The International Journal of Engineering and Information Technology

journal homepage:www.ijeit.misuratau.edu.ly

MISURATA UNIVERSITY

Design of Intelligent Chatbot for Stress Management

Eluheshi, A.
Libyan academy for postgraduate studies
adel.eluheshi@academy.edu.ly

Shlebik, A.
Libyan National Oil Corporation
Ashlibek@noc.ly

ABSTRACT - **This paper focuses on using natural language processing (NLP) in chatbots to manage stress in war-affected countries. A Java-based chatbot was designed to alleviate stress using two algorithms: TextRank and Stanford_CoreNLP. The problem was solved by integrating different languages using a plugin. The chatbot was tested with fifteen people and received positive feedback. Modifications were made based on user feedback, with journaling being a winner. However, the chatbot faced limitations like lack of Arabic language support and voice chat feature.**

Keywords - **ChatBot, Machine Learning, Sentiment Analysis, Web-Scraping, Text Summarization, NLP.**

I. INTRODUCTION

NLP is crucial in mental health practice due to its ability to understand human language and understand the meanings of underlying words. It has been proven to accurately identify and classify suicide ideation and attempts in Electronic Health Record data (92% and 83% accuracy) and extract 90% of symptoms in English text discharge summaries for patients with severe mental illness (SMI) and non-SMI diagnosis (87% and 60% respectively) [1]. The study by [2] and [3] utilized NLP-based models to predict suicide ideation, with the former showing high predictive value and the latter achieving a 63% accuracy rate in identifying depression-related social network users.

A chatbot (Chatter and Robot) is a computer program that uses AI (Artificial Intelligence) and NLP to understand users' questions and automate responses, simulating human conversation. There are three types of chatbots: simple, smart, and hybrid. Simple chatbots have limited capabilities and are rule-based, posing predetermined questions. Smart chatbots simulate near-human interactions and require programming to understand context. Hybrid chatbots combine simple and smart chatbots, making them a balanced tool for businesses to interact with users.

Received 08 Apr , 2024; revised 14 May, 2024; accepted 15 Mar 2024.
Available online 08 Aug, 2024.

Examples include ordering pizza and medical diagnosis. Different types of chatbots are used in health care described here, Woebot, a chatbot therapist designed for young adults and graduate students, uses Cognitive Behavioral Therapy (CBT) to help patients express their thoughts and reduce depression symptoms [4]. Wysa and Babylon are chatbots that help manage thoughts and emotions using evidence-based CBT and guided meditation. While critics argue for a lack of human connection, these platforms can be helpful if designed and delivered humanely. As tech-based medicine advances, more robust AI-based chatbots and treatment platforms are expected [5], [6].

II. RESEARCH PROBLEM STATEMENT

As people around the world face everyday events that cause them stress, the need for an accessible and reliable way to deal with this stress is sorely needed. Almost everyone has a smart phone in his hands, ChatBot presents the best choice for applying AI techniques in handling stress problems. A hybrid (Rule-based and ML-based) chatbot is going to be built using multiple ML techniques with the intention of determining the stress factors and stress levels among users and offer recommendations on how to ease these factors, consequently, ease the stress level. The Research Aim is to assess the effectiveness of utilizing AI techniques and methods, i.e., NLP, Sentiment Analysis (SA), text summarization and supervised ML to reduce stress among users through designing a conversational user interface.

III. RELATED WORK

Chatbots can be more comfortable for people to disclose information due to their lack of judgment and judgment. Studies show that chatbot interactions offer emotional, relational, and psychological benefits, with a focus on personal disclosure [7] . Preliminary evidence suggests that chatbots could be beneficial in psychiatric treatment [8], but further research is needed to fully understand their effectiveness. A study by [9] found positive perceptions of chatbots for mental health, but noted their linguistic capabilities need improvement. A pilot study by [10] found chatbots to improve well-being and reduce stress, with high engagement. A study by [11] found empathic agents promising for hospitalized medical patients, preferring their discharge information. Conversational agents, which allow tailored, anonymous, and convenient access, may help overcome health literacy barriers and change attitudes and behaviors. This research work proposes a chatbot that maintains a fun and sympathetic personality throughout conversations, randomly selecting responses from a set of different syntax and meanings. The chatbot is free, works with or without internet connection, and is ideal for those dealing with dilemmas, offering 24/7 use and better results when online.

IV. MACHINE LEARNING ALGORITHMS

ML algorithms are different from other algorithms. With most algorithms, a programmer starts by inputting the algorithm. However, with ML the process is flipped. With ML, the data itself creates the model. The more data that is added to the algorithm, the more sophisticated the algorithm becomes. As the ML algorithm is exposed to more data, it can create increasingly accurate algorithm [12]. Various algorithms and computation techniques are used in ML processes. The most used supervised learning algorithms include linear regression, Logistic Regression (LR), k-nearest neighbors, decision trees, Random Forest (RF), gradient boosting machines, Xgboost, support vector machines and neural networks. As for unsupervised learning, the most popular algorithms are k means clustering, hierarchical clustering, and neural network.

