An NLP-based Chatbot to Facilitate RE Activities: An Experience Paper on Human Resources Application

Pedram Khatamino[a], Mustafa B. Çamlı[a], Bilgehan Öztekin[a], Umutcan Gözümoğlu[a], Emre Tortumlu[a] and Hüseyin M. Gezer[a1]

[a]*Westerops Software and Information Technologies Company, Istanbul, Turkey*

Abstract

Companies use multiple applications and platforms on a daily basis to manage their operations. The context switch between these applications causes distractions in employees. We propose a chatbot integrated to the main workspace of a company as a quick interface to multiple applications to prevent employees from switching to other applications. This paper presents our experience building the prototype of this chatbot, demonstrate its usage for a human resources application and provide a use case for requirements engineering activities.

Keywords

Chatbot, Natural Language Processing, Requirements Engineering

1. Introduction

A chatbot is a computer program that uses Artificial Intelligence (AI) and Natural Language Processing (NLP) to automate responses to user queries, simulating human conversation. Chatbots can facilitate users to find the required information. Using natural language, the chatbot technology responds to users' questions and requests through the textual and/or audio input without human intervention.

A company deploys multiple applications and platforms for different purposes such as human resources, project management, accountancy operations, and issue management. Switching the context within these platforms causes fatigue and confusion for clients and employees. In order to prevent this, we aim to build a Super App as a workspace which contains Mini Apps for the applications that are routinely used. A chatbot is integrated to this Super App as an interface that provides quick access to the connected applications. In this paper, we describe our experience of building a chatbot for such a workspace. We document the requirements (Section 2), provide the architecture, and the development details (Section 3) demonstrate the use for a human resources application, and relate to another use case related to requirements engineering (Section 4).

2. Chatbot Requirements

In this section we briefly provide the requirements for the chatbot. The requirements are elicited through semi-structured elicitation interviews with the other employees of our company. The system- to-be is a chatbot that acts as a quick interface to multiple applications that are used within the company on a daily basis in order to prevent the frequent context switch by the employees. Below we list the requirements.

- R1: Chatbot shall allow the user to type the kind of the application they want to use.

In: F.B. Aydemir, C. Gralha, S. Abualhaija, T. Breaux, M. Daneva, N. Ernst, A. Ferrari, X. Franch, S. Ghanavati, E. Groen, R. Guizzardi, J. Guo, A. Herrmann, J. Horkoff, P. Mennig, E. Paja, A. Perini, N. Seyff, A. Susi, A. Vogelsang (eds.): Joint Proceedings of REFSQ-2021 Workshops, OpenRE, Posters and Tools Track, and Doctoral Symposium, Essen, Germany, 12-04-2021
EMAIL: pedram.khatamino@westerops.com (P. Khatamino)

- R2: Chatbot shall terminate the process when an application that is not supported is chosen 3 times.
- R3: Chatbot shall allow the user to type specific parameters according to the application type.
- R4: Chatbot shall detect the dates written in natural language.
- R5: Chatbot shall calculate the dates based on the detected dates from the conversation.
- R6: Chatbot shall allow the user to specify dates from a calendar drop-down when the application type is human resources and the dates cannot detect from the conversation.
- R7: Chatbot shall allow the user to confirm the Chatbot's responses.
- R8: Chatbot shall warn the user if the user's message is misunderstood or not entered.
- R9: Chatbot shall show all parameters of the request at the end.

R10: Chatbot shall pass the user's responses to the API and pass the response from the API to the use

Although we aim for a chatbot that is connected to multiple applications through their APIs, we choose a human resources application as our pilot application. In Figure 1, we provide the use case diagram of the chatbot only for cases related to the human resources application. The user can type messages via the Chatbot GUI in order to communicate with the Chatbot. The purpose of these messages can be enhanced in the future. Our focus is to create a leave request using the chatbot. The user can create a leave request by providing her/his name, defining the type of the leave, specifying the start and the end dates of the request. The user needs to approve the request at the end. If the start and end dates are not understood, a calendar shall be shown by the Chatbot.

