Chatbot as an Alternative Means to Access Online Information Systems

A F Sugondo, R Bahana*

Computer Science Department, Faculty of Computing and Media, Bina Nusantara University, Jakarta, Indonesia 11480

*Corresponding author: rbahana@binus.edu

Abstract. Chatbots are having a spotlight in the current market. The reasoning behind this would be mobile applications are becoming a saturated market and messenger applications are surpassing social networking applications regarding to the number of active users. Binus Maya is becoming less attractive since it is not engineered for mobile devices, which creates inconvenience for students to access campus information. The research creates a chatbot as an alternative solution to Binus International’s existing information systems. The application created is accessible from both desktop and mobile, created with Node.js with the natural language processor being used is called API.AI. Taking accounts from the user acceptance testing, 80% of testers agreed that the created chatbot can be a good and faster alternative information retrieval compared to Binus Maya.

Keywords: Chatbot, Information System, Web Application, Mobile Application

1. Introduction

Whether we notice it or not, artificial intelligence has been creeping in to empower our daily lives. E-commerce's featured item recommendation, song or video recommendation by multimedia player platforms are probably the most noticeable applied artificial intelligence for daily use. On top of that, the most recent advances in artificial intelligence or AI has been getting inside people's home with smart speakers, which in this example is Amazon's Amazon Echo, a home personal assistant called "Alexa".

How do this question that human asks to be processed with computers? This area is being covered by the subset area of Artificial Intelligence called Natural Language Processing. Natural Language Processing or NLP's definition as stated by Matt Kiser, "is a way for computers to analyze, understand, and derive meaning from human language in a smart and useful way" [1]. Tasks such as translation, sentiment analysis, automatic summarization can be performed by developers by taking advantage of NLP. Amazon Echo's Alexa is also an appliance of NLP subset called Natural Language Understanding or NLU, in a form of chatbot.

Chatbot is program that simulates conversation between humans and computers. Chatbots has been around since 1960s, with the first being ELIZA, a chatbot developed by Joseph Weizenbaum mat MIT Artificial Intelligence Laboratory. ELIZA does not bring any learning support or intelligence, based on certain keywords the said bot uses string substitutions and canned responses [2]. Since then chatbots has been significantly upgraded. Apple has also come up with voice enabled "intelligence assistant" since 2001, called "Siri". Since then, Siri's has been improved over time with an astounding 76% accuracy on answered question [3]. This means that 76% questions asked by the user, Siri...
understands. The other 24%-time Siri does not, she forwards the question to a search engine query.

Chatbots are getting major attention in the market nowadays. With Siri, Alexa and another intelligent assistant, all are trying to accomplish the same goal, which is to enhance user experience [1]. Having a chatbot in a company means having an ever-resting customer service. A lot of businesses are also coming up with chatbots to boost user experience. Considering this trend in using chatbots, creating a college personal assistant for Binus International students is expected to enhance the current information system used by the university.

The existing web portal has been running since March 2017. It provides students with information such as class and exam schedules, course materials, and campus admissions. Unfortunately the site’s technology has been outdated in a lot of area, making the website slow to respond and inaccessible at times. By creating a chatbot that understands natural language, students can have faster access to information and have an enhanced experience while using the web portal. To optimize the performance, the chatbot application will be created in a form of desktop and mobile platform.

2. Theoretical Foundation

Chatbots are only one of the applications of natural language processing field. Text classifications, sentiment analysis, text summarizer is one of the few fields using NLP. Not all of NLP field uses machine learning, though a lot of advanced systems uses machine learning to further enhance. Because there are a lot of preprocessing in the natural language pipeline, it is worth explaining the majority of them.

2.1. Tokenization

Taken from Cambridge University Press, tokenization is a task that takes a character sequence and chopping it into pieces that we call tokens [4]. The world as shown in the Figure 1 below:

```javascript
console.log(tokenizer.tokenize("string to tokenize"));
// ['string', 'to', 'tokenize']
```

**Figure 1. Inline Tokenizer Syntax**

2.2. String Distance

After the strings are tokenized, it can be processed individually. By processing it individually, the system may perform a string distance. String distance is a measurement of two strings, which in this case is the source string and target string. The calculation of distance involves in deletion, inserting or substituting the source into target string [5].