V. SENTIMENT ANALYSIS

Sentiment analysis (SA) is an NLP technique used to determine data polarity, emotions, urgency, and intentions. It uses rule-based and ML-based methods, with rule-based analysis being more rigid and ML-based analysis being more detailed. Popular types of sentiment analysis include Graded SA**.** When polarity precision is important, polarity categories can be expanded to include different levels of positive and negative; very positive, positive, neutral, negative, and very negative. Aspect-based SA (ABSA) aims to analyze and understand people's opinions at the aspect level [13]. For example, the bike’s color, the cloth’s texture. Intent Analysis is a deeper understanding of the intention of the customer. This means that the intention of a particular customer can be tracked, forming a pattern, and then used for marketing and advertising [14].

Emotion detection allows going beyond polarity to detect emotions, like happiness, frustration, anger, and sadness. Emotion detection systems use either lexicons (i.e., lists of words and the emotions they convey), or use other advanced methods that include ML [15]. Examples of ML-based emotion detection systems: Stanford CoreNLP - NLTK - openNLP – spaCy. Stanford CoreNLP is a deep learning model that identifies sentence sentiment by splitting text into sentences and tokens. Sentiment is assigned to each sentence, and the mean sentiment value is used to estimate the sentiment. The average sentiment ranges from Neutral to Negative, with rare positive sentiments. Sentiment Analysis Challenges Machine-based Subjective Analysis (SA) uses Natural Language Processing and Text Analysis to extract subjective information from text, addressing challenges like subjectivity, tone, context, irony, comparisons, and neutrality [16].

A. Sentiment Analysis Tools

The SA tool is a software that analyzes messages to determine tone, intent, and emotion by breaking them into chunks and assigning a sentiment score. It's useful for companies engaging with customers on social media, live chat, and email [17]. This paper focuses on Python and Java tools for Support Vector Machines (SA), citing their ease of implementation, understanding, and support from a large community, including Scikit-learn, NLTK, SpaCy, OpenNLP, and Stanford CoreNLP.

B. Text Summarization

Text summarization is the technique to identify the most useful and necessary information in a text [18] Google uses text summarization techniques to show the summary of the article or the answer for the user’s query [19]. There are two approaches of text summarization, Extractive Summarization means an important information or sentence are extracted from the given text file or original document [18]. Abstractive Summarization Where the model forms its own phrases and sentences to offer a more coherent summary, like what a human would generate. This approach is much more difficult than extractive summarization [19].

C. Data Scraping

Is a technique in which a computer program extracts data from output generated from another program [20]. There are two types of data scraping; screen scraping and web scraping, which are defined as follows: Screen Scrapping is the process of collecting screen display data from one application and translating it so that another application can display it [21].Web Scraping is the process of collecting structured web data in an automated fashion. web data extraction is used by people and businesses who want to make use of the vast amount of publicly available web data to make smarter decisions [22]

VI. TOOL AND MATERIALS

Illustrates tools (IDEs, databases), materials (programming languages, libraries, plugins) used in developing the chatbot. It also explains the selection and evaluation of the implemented techniques. Different tools have been utilized according to the need and the target of the research, i.e., building an intelligent chatbot. IDE is an abbreviation for integrated development environment. It is a software for building applications that combines common developer tools into a single graphical user interface. Table

I. illustrates tools' specifications used in developing the chatbot.

TABLE I. TOOLS USED IN DEVELOPING THE CHATBOT.

IDE	Version	License	Selection Reason
Android Studio	2021.1.1	Open Source	The official ide for Google's Android OS
PyCharm	2022.1	Community Edition	To code and test python libraries
SQLite	3.25.0	Public domain	DB is a light single file that can be sent over any network

The term ‘Materials’ refers to the languages, libraries and plugins that have been used in building the proposed chatbot. Two languages have been used in building chatbots, i.e., Java and Python. A library is a collection of methods, algorithms, or documents that software developers use to reuse code written by other programmers. Table II. illustrates libraries and the plugin used in building the chatbot.

TABLE II. LIST OF LIBRARIES AND PLUGINS.