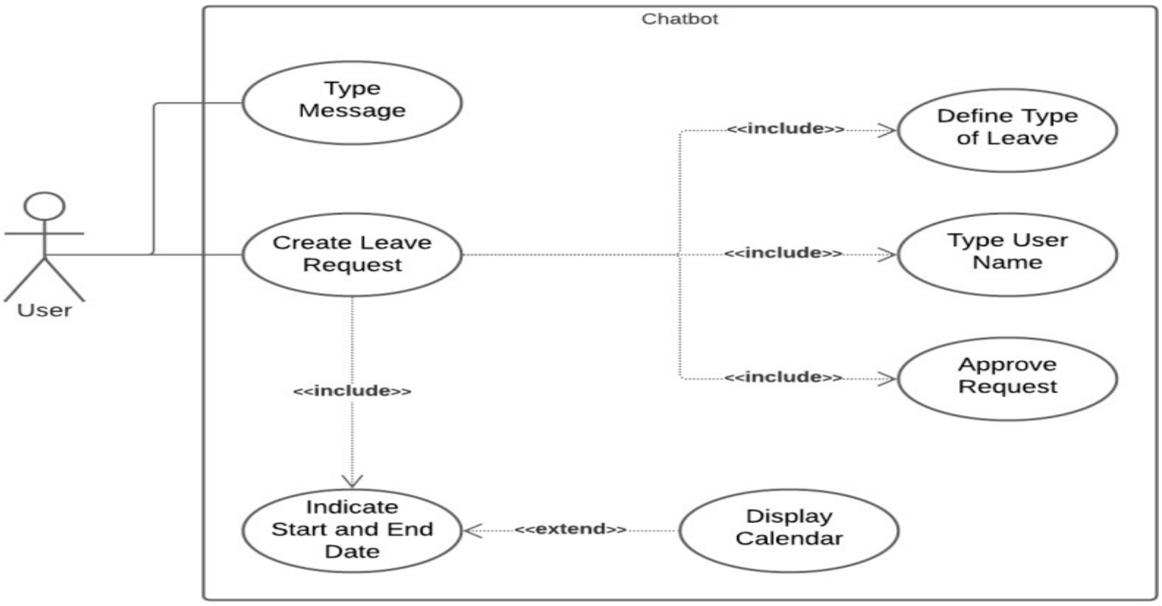

Figure 1: The use case diagram of the chatbot for the human resources application.

3. CHATBOT ARCHITECTURE

In this project, we are planning to use a multi-tenant structure for increasing security however, prototype is sunning on SQLite. In addition, it is aimed to establish an automatic structure as much as possible with artificial intelligence and rule-based algorithms.

In Figure 2, the architecture diagram is shown. The user interacts directly with the Chatbot framework via the Chatbot GUI. The Chatbot GUI sends the user's request and takes the response by the Message Channel. Moreover, the router, NLP Engine and the database are located in AWS Cloud. The router is a REST API that makes the connection between the Message Channel and the NLP Engine. The NLP Engine consists of three parts which are the Application Type Classifier, the Leave Type Detector and the Date and Time Extractor. They take an input from the Message Channel via the Router and gives proper outputs back; furthermore, the NLP Engine uses the Database for training necessary machine learning models.

