BeeKnote: Voice Chatbot Assistant for the Beekeepers

Metidji Sid Ahmed
Université Paris-Panthéon-Assas
Efrei Research Lab
Paris, France
sid-ahmed.metidji@efrei.fr
0009-0006-1757-7422

Bougueroua Lamine, Huet Jean-Charles
Université Paris-Panthéon-Assas
Efrei Research Lab
Paris, France
{lamine.bougueroua, jean-charles.huet}@efrei.fr
{0000-0002-5322-8231, 0000-0002-0859-4734}

Mokeddem Hakim
Higher School of Computer Science
ESI Algiers
Algiers, Algeria
h_mokeddem@esi.dz

***Abstract*—With the continuous advancement of Artificial Intelligence technologies, chatbots assistants become very popular lately due to their unique characteristics in improving human-computer communication thanks to the Natural Language techniques to process and understand natural human language, and this makes the chatbots assistants show a positive impact in enhancing the user experience for the customers and increasing the business profit for the stakeholders in almost every field, including the beekeeping. In fact, due to the different encountered challenges by the beekeepers on their work, the need for chatbots to assist them was clearly expressed in many surveys. In this paper, firstly, a state of the art is presented about existing solutions as chatbot assistants in the agriculture and beekeeping field before presenting "BeeKnote". Our chatbot assistant aims to assist the beekeepers in their work by extracting relevant information from the vocal commands. The "BeeKnote" specificity is in their flexible architecture that addresses the limitations of the related work and provides a new approach to assist the beekeepers by offering them a hands-free experience.**

***Keywords*—Vocal assistance; Natural language processing; Natural language understanding; Beekeeping; Smart agriculture.**

I. INTRODUCTION

Beekeeping is the art of cultivating bees in order to obtain from this industry the maximum yield with the minimum expense by providing the hive with a perfect environment that will ensure the maximum productivity of the bees that are inside [1]. Besides being a profitable business, beekeeping is considered a crucial field that is essential to the balance of ecosystems and the assurance of food security. Indeed, bees are the key actors of biodiversity, as around 85% of flowers are pollinated by them, and the success of the different agriculture branches is dependent on the success of the pollination process by the honey bees in different countries [2].

However, nature takes its course, and our influence disrupts normal natural processes, altering the balance of natural perfection, which has led to the decline of honey production all over the world and make the production of high-quality honey a more complex goal to be achieved. For example, in France, despite being considered one of the big producers of honey, the volumes of imported honey thus increased by 36% between 2010 and 2020 (6% in 2020) [3].

Consequently, providing an assistant to guide the beekeepers on their work and increase their productivity is important, and chatbots can be an appropriate way to help the beekeepers due to their nature of being user-friendly and convenient in the beekeeping environment by facilitating any average user to interact with any device, anywhere and at any time, without the need of special skills or training by involving a variety of written or oral natural communication forms in order to simulate conversations.

In fact and according to a survey carried out by "ITSAP" in 415 beekeepers [4], voice inputs(77%) and data storage(72%) are already wanted features for the beekeepers.

Additionally, "the Preservation of Bee colonies" and "minimizing the beehives inspections" are among the top 3 most important areas for beekeepers, and it's confirmed that a "Digital notes" app is considered a valid solution to fulfill these needs [2].

However, the usage of innovative technologies that may help the stakeholders and the workers in the agriculture field (including the beekeepers) is still limited [5]. "ITSAP" survey shows that "practicality in the field" (around 15%), "the complexity of the developed solutions" (around 15%), and "the data loss or corruption" (around 10%) are considered the major factors that prevent the beekeepers from cooperating with the existed platforms in the beekeeping field [4].

This paper is organized as follows, Section 2 introduces a comparative survey of the existing chatbots assistant in the agriculture and beekeeping field, the Section 3 will focus on presenting "BeeKnote" our newly developed chatbot assistant for beekeepers that aims mainly to cover the limitation of the mentioned systems by providing a user-friendly, accurate and natural language-based assistant for the beekeeping business, Section 4 presents a discussion about the obtained results from testing the system components and Section 5 provides the summary of the work and future plans to improve the beekeeper's experience while using our system.