Library	Language	Version
stanford-corenlp-4.4.0	Java	4.4.0
stanford-corenlp-4.4.0-models	Java	4.4.0
ejml-core-0.39	Java	0.39
ejml-ddense-0.39	Java	0.39
org.jsoup	Java	1.14.3
ChatterBot	Python	1.0.4
NLTK	Python	3.7
Numpy	Python	1.22.3
Networkx	Python	2.8
Mathparse	Python	0.1.2
python-dateutil	Python	2.7.5
Sqlalchemy	Python	1.2.19
Pytz	Python	2022.1
Cython	Python	0.29.28
Spacy	Python	3.2.3
'Chaquopy'	Plugin	15.0

VII. SELECTION AND EVALUATION OF IMPLEMENTED MODELS

The study compares various AI techniques and tools, including Vader, TextBlob, RF, LR, CNN, and Stanford CoreNLP, based on parameters like training time, prediction accuracy, and usability level. The data was sourced from the IMDB website. The mentioned dataset has 50K movie reviews for natural language processing. The data was originally divided into 50/50 for training and testing respectively. To increase the model accuracy, the two sets were merged and split into 80/20 for training and testing. The data contains only two columns; one for the review and the other is for the sentiment. The results of the comparison between the models are presented in Table III.

TABLE III. COMPARING MODELS FOR SENTIMENT ANALYSIS.

Model	Concept	Training Time	Prediction Accuracy Small data	Prediction Accuracy Large data	Usability
Vader	rule-based	0	96%	71%	very simple
TextBlob	naive bayes classifier	30 mins	85%	63%	moderate
TextBlob	decision tree classifier	20 mins	61%	91%	moderate
RF	many decision trees		95%	82%	simple
LR	probability	≈ 4hrs	83%	73%	simple
CNN	multiple layers of different models and concepts	≈ 9hrs	97%	89%	hard
Stanford CoreNLP	rule-based, probabilistic ml, and deep learning components	≈ 5hrs	91%	98%	very simple

The Stanford CoreNLP framework was chosen for sentiment analysis in the chatbot due to its simplicity and ability to perform tasks like tokenization and speech tagging. Two other techniques, abstractive text summarization and web scraping, were also chosen for their popularity.

A. *Implementing Stanford CoreNLP framework for the chatbot*

CoreNLP processes the user input: In chat view when chatbot asks user ‘how are you?’, then the chatbot analyses user’s answer and scores it between (0,4). Fig. 1 illustrates the range of scores for the SA.

Figure 1 Sentiment Analysis Scores Range

Code used to perform the sentiment analysis:

```
public static int getPrediction(String userMsg) {
    nlpPipeline.init();
    return nlpPipeline.estimatingSentiment(userMsg);
}
```

```
public class nlpPipeline {
    static StanfordCoreNLP pipeline;

    public static void init() {
        Properties props = new Properties();
        props.setProperty("annotators", "tokenize, ssplit, parse, sentiment");
        pipeline = new StanfordCoreNLP(props);
    }

    public static int estimatingSentiment(String text) {
        int sentimentInt = 2;//defaults to neutral sentiment
        Annotation annotation = pipeline.process(text);
        for (CoreMap sentence : annotation.get(CoreAnnotations.SentencesAnnotation.class)) {
            Tree tree = sentence.get(SentimentCoreAnnotations.SentimentAnnotatedTree.class);
            sentimentInt = RNNCoreAnnotations.getPredictedClass(tree);
        }
        return sentimentInt;
```

In journaling view where user can write his random thoughts. After user saves his writings, the chatbot analyses user's writings, scores it between (0,4), waits for the user's preferred time to talk, and offers the user to discuss his thoughts.

B. *Implementing abstractive text summarization*

The chatbot summarizes and displays a previous conversation using Cosine Similarity and the TextRank model when the user chooses to resume a previous chat. Cosine similarity measures the similarity among user sentences by measuring the cosine of the angle between vectors, with an angle equal to 0 indicating similarity.

TextRank is an unsupervised graph-based text processing model that aids in identifying the most relevant sentences and keywords in text.

Code used to perform abstractive text summarization:

```
def summarize(text, per):
    try:
        nlp = spacy.load('en_core_web_sm')
        doc = nlp(text)
        tokens = [token.text for token in doc]
        word_frequencies = {}
        for word in doc:
            if word.text.lower() not in list(STOP_WORDS):
                if word.text.lower() not in punctuation:
                    if word.text not in word_frequencies.keys():
                        word_frequencies[word.text] = 1
                    else:
                        word_frequencies[word.text] += 1
        if len(word_frequencies.values()):
            max_frequency = max(word_frequencies.values())
            for word in word_frequencies.keys():
                word_frequencies[word] = word_frequencies[word] / max_frequency
            sentence_tokens = [sent for sent in doc.sents]
            sentence_scores = {}
            for sent in sentence_tokens:
                for word in sent:
                    ...
                    else:
                        sentence_scores[sent] += word_frequencies[word.text.lower()]
            select_length = int(len(sentence_tokens) * per)
            summary = nlargest(select_length, sentence_scores, key=sentence_scores.get)
            final_summary = [word.text for word in summary]
            summary = ''.join(final_summary)
        else:
            summary = "Lost Internet Connection!"
    except Exception:
        summary = "Lost Internet Connection!"
    return summary
```