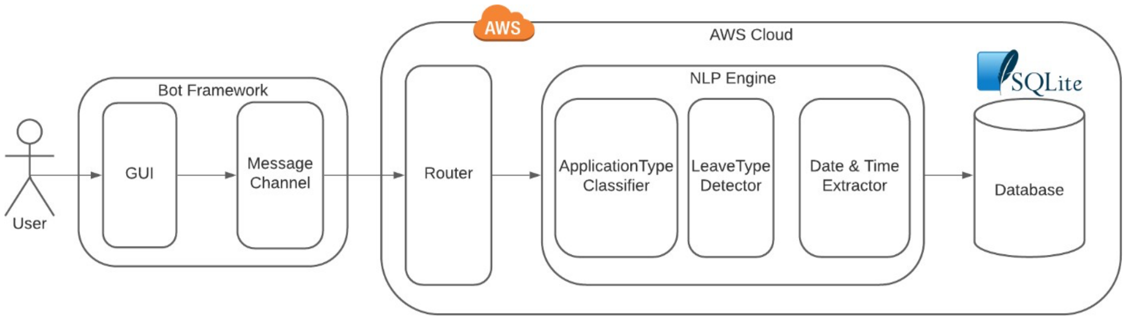

Figure 2: Architecture diagram.

3.1. Interaction FLOW

The Figure 3 presents the interaction flow; at the very beginning, the Chatbot shows the welcome message and then ask name of the user. After that, the Chatbot offers help to understand the user's need and takes input from the user. The received input goes directly to the API. If the user types the whole request parameters at once, the API responses as correct detection, or else incorrect detection. The input goes through two API if the response from the API is correct detection. Later on, if the user approves the parameters, the API is called again to create the request and then the chatbot is finished or, else the user is connected to human call; however, if the whole request parameters are marked as incorrect detection from the API, then the user shall type each parameter by one by. Firstly, the Chatbot shows the type of applications so that the user can choose the application which is needed. If the selected application is other than the human resources application, the Chatbot is terminated due to the fact that applications except the human resources application are not ready to be used yet. However, if selected application is the human resources application, the Chatbot asks type of the leave and the user answers. Secondly, the Chatbot opens the calendar drop-down twice for start and end date with a list of dates and times to be chosen so that the user can choose the start and end date of the request. Finally, the parameters are shown to the user in order to check the user's approvement. If the user approves the parameters, the API is called again to create the request and then the chatbot is finished or, else the user is connected to human call.

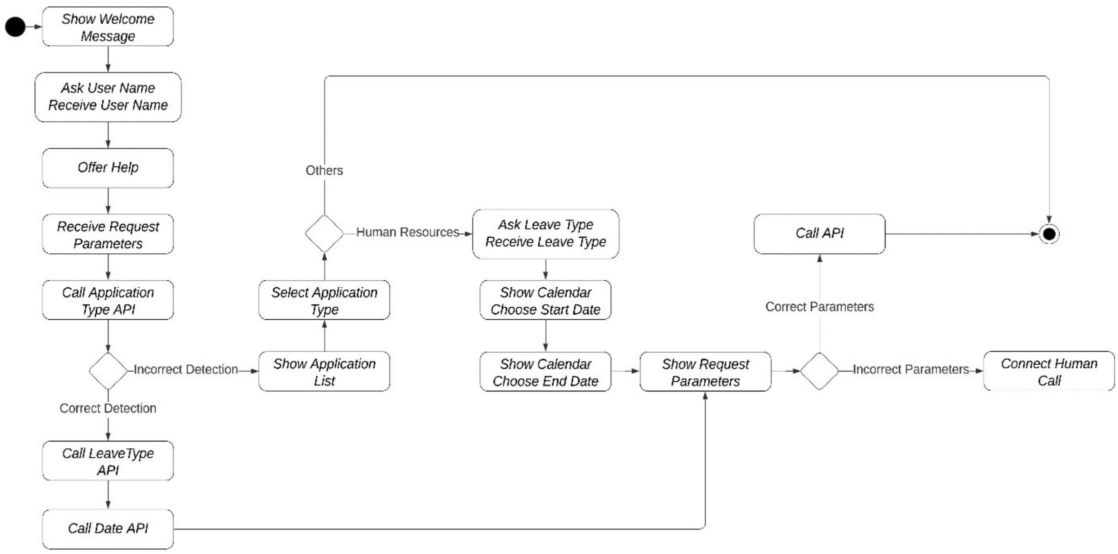

Figure 3: Chatbot interaction flow.