Currently, NLP framework nowadays supports a number of string distance algorithms. There are Damerau-Levenshtein Distance, Jaro-Winkler Distance and Sørensen-Dice coefficient, as shown in the Figure 2:

```javascript
console.log(natural.levenshteinDistance("ones", "onex"));
// Output: 1
console.log(natural.levenshteinDistance("one", "one"));
// Output: 0
```

**Figure 2. Inline Levenshtein Distance**

A great explanation by Wolf Garbe (CEO of a peer-to-peer search engine called FAROO), about Damerau-Levenshtein distance [6]:

“Both try to find the dictionary entries with the smallest edit distance from the query term. If the edit distance is 0 the term is spelled correctly, if the edit distance is <=2 the dictionary term is used as spelling suggestion.”

2.3. Stop Words

Stopwords are words that are not containing significance to the search query. Every language should
have a list of stopwords. We can use English as an example, being the most common are the, is, at, which and on [7].

To take the simplest possible case, take the string “Show me the latest news about Jakarta” and the intent “SHOW_JAKARTA_NEWS”. After applying the stop words, the string will be cut to only “latest news Jakarta” which could make the classifier score much higher on the “SHOW_JAKARTA_NEWS” intent than other intents, let’s say “SHOW_BALI_NEWS”.

2.4. Stemmer

A word can come in many forms, “play”, “playing”, “played”. Stemmer or stemming algorithm comes into play here by reducing a word to their word stem or root word, in this example, all the words above will be reduced to only “play”.

There are a couple of stemming algorithms used by language processing frameworks, being the most used are Porter and Lancaster Stemmer.

2.5. Part-of-speech tagging

Part-Of-Speech Tagging or POS Tagging is essentially a task of tagging a text to its corresponding grammar, in English language it would be noun, verb, adjective, adverb, pronoun, preposition, conjunction and interjection. The POS Tagging is illustrated in Figure 3 as follows:

![Figure 3. Inline Part-of-Speech Tagging](image_url)

2.6. Classifiers

The whole steps above are preprocessing techniques. Up until the data reaches classifier, data should be already cleansed to make the result of the classifying algorithm sharper. Figure 4 below displayed the Logistic Regression, Naive Bayes, Support Vector Machine (SVM), Neural Network, as a small example list of a classifier algorithms

```
var natural = require('natural'),
classifier = new natural.BayesClassifier();

classifier.addDocument('i am long qqq', 'sell');
classifier.addDocument('buy the q\'s', 'buy');
classifier.addDocument('short gold', 'buy');
classifier.addDocument('sell gold', 'sell');

classifier.train();

console.log(classifier.classify('i am short silver'));    //buy

console.log(classifier.classify('i am long copper'));    //sell

console.log(classifier.getClassifications('i am short silver'));

```

![Figure 4. Bayesian Classifier Example Code](image_url)

Take the above picture as an example. There are two intents registered which are buy and sell. The classifier has been trained a set of queries to their matching intents. Comes a new form of query “i am short silver”. The Naive Bayes algorithm will classify this as a buy intent.

On why it classifies it as a buy intent, if the getClassifications were to be printed, it outputs a score like displayed in Figure 5 below:

```
[ { label: 'buy', value: 0.3999999999999997 },
  { label: 'sell', value: 0.1999999999999995 } ]

```

![Figure 5. Bayesian Classifier Score from Above-Code](image_url)
The buy scores out dominants the sell score, making the query “i am short silver” classified to the buy intent. This is what it meant by using stop words and stemmers to score the classifier higher. If “i am short silver” were to be stripped to only “short silver” the classifier algorithm will be able to classify the query easier to the matching intent by outscoring the other (incorrect) intents. This section will focus on the part of methodologies that will be used for this paper. The system development methodology that will be used would be object-oriented analysis and design (OOAD) with waterfall method. Reference [8] explains that waterfall is traditional SDLC where scope is fixed with estimated time and cost whereas good for project with scopes are known and fixed which suited for this paper. For requirement gathering, the research utilize focus group discussion and existing document investigation that provided by XYZ.

3. Existing Solution for Bike Computer Application
Several solutions have been offered by Binus International for students to access information, though as time goes these solutions has become less attractive by not being engineered for mobile.