C. *Implementing Web Scraping Technique*

Jsoup is an open-source Java library that parses, extracts, and manipulates data from HTML documents using HTML5 DOM methods and CSS selectors, often used as a last resort in chatbot scenarios. Code used to perform web scraping:

```
def parseWebToText(url):
    try:
        user_agent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36"
        config = Config()
        config.browser_user_agent = user_agent
        article = Article(url,config=config)
        article.download()
        article.parse()
    except Exception as e:
        return "Lost Internet Connection!"
    return summarize(article.text, 0.5)
```

D. *Ethics*

[23] outlines several ethics codes for researchers to follow to conduct ethical research, outlining rules for ethical behavior towards participants before, during, and after the study. This study adhered to four principles to ensure participant information and comfort. The first principle involved informing participants about the study and its terms, confirming their consent through an informed consent document, and allowing them to discontinue at any time. The second principle ensured anonymity and confidentiality of collected data, ensuring it was used solely for the study's purpose and not shared with others. The study's principles were crucial in ensuring participant satisfaction and participation.

VIII. CHATBOT IMPLEMENTATION

This section defines the concept and implementation of the proposed chatbot, lists the different chatbot views, explains possible workflow scenarios, and what are the possible forks and ends for each scenario. The chatbot concept outlines the problem, proposed solution, targeted audience, and content. A feasibility study is conducted to determine feasibility, followed by determining the target audience and content, forming the overall chatbot concept.

The chatbot aims to address daily stressful events by implementing multiple AI techniques. Based on research, it's best to build a chatbot targeting individuals aged 16 and above, who can speak English, as mobile phones are ubiquitous.

A. *Chatbot Interface Design*

The interface was designed in a proficient, clear, and uncomplicated way. The proposed chatbot has the following views: Authentication (register new users and login subscribed users), Survey, Home, Chat with AI, Journaling, Recommendations, Features, and Dashboard Views.

The chatbot uses a revised survey to assess a user's psychological state. The survey is divided into five sections: physical, sleep, behavior, emotion, and results. The user is offered to take the survey at the beginning of their interaction. The survey is aiming to identify the user's weak points and stressors. Fig. 2 to Fig. 3 illustrate some survey views.

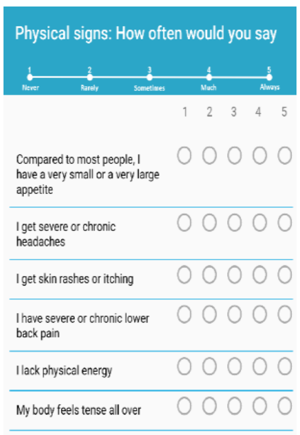

Figure 2 Physical Survey

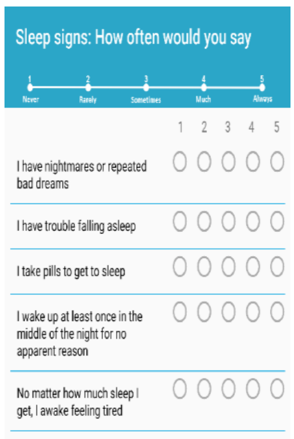

Figure 3 Sleep Survey

provide feedback, and a notification service view for users to select a preferred time for communication. Fig. 6 to Fig.

Chat view as shown in Fig.4 allows user-chatbot interaction with intelligent techniques like sentiment analysis, text summarization, and web scraping. Home view as shown in Fig. 5 redirects users after logging in, while journaling view allows users to write and leave without interaction.

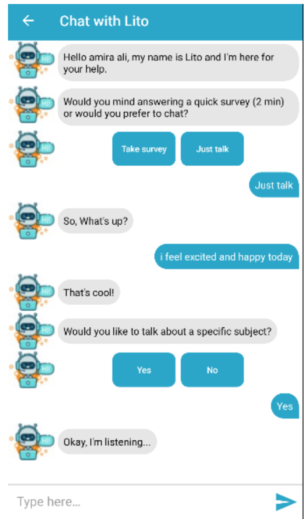

Figure 4. Chat View

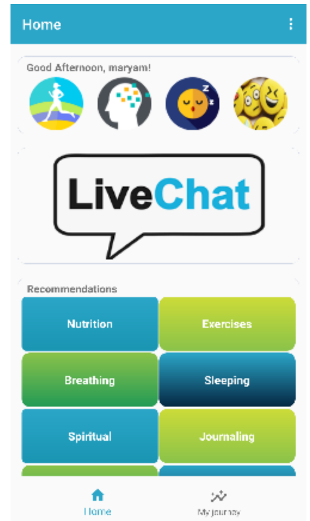

Figure 5. Home View

The chatbot provides eight recommendations from specialized medical and nutritional websites. These include journaling, nutrition, self-esteem, breathing, spiritual, sport, sleeping, and coping with changes. Journaling explains journaling benefits and links to it. Nutrition lists stress-busting foods, self-esteem defines causes and improvement, breathing exercises, spiritual resources, sports exercises, sleep tactics, and coping with changes lists common stress symptoms and ways to deal with life changes. It also offers three additional features: a dashboard view to view interaction history, a feedback view for users to rate and 11 Illustrate the Recommendations, Features, and dashboard Views.