3.2. Technologies

We use Python as the main development language due to our expertise and the available libraries for NLP, machine learning, and chatbot development. For NLP, we use NLTK [1] and Spacy [2]. For machine learning related tasks, we use Scikit-learn [3] and Pandas [4]. We select BotStar [5] as the chatbot builder platform due to its model driven approach, easy deployment and integration, and detailed documentation. BotStar is a conversational bot software for building chatbots on websites and Facebook Messenger [6]. Most important feature BotStar presents is the visual editor which allows us to create flows with many built-in elements without the need of code. Connections between elements can include many conditions to control the flow. These conditions can work with external variables implemented through APIs. This extends to another important feature which is the extension of the chatbot's capabilities with developer tools. BotStar comes with a built-in code editor that lets programmers execute complex code within the context of the chat flow. BotStar also has a CMS (Content Management System) to create and manage digital contents for the bots. BotStar also includes customization for behavior such as mimicking human behavior as if the bot is reading or typing a message. BotStar can be integrated with common apps like Google Sheet [7], Zapier [8], Stripe [9], Integromat [10]. We connect our backend and chatbot UI with JavaScript. Docker [11] allows us to develop, ship, and run our backend. We use Nginx for web serving this project. Finally, Amazon Web Service [12]is deployment server of this project.

3.3. NLP PIPELINE

We collect text data from open-source websites to create a data set for our project. We could not benefit from processing such as web scraping, since the target-related text data was not available on a particular website. We searched with manually determined keywords for data consisting of three main labels (Human Resource, Project Management, Accounting). For instance, our search consists words such as sickness, sample leave request letters and mails, different religious and national holidays and corporate positions for HR label. In a week, we prepared a data set of approximately

1500 samples with a team of three. These patterns were collected in sentences and paragraphs.

The NLP pipeline starts with the classification of the message of the user after the greeting. The chatbot supports English. We use NLTK for processing the messages before classification as it supports English. In the training part, the models to be used for the classification are chosen as the basic and widely used models in the Scikit-Learn library since the problem is partially easy to classify. The models are Gaussian Naïve Bayes, Bernoulli Naïve Bayes, Multinomial Naïve Bayes, Support Vector Machines (SVM), linear SVM, NuSVC, Logistic Regression, SGD Classifier and MLP. Furthermore, the parameters of all models are selected as the default values defined by the Scikit-Learn library. Moreover, the train set consists of 75% of the data and the rest, 25% of the data, is used as the test set to validate our models. Also, existing same amount of each class in train and test data guaranteed by manually designed split approach. At the beginning, the train set was trained with these classifiers separately and the results are printed. Table 1 shows the accuracy results for each label and overall, of models on the test set.

Table 1

Accuracy of the labels and the overall

Classifier	Human Resources	Accounting	Project Management	Overall
Gaussian N.B	0.80	0.88	0.86	0.84
Bernoulli N.B	0.92	0.91	0.97	0.93
Multinomial N.B	0.88	0.88	0.98	0.91
SVM	0.94	0.90	0.95	0.93
Linear SVM	0.94	0.93	0.97	0.94
Nu SVM	0.94	0.94	0.96	0.95
Logistic Regression	0.94	0.93	0.96	0.94
Multilayer Perceptron	0.93	0.95	0.95	0.94

It can be seen that the Nu-Support Vector Machines performs as the best model among all models. Furthermore, one of the Ensemble Learning method is utilized which is called hard voting classifier in order to increase accuracy. The hard voting classifier basically includes several classifiers and every individual classifier vote for a class, and the majority wins. In our work, the hard voting classifier contains all classifiers in Table 1 and gives average accuracy for each class and same overall accuracy with the Nu-Support Vector Machines. Therefore, the hard voting classifier is not needed due to its poor work.