3.1. Binus Maya
Binus Maya is a web application platform that serves information about students and lecturers’ activities and learnings. Students can access information such as schedules, grades, materials, finances etc. Every question about campus life, most of the time is answerable by Binus Maya. Lecturers can also use this platform for uploading materials. For this purpose (online materials) Binus Maya provides a great way for e-learning.

3.2. Binus Campus Solutions
Binus Campus Solutions or BCS, is also one of the campus solutions offered by Binus International, similar as Binus Maya. There are 6 main features that this web application holds, which are academic planning, enrollment, campus finances, campus personal information, academic records and degree progress/graduation.

3.3. BeeBot
Binus has also released a chatbot called BeeBot, that is available through a popular messenger platform called Telegram. Right now, BeeBot serves six features that student can use that is to check courses (today and weekly schedule), GSLC (Guided Self Learning Class), forum, messages, news and finances. The features are shown in the Figure 6.

Figure 6. BeeBot Usable Commands
4. Problem Solution

4.1. System Architecture

The proposed application’s architecture will be implemented with a headless CMS architecture. All three servers, API Server, Binus Chat App and Binus Chat CMS will be built with Node JS. Binus Chat CMS and Binus Chat App may access the data through RESTful URL endpoints from the API Server. The author’s work is the ones being squared and bolded, which are creating the API server, CMS application and push features [9]. The application diagram is pictured in Figure 7.

The API Server is a Node JS RESTful API web services that is created for scalability issues and decoupling the system. This server will contain the database logics for the client’s views. This will be useful when the developers want to expand the Binus Chat App to another messaging platform.

Students may access the chatbot application through desktop, smartphones, or tablets by sending a request to the server, while interacting with the data from URL endpoints to the API Server and API.AI, only to fetch data. There will also be messages being pushed to the student’s side, delivered by administrators of the Content Management Systems. The push messages come in two kinds, one being a Firebase Cloud Messaging Push message and, the other being the actual message sent to the Binus Chat App client through Socket.io.

Administrators may access the Content Management System through desktop. Identical to the Binus Chat App, Binus Chat CMS also have access to the API Server to interact with the data from URL endpoints to the API Server. Administrators have the control to push messages to the Binus Chat App users.

At first glance, the author does not think that it needs a Content Management System, because of the existence of BinusMaya’s internal API. However, at designing stage, it is proved that a new form of Content Management System is needed to be built in order to upscale the chatbot application’s user experience. The chatbot application has a push technology features. The Content Management System that is built by the author has a function to broadcast (or push) message to users of chatbot application, creating a feel of two-way conversation on the chatbot user experience.

The natural language processor that powered the chatbot application has been provided by a third-party service called api.ai in a form of cloud web services. When students interacted with the chatbot application, the message goes through api.ai for the intents to be parsed. After the server knows the intent, it will next interact to the database to which data it will process, or no database interaction at all.

The CMS and chat application will be implemented in a Model-View-Controller (MVC) architecture provided by Express.js web framework. The way it is built is a little modified version from the out of the box MVC structure provided by Express.js, making the controller separated to the routing logic. Figure 8 shows the solution design.
The main reason of using MVC pattern, it is less complicated for the code to be maintained and easier to organize by having a separation of concerns and loosely coupled of components. This will also make it to be less difficult to modify a section of a code, without the other section to be requiring extensive modifications.

4.2. Use case diagram

The use case diagram on the figure depicted in Figure 9 below is for the chatbot and the CMS application. Two actors for this use case, student which is the front-end user and an admin for the backend or CMS user. The scope of the author’s would be the admin side and the author’s partner would be on the student’s side, while populating the database is contributed work from both.

5. Testing and Implementation

Upon creating this thesis project, the application is built on the author and the partner’s author machine. To maintain the code from the collaborative work of both authors, BitBucket is used. BitBucket is Github and web interfaces combined, allowing user to do basic Github operations such as merging and reviewing code as illustrated in Figure 10.
For local development, the author is using SourceTree desktop application. SourceTree is a Git client for Windows and Mac, giving a visual representation of Git. The reason of using this is to get better understanding of the Git itself from its native command line interface.

Git-Flow is also enabled on this (SourceTree) project. Git-Flow is a branching model for Git, made by Vincent Driessen [10]. It is used to divide branch into features, develop, release, master and hotfix.