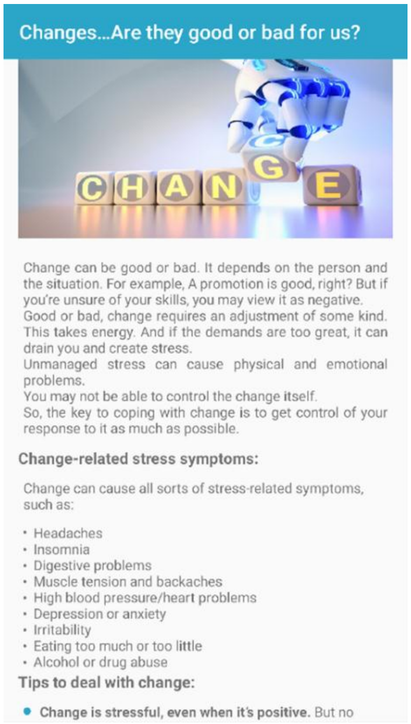

Figure 6 Coping with Changes

Figure 7 Nutrition View

Figure 8 Spiritual View

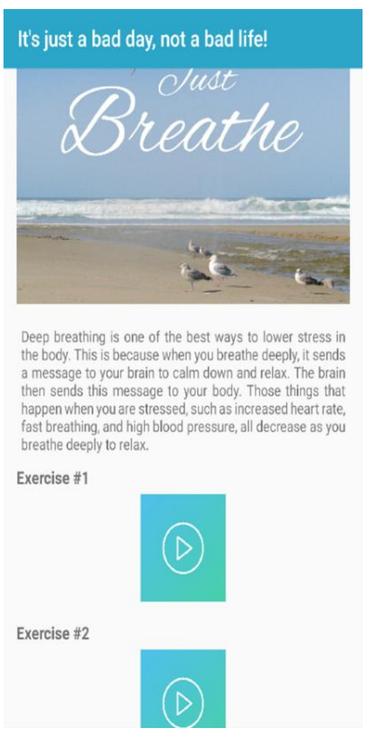

Figure 9 Breathing View

Figure 10 Sport View

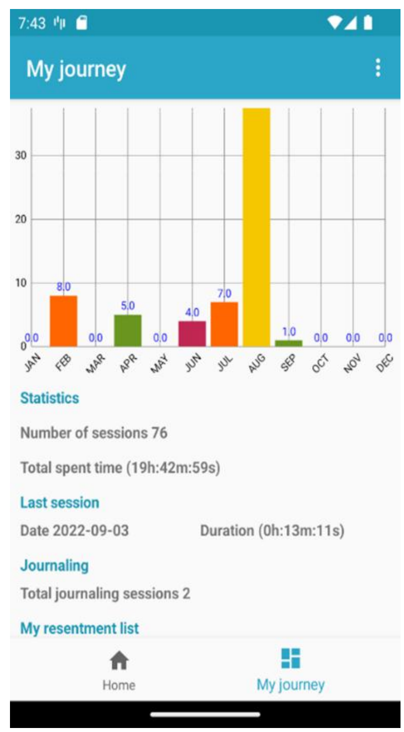

Figure 11 Dashboard View

B. Chatbot Process Flow

The chatbot process flow consists of four scenarios, each based on whether the user has taken a survey or chatted with the chatbot before. The chatbot offers different options to the user, starting with greeting and ending with solving their problem. Scenarios 1 and 2 offer two options for first-time users, while Scenarios 3 and 4 offer three options for users who have previously chatted with the chatbot. All three intelligent techniques, sentiment analysis, web scraping, and abstractive text summarization, are implemented in each scenario. Figure 12 and Figure 13 illustrate the process flow of the different scenarios.