Table 2

Sample sentences and related class

Comment	Result
I want to sick leave for next week.	Human Resources
I want to receive the balance sheet for the last month.	Accounting
I want to see activities regarding DWS project.	Project Management

Term Frequency - Inverse Document Frequency (TF-IDF) is a statistic that aims to better define how important a word is for a document, while also taking into account the relation to other documents from the same corpus [13]. TfidfVectorizer, transforms text to feature vectors that can be used as input to estimator. We use scikit-learn TfidfVectorizer with some bult-in text preprocessing methods and manually designed operations such as tokenizer, stop words, lowercase, removing numbers, which are the main concepts of NLP text classification.

Additionally, stemming method is one of our text preprocessing operations. Stemming could be defined as reducing inflectional and some derivational forms of words which leads software engineers to get better results. By stemming, related words are reduced to their base forms. While training datasets, for instance it is better to reduce –ing or –s suffixes due to the fact that these suffixes do not change the meaning vastly. NLTK contains both Porter and Snowball stemmers and for the English dataset, the Snowball stemmer are used. Likewise, our preprocessing operations consist of removing stop-words by using NLTK library. We used Scikit-learn data split method for splitting train and test data.

Part of speech (POS) tagging is categorizing words in a corpus depending on the definition of the word and its context. For instance, most verbs in English also could be used as nouns in sentences and by tagging them correctly would result in better results. Up to now, only the English dataset is tagged. Part of speech tagging is mostly used in date and duration detection. Table 3 shows the output of our NLP for date-duration detection.

Table 3
Example Chatbot Inputs

Comment	Result
I am sick so I will be off from tomorrow to Thursday.	[2021/02/16 09:00, 2021/02/18 18:00, sick leave]
I want to get a maternity leave for 2 months.	[2021/02/15 09:00, 2021/04/15 18:00, maternity leave]
I need to pick my son up from his school. I want to take personal leave for 2 hours.	[2021/02/15 14:05, 2021/02/15 16:05, personal leave]

3.1.1 Named Entity Recognition (NER)

NER a process where an algorithm takes a string of text (sentence or paragraph) as input and identifies relevant nouns (mainly people, places, and organization etc.) that are mentioned in that string. In this project, we use the NER structure to automate which tasks are assigned to which personnel when teams are assigned tasks, and to determine the reason for which personnel take time off when they get leave request [2]. Figure 4 shows our NER concept.

Figure 4: Sample sentence for demonstration of NER concept.

3.4. Use Case for RE Applications

Collecting and analyzing user feedback are two RE activities that can be facilitated with our chatbot in the future. Instead of providing the customers with forms or interfaces that can be intimidating or assigning a human that can be costly, we plan to deploy a chatbot for collecting issues related to our software and collect necessary information about bugs and feature requests using a chatbot. In addition, when a message is entered in our chatbot about a relevant field within the company, that field will be added to our data, and in the future, we will identify which areas our customers have less access to and make improvements in those areas.

4. Related WORKS

Chatbot Platforms. There are many successful chatbot making platforms such as Botstar, WotNot, Intercom, Drift Chatbot, Landbot.io, LivePerson, Bold360, Octane AI, etc. Each of them has special features which are designed for competition in marketplace. In order to choose between different chatbot platforms, we have to pay attention to certain criteria. These criteria mainly cover the following topics: identifying the use cases, integrations, natural language and AI capabilities, training, pricing.

In researching phase, first consideration has to be "what is the use case for using the chatbot in the project". A thorough understanding of the project's use case can help the researcher determines what exactly expectation of the chatbot. It is vital to have the right chatbot integrations in place to get the finest results out of the chatbot platform. The user-chatbot conversation is one of the most critical components that make chatbots so intriguing for the users. However, consider a platform which supports NLP and has AI capabilities to expand the use case and chatbot's capabilities down the line. Organizations need a human-independent chatbot solution, that supports continuous learning and gets smarter with each conversation using machine learning and semantic modeling. Today, most of the chatbot platforms use a combination of a pay-per-call, monthly license fee, and pay-per-performance pricing models. A researcher needs to go with a chatbot pricing plan that is predictive, guarantees savings and allows him to pay according to his achieved or non-achieved goals.