A simple explanation, every time a new feature is made, the author creates a new feature branch. When finished it is merged to develop. After all features for a certain period of time is finished by both authors, it should be merged again to develop branch and creates a release branch along with the new version of the application. If everything goes as planned, it will be merged back to develop and master branch. Every bug fix will go to hotfix branch. Git-Flow is important in this project for it is much suited for collaboration project and scaling the development [11].

6. Discussion
The thesis statement for this project is to create a new means of information system for Binus International students that provides less friction, faster information retrieval and hopefully reaches out more students than the existing system. To provide a better understanding on how this project has made, the author will show a compare and contrast of the existing system against Binus Chatbot. Points below will compare and contrast the chatbot application created with Binus Maya.

6.1. Friction
One of the main designs of the Binus Chatbot is to create less friction of information retrieval to the existing information system. For BinusMaya, (after logging in) students must go through one or more clicks to retrieve the information being searched. Binus Chatbot on the other hand, as a text-based information system (after logging in) requires students to type questions to get the information’s being searched.

Because of Binus Chatbot is text based, it removes menu and submenus clicking systems, leaving users to only type on the information’s being searched. Moreover, to gain another information, for instance, when a student wants to examine score after examining schedule, they would have to just type another question, rather than maneuvering the menus twice.

6.2. Platform Compatibilities
BinusMaya is optimized for the desktop application. While it is accessible from any mobile devices, it is at times challenging to maneuver when the menus and submenus goes off-screen. Binus Chatbot is built upon UI design that is built for desktop, while emphasizing for mobile devices. The application offers a native-like app experience, with home icons, splash screen and push notifications. Again, because it is a chat-based information system, there is only one menu that can be used which is the settings menu, making students not bother about going through numerous submenus.

6.3. Speed of Information Retrieval
Being one of the most important measures on designing the system, the speed of information retrieval is key in Binus Chatbot application. It must be stressed beforehand that the speed that is being discussed here is about having less friction to Binus Maya, and not about page load speed, database access speed, etc.

Because that Binus Chatbot is a text-based information retrieval, it does not have to worry about rendering unwanted UI displays. This leaves the application to only worry about database queries. This is contrast to Binus Maya. For example, when students want to check course grade, they must first render the dashboard that has numerous items, and then the grade page itself.

As for the replies from the user acceptance testing, four out of five respondents that the Binus Chatbot is the faster way to retrieve information, opposed to Binus Maya and BCS. From this point, it
is proven that Binus Chatbot triumphs over Binus Maya in terms of information retrieval speed. The users’ acceptance to Binus Chatbot is shown in Table 1.

**Table 1. User Acceptance Testing Results**

| Gender | Age   | Version | Interface | Feasibility | Binmay or BCS | Faster | Important | Less Important | Rate |
|--------|-------|---------|-----------|-------------|---------------|--------|-----------|---------------|------|
| Male   | 20-22 | Mobile  | 3         | 4           | 3             | 4      | Financial | SAT and Community Service | 4    |
| Male   | 23-25 | Mobile  | 3         | 4           | 4             | 4      | Financial | SAT and Community Service | 4    |
| Male   | 20-22 | Mobile  | 3         | 4           | 4             | 5      | Financial | Score          | 3    |
| Female | 20-22 | Mobile  | 4         | 5           | 5             | 4      | Course    | SAT and Community Service | 5    |
| Male   | 23-25 | Mobile  | 4         | 4           | 3             | 4      | Financial | SAT and Community Service | 4    |
| Female | 17-19 | Mobile  | 3         | 4           | 3             | 5      | Financial | SAT and Community Service | 4    |
| Male   | 20-22 | Mobile  | 4         | 3           | 4             | 4      | SAT and Community Service | Financial | 4 |
| Female | 23-25 | Mobile  | 3         | 4           | 4             | 4      | SAT and Community Service | Financial | 4 |
| Male   | 20-22 | Mobile  | 3         | 4           | 5             | 4      | Course    | SAT and Community Service | 4    |
| Male   | 23-25 | Mobile  | 3         | 4           | 4             | 5      | Financial | Course         | 4    |
| Male   | 20-22 | Mobile  | 3         | 4           | 4             | 4      | Financial | Score          | 3    |
| Male   | 23-25 | PC     | 3         | 4           | 5             | 4      | Course    | Score          | 4    |
| Male   | 20-22 | PC     | 3         | 5           | 4             | 5      | Course    | Score          | 4    |
| Male   | 23-25 | Mobile  | 5         | 4           | 4             | 5      | Financial | Score          | 4    |
| Male   | 20-22 | Mobile  | 3         | 4           | 4             | 4      | Financial | Score          | 4    |
| Female | 20-22 | Mobile  | 4         | 4           | 4             | 4      | Financial | SAT and Community Service | 5    |
| Female | 23-25 | Mobile  | 4         | 4           | 4             | 4      | SAT and Community Service | Score | 4 |
| Male   | 20-22 | Mobile  | 3         | 4           | 4             | 4      | SAT and Community Service | Score | 4 |
| Male   | 20-22 | Mobile  | 4         | 4           | 5             | 5      | SAT and Community Service | Score | 4 |
| Male   | 20-22 | Mobile  | 5         | 5           | 4             | 5      | SAT and Community Service | Score | 4 |
| Female | 20-22 | Mobile  | 4         | 4           | 4             | 4      | SAT and Community Service | Financial | 5 |
| Male   | 20-22 | Mobile  | 4         | 3           | 4             | 4      | Financial | SAT and Community Service | 4    |
| Male   | 20-22 | Mobile  | 4         | 4           | 4             | 5      | Course    | Financial | 5    |
| Female | 20-22 | Mobile  | 4         | 4           | 4             | 4      | SAT and Community Service | Score | 4 |
| Male   | 17-19 | Mobile  | 4         | 4           | 4             | 4      | SAT and Community Service | Score | 4 |
| AVERAGE| 20-22 | Mobile  | 3.607     | 4.036       | 4.000         | 4.571  | Financial | SAT and Community Service | 4.036 |