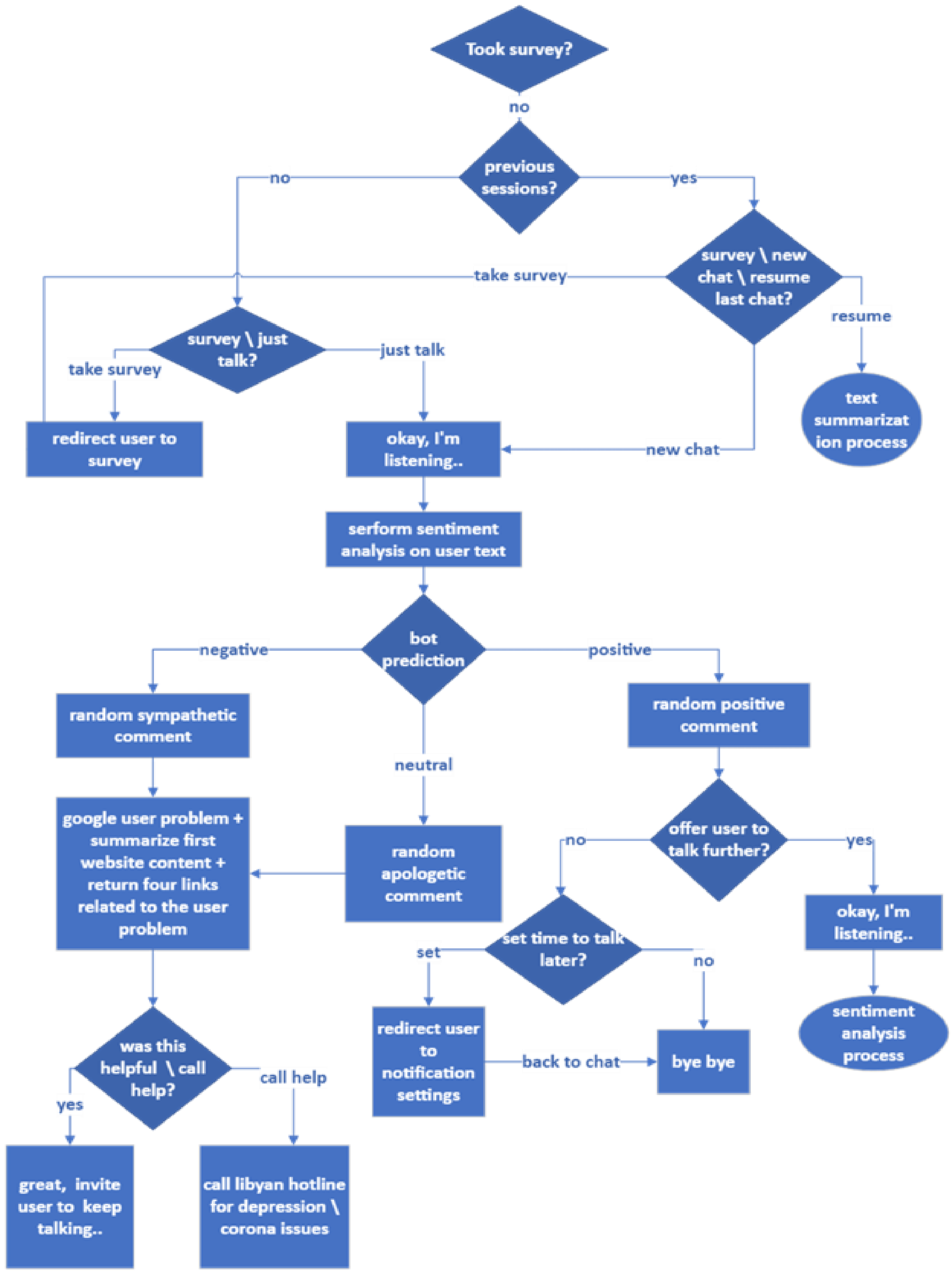

Figure 12 Scenario 1 & Scenario 2

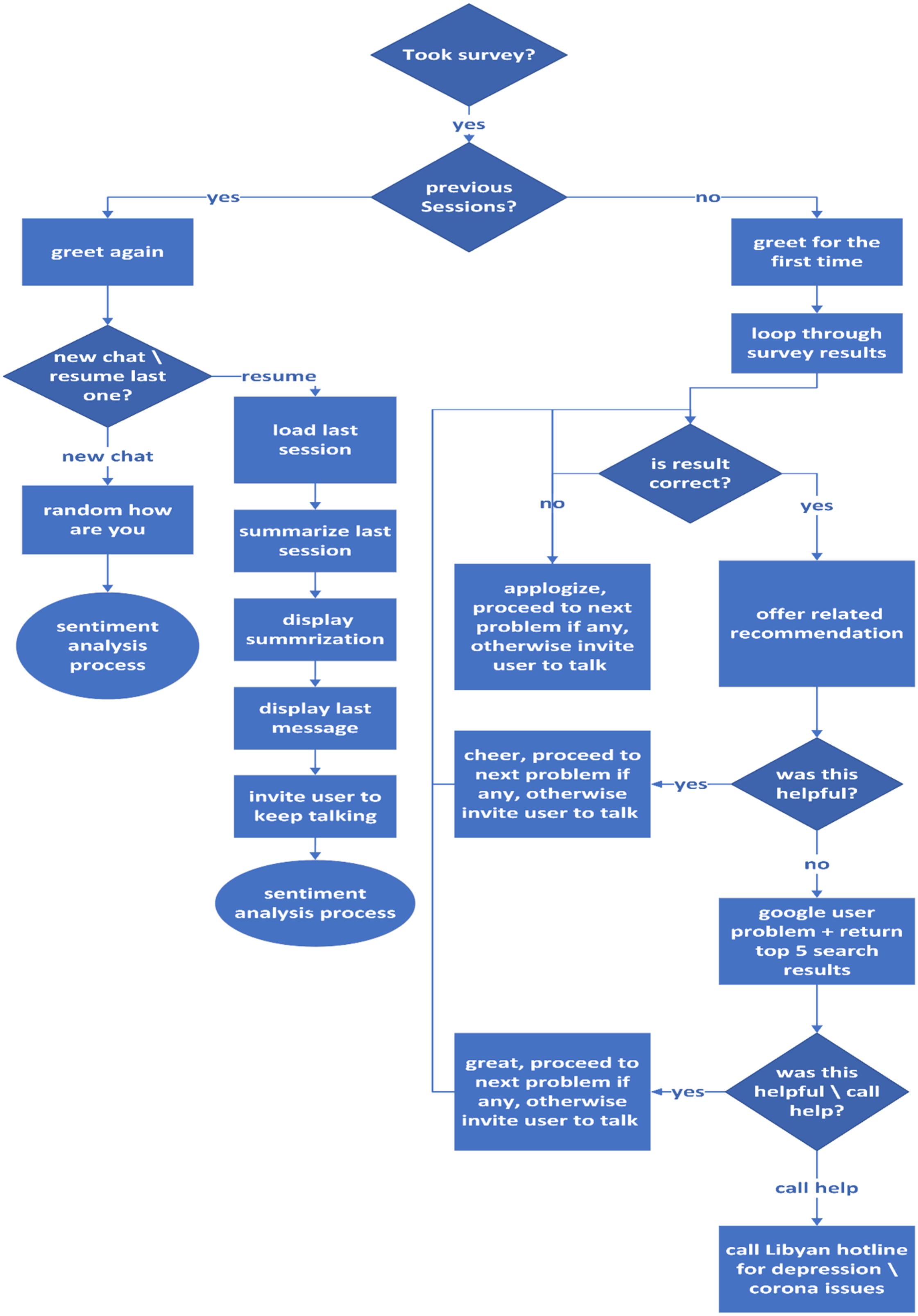

Figure 13 Scenario 3 & Scenario 4

IX. EVALUATION AND DISCUSSION

This section evaluates user feedback on the design, features, control level, and ability of the chatbot to refer to the past. The aim is to reduce stress among users, and a group of fifteen people was selected to test the chatbot. The test group was selected based on age, English proficiency, and avoiding bias. Two sections of questions were distributed to the group, focusing on design, features, control level, and past references. Feedback was collected through email, social media, and in-person, and analyzed in an Excel file.

The feedback included negative and neutral points that needed to be addressed. Measures included focusing on chatbot design, analyzing setting time feature code, showing alert messages before note deletion, adding terms and conditions, stating data is saved on mobile phones, and requiring internet connection for Google user questions. A dashboard view was added to summarize user interaction and improvement over time, and a mobile internet connection request was included for first-time installations. The accuracy of artificial intelligence methods depends on the data quality and size. Enhancing chatbots requires continuous work and commitment. Fig. 14 and Fig 15 show user feedback on design, and features, respectively.