Experiment Reports. A Review of AI Based Medical Assistant Chatbot aims to address the language divide between consumer and health care professionals by delivering direct answers to customer inquiries [14]. Instead of searching through the web-based set of theoretically relevant text, it is easy to create question response forums to discuss those queries. Main use-case of the chatbot's NLP feature is to act as a diagnosis and treatment authority. Text classification provides the diagnosis and recommendation is sent by the chatbot in return. This project consists of text classification to diagnose the patient with certain illness through multiple sequence questions.

Another work includes a chatbot created for ticket reservation which is can classify the entered message [15]. It uses NLP concepts to analyze the message such as POS method for tagging to make sense of the text entered. When NLP block using POS, a rule has been created from certain keywords such as departure city, destination city and flight city. This study was carried out for customers to buy tickets faster.

In our study, we use both a model-based and a rule-based structure while making the necessary classifications. There are many keywords related to the classes we have determined in the rule-based structure in field of detecting date, time, leave type, humanoid answers responses, etc. If no result is found with the model we trained from our own data, we aimed to find a result with the rule-based algorithm we created.

5. Conclusions

In today's technologically advanced business environment, chatbots help the business stay accessible round the clock, without investing heavily in hiring extra customer support reps. Our work's important point is to prevent the loss of time, effort and resources by directing the text as automatically as possible in our Super App project, which will cover all in-house text, and provide fast and easy solutions to the employees.

In conclusion, we introduce an NLP-based chatbot solution to the fatigue caused by multiple platforms in business environments. Companies use multiple platforms to run their operations which may not be excelled by their employees and customers. We propose integrating a chatbot integrated to the Super App of the company. The chatbot is currently integrated to the human resources application. We also provide an RE related use case as part of our on-going work.

For future perspective we aim to integrate this concept for all HR operations plus Project management and accounting platforms (for example, Slack, Paraşüt, Teamwork, Clockify, etc.) which are supported by our main Super App. For this goal, we will research and determine the requirements for each app.

In the future, we want to update our data set, train deeper models, and advance in processes such as classification and text prediction. To collect this data, we are planning to collect data during beta tests from several companies, including our own company. Additionally, we want to contribute to the literature by preparing data set and pre-trained models in both English and Turkish languages with manually labeled data set.

RefereNCES

[1] NLTK. URL:https://www.nltk.org/
[2] Spacy. URL:https://spacy.io/
[3] Scikit-Learn. URL:https://scikit-learn.org/stable/
[4] Pandas. URL:https://pandas.pydata.org/
[5] Botstar. URL:https://botstar.com/
[6] Facebook Messenger. URL:https://www.messenger.com/
[7] Google Sheets. URL:https://www.google.com/sheets/
[8] Zapier. URL:https://zapier.com/
[9] Stripe. URL:https://stripe.com/
[10] Integromat. URL:https://www.integromat.com/en
[11] Docker. URL:https://www.docker.com/
[12] Amazon Web Service. URL:https://aws.amazon.com/
[13] Ramos, J. (2003, December). Using tf-idf to determine word relevance in document queries. In Proceedings of the First Instructional Conference on Machine Learning (Vol. 242, No. 1, pp. 29-48).
[14] Bulla, C., Parushetti, C., Teli, A., Aski, S., & Koppad, S. (2020). A Review of AI Based Medical Assistant Chatbot. Research and Applications of Web Development and Design, 3(2).
[15] Handoyo, E., Arfan, M., Soetrisno, Y. A. A., Somantri, M., Sofwan, A., & Sinuraya, E. W. (2018, September). Ticketing chatbot service using serverless NLP technology. In 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE) (pp. 325-330). IEEE.