7. Conclusion

As messaging platform users surpass the number of social network users, chatbot is a new means for companies to connect with people and customers. Hence, introducing chatbot is a timely alternative to Binus International’s existing information system called Binus Maya.

The chatbot created is designed to be faster to retrieve information, supports for desktop and mobile and interactive in natural language. The main frameworks being used are API.AI for the Natural Language Processing library, Progressive Web Apps to deliver native-like experience and a headless CMS architecture for scalability issues. Third-party NLP libraries provides a quick development stage and generate user impression about the chatbot. Choosing headless CMS as the architecture was an investment for the future. When Binus Chatbot needs to be used in another existing messaging platform, maintainers would not have to code the database logic again, since the author has already built the API for the database logics.

It can be concluded that the creation of chatbot has enabled a faster information retrieval from the existing information systems. In terms of reaching out more students, it cannot really be measured since it will require having the majority of the students of Binus International to use the app and examine over time.

References

[1] Martin E 2016 3 reasons 2017 is the year to develop a company chatbot Online: [https://thenextweb.com/opinion/2016/12/09/3-reasons-2017-year-develop-company-chatbot/#.tnw_072Wqknl](https://thenextweb.com/opinion/2016/12/09/3-reasons-2017-year-develop-company-chatbot/#.tnw_072Wqknl)
[2] Wallace MJ Eliza, computer therapist Online: http://www.manifestation.com/neurotoys/eliza.php3
[3] Ritchie R 2015 Siri ‘crushing’ competitors at language accuracy Online: https://www.imore.com/siri-crushing-competitors-language-accuracy
[4] Press C U 2009 Tokenization Online: https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html
[5] Gilleland M 2016 Levenshtein distance, in three flavors Online: http://people.cs.pitt.edu/~kirk/cs1501/Pruhs/Spring2006/assignments/editdistance/Levenshtein20Distance.htm
[6] Statista 2017 Most popular mobile messaging apps worldwide as of January 2017, based on number of monthly active users (in millions) Online: https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/
[7] Schlicht M 2016 How bots will completely kill websites and mobile apps Chatbots Magazine Online: https://chatbotsmagazine.com/how-bots-will-completely-kill-websites-and-mobile-apps-656db8e6fc03#.a9fwz51kk
[8] Satzinger J, Jackson R, Burd S, 2012 System analysis and design in a changing world (Boston, MA: Course Technology, Cengage Learning)
[9] Walsh W 2010 WebSocket and socket.io Online: https://davidwalsh.name/websocket
[10] Introducing gitflow Online: https://datasift.github.io/gitflow/IntroducingGitFlow.html
[11] 2017 Entities Online: https://docs.api.ai/docs/concept-entities