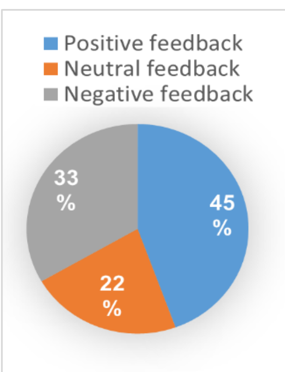

Figure 14 Sentiment Classification of User Feedback on Design

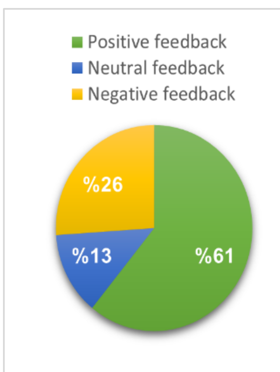

Figure 15 Sentiment Classification of User Feedback on Features

X. CONCLUSIONS

A chatbot was developed for stress management using various intelligent models and techniques. A prototype was created and tested using Java and Android Studio IDE. The chatbot was distributed to fifteen participants for evaluation. User feedback showed satisfaction with the design and features, with some highlighting the grouping of stress-reducing recommendations and journaling features. However, limitations were noted, such as Arabic language support and audio capability. These limitations were addressed as recommendations for future work. Although not perfect, the chatbot's good foundation allows for future improvements by future AI researchers.

XI. FUTURE WORK

The chatbot's evaluation revealed recommendations for improvement, including Arabic language support, voice interaction, and automatic location hotline number retrieval. Stanford-CoreNLP supports Arabic, eliminating the need for additional libraries or intelligent models.

