these methods may affect the score of similarity.

## 8. References

1. Lisna, Z., Computer Engineering and Applications, 5(11), 2016.

2. Singh A. K., Shashi, M., International Journal of Advanced Computer Science and Applications(IJACSA), 10(7), 2019.

3. Yan M., Castro P., Cheng P., Ishakian V., In Proceedings of the 1st International

Workshop on Mashups of Things and APIs. ACM, 2016.

4. Jwala, K., Sirisha, G.N.V.G, Padma Raju, G.V., IJRTE, 8(1S3), 2019.

5. Sandhini S., Binu R., Rajeev R.R, Reshma M.M, IJCSE, 6(6), 2018.

6. Gomaa, W. H.; Fahmy, A.A. International Journal of Computer Applications, 68(13), 2013.

7. Sheikh, S., Singhal, S., Tiwari, V., Journal of Emerging Technologies and Innovative Research (JETIR), 6(1), 2019.

8. Rohan, K., Haldar R., International Journal of Advanced Computer Science and Applications, 7(11), 2016.

9. Bala, K., Kumar, M., Hulawale, S., & Pandita, S., International Research Journal of Engineering and Technology IRJET,2017, 4(11)

10. D'silva G.M., Thakare S., More S., Kuriakose J., International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2017.

11. Gali, N., Mariescu-Istodor, R., Fränti, P., 23rd International Conference on Pattern Recognition (ICPR) Cancún Center, 2016

12. Harsh P., IJRASET, 6(3), 2018

13. Pawar S., Rane O., Wankhade O., Mehta P., IJIRSET, 7(4), 2018.

14. Vichare, A., Gyani, A., Shrikhande, Y., & Rathod, N., IJARCET, 4(10), 2015.

15. Geethanjali, S., Birunda Antoinette Mary J., IJIRSET, 6(11), 2017.