REFERENCES

[1] Jackson RG, Patel R, Jayatilleke N, Kolliakou A, Ball M, Gorrell G, Roberts A, Dobson RJ and Stewart R, "Natural language processing to extract symptoms of severe mental illness from clinical text: the Clinical Record Interactive Search Comprehensive Data Extraction (CRIS-CODE) project," *BMJ Open*, p. 17;7(1):e012012, 17 Jan 2017.

[2] Cook BL, Progovac AM, Chen P, Mullin B, Hou S and Baca-Garcia E., "Novel Use of Natural Language Processing (NLP) to Predict Suicidal Ideation and Psychiatric Symptoms in a Text-Based Mental Health Intervention in Madrid," *Comput Math Methods Med.*, p. 2016: 8708434, 26 Sep 2016.

[3] M. M. Aldarwish and H. F. Ahmad, "Predicting Depression Levels Using Social Media Posts.," in *IEEE 13th International Symposium on Autonomous Decentralized System (ISADS)*, Saudi Arabia, 2017.

[4] K. Fitzpatrick, A. Darcy and M. Vierhile , "Delivering Cognitive Behavior Therapy to Young Adults With Symptoms of Depression and Anxiety Using a Fully Automated Conversational Agent (Woebot): A Randomized Controlled Trial," *JMIR Mental Health*, p. 4(2): e19., 6 June 2017.

[5] C. Beatty, T. Malik, M. Saha and C. Sinha, "Evaluating the Therapeutic Alliance With a Free-Text CBT Conversational Agent (Wysa): A Mixed-Methods Study," *Frontiers in Digital Health*, 2022.

[6] D. Azevedo, A. Legay and S. Kieffer, "User Reception of Babylon Health's Chatbot," in *17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2022.

[7] G. Cameron, D. Cameron, G. Megaw, R. Bond, M. Mulvenna, S. O'Neill, C. Armour and M. McTear, "Assessing the Usability of a Chatbot for Mental Health Care," in *International Conference on Internet Science*, France, 2018.

[8] A. N. Vaidyam, H. Wisniewski, J. D. Halamka, M. S. Kashavan and J. B. Torous, "Chatbots and Conversational Agents in Mental Health: A Review of the Psychiatric Landscape," *Canadian Journal of Psychiatry*, pp. 64(7):456-464, 2019.

[9] Abd-Alrazaq A, Alajlani M, Ali N, Denecke K, Bewick B and Househ M, "Perceptions and Opinions of Patients About Mental Health," *JOURNAL OF MEDICAL INTERNET RESEARCH*, p. 23(1):e17828, 2021.

[10] K. H. Ly, A.-M. Ly and G. Andersson, "A fully automated conversational agent for promoting mental well-being: A," *Internet Interventions*, pp. 39-46, 2017.

[11] B. TW, M. SE, B. W. Jack, M. K. Paasche-Orlow, L. M. Pfeifer and J. Odonnell, "Response to a Relational Agent by Hospital Patients with Depressive Symptoms," *Interacting with Computers,* pp. 22(4):289-298, 2010.

[12] J. Hurwitz and D. Kirsch, Machine Learning For Dummies, John Wiley & Sons, Inc., 2018, p. 16.

[13] X. L. Y. D. L. B. W. L. Wenxuan Zhang, "A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges," 2022.

[14] P. L. T. Say Hong Lye, "Customer Intent Prediction using SentimentAnalysis Techniques," in *The 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 2021.

[15] J. Kaur and J. R. Saini, "Emotion Detection and Sentiment Analysis in Text Corpus: A Differential Study with Informal and Formal Writing Styles," *International Journal of Computer Applications* , 2014.

[16] D. M. El-Din, "A Survey on Sentiment Analysis Challenges," *Journal of King Saud University - Engineering Sci-ences,* 2016.

[17] S. Rani, . N. S. Gil and . P. Guli, "Survey of Tools and Techniques for SentimentAnalysis of Social Networking Data," *International Journal of Advanced Computer Science and Applications,* 2021.

[18] J. Madhuri and R. G. Kumar, "Extractive Text Summarization Using Sentence Ranking," in *2019 International Conference on Data Science and Communication (IconDSC)*, 2019.

[19] D. Patel, N. S. Shah and V. Hole, "Abstractive Text Summarization on Google Search Results," in *2020 4th International Conference on Intelligent Computing and Control Systems*, 2020.

[20] V. Singrodia, A. Mitra and S. Paul, "A Review on Web Scrapping and its Applications," in *2019 International Conference on Computer Communication and Informatics (ICCCI)*, 2019.

[21] S. Flores-Ruiz, R. Perez-Castillo, C. Domann and S. Puica, "Mainframe Migration Based on Screen Scraping," in *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2018.

[22] S. Mathur and D. M. Thomas, "Data Analysis by Web Scraping using Python," in *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2019.

[23] Swedish Research Council, "Ethics in research," 2019. [Online]. Available: https://www.vr.se/english/mandates/ethics/ethics-in-research.html. [Accessed 2022].