II. RELATED WORK

Chatbot assistant systems are not a new area of research, they are widely used in different fields, such as e-commerce, educational support, and customer service [6]. However, according to our research, there are few and limited efforts to build chatbot assistants in the field of agriculture in general and in the field of beekeeping in particular. This paper will focus on mentioning the founded chatbot assistants applied in

 ISBN: 978-1-68558-051-3

the field of agriculture in general, or exclusively in the field of beekeeping, presenting their offered functionalities and the implementation technologies used.

- Iot-AgriBot [5]: they developed a "Facebook Messenger" chatbot that took any natural language textual/vocal input entered by the user and passed it to “DigitalFlow”: the Google AI platform, which will establish a connection with the concerned IoT agricultural device, in order to provide monitoring information (such as the temperature, and taking the control of the equipment)
- Chatbee [6]: this system consists of answering pre-defined questions related to beekeeping production, it is developed using the closed source code Google AI platform: “Degitalflow“ in the form of a "Telegram" bot.
- API digital bot [7]: this work aims to develop a system to remotely monitor the connected hives with embedded IOT systems, providing real-time information to the beekeeper. The system was implemented in the form of a "Telegram" bot and the communication with the system is done through predefined commands (for example, the command "/start" is used to perform the authentication and initialize the communication with the chatbot, and "/ultima" is used to get the latest records from the connected beehives).
- LINE [8]: LINE is an interactive chatbot in the form of an Android mobile application where the user can ask about the state of a specific plant (temperature, humidity, soil moisture, etc.) by using IOT devices as a sensors that transmit data related to air humidity, soil moisture, light conditions, and ambient temperature of the connected plant. Communication with this mobile app is possible with natural human language thanks to using Natural Language Processing (NLP) techniques to perform the Natural Language Understanding (NLU) of the provided question by the farmer.
- Zhang et al. [9] designed a FAQ chatbot in the agriculture field, the predefined template-based questions will be answered using AIML (Artificial Intelligence Markup Language) technique while the service-based questions will be answered using Latent Semantic Analysis (LSA) technique, these two techniques are considered as predecessors to the modern NLP techniques.
- Symeonaki et al. [10] proposed an agent chatbot system that is capable to monitor and control a group of IOT devices composed of boiler, fan, and lighting controllers installed in the greenhouse in order to control the environment inside the greenhouses using a mobile application with natural human language thanks to the integration of NLP and NLU techniques, the system offers also the control of a plant water demand and consumption
- Beeking [11] (cited in [2]): it is a mobile app and portal that was developed by the SIA BeeTech Services, this app is used by more than 400 beekeepers in Latvia and almost 500 beekeepers abroad. Among the several features offered by this portal, there is the vocal recording option. Regrettably, there isn’t an integrated intelligent assistant that can understand what the beekeeper has said.
- AGRI-QAS [12]: this Question-Answering system is developed to be a domain-specific question-answering system targeting the agriculture domain. Its purpose is to help farmers get information and resolve their queries related to agriculture in the form of FACTOID questions, such as ’which’, ’what’, ’who’, and where. It uses a Question Classifier (diseases, pests, weeds, crops, etc.) to detect the question subject and domain-specific named entity recognizer to find out the best answer. For example, for a question seeking the name of a crop, a Named Entity Recognizer (NER) is used to find candidate crop names from the ranked documents.
- ADANS [13]: this system presents a semantic question-answer system on agriculture developed using a combination of NLP and semantic web technologies, in contrast to the previously cited systems that used database-based data to do the learning process, this system is fed by an ontology-based data, which has a more complicated structure than relational databases, which is a time-consuming task.

Table I summarizes the different main characteristics of each mentioned system and positions our system “BeeKnote” among them.

Unfortunately, and as it can be concluded from Table I, according to our research, we have not found any paper that presents or mentions the existence of an AI-based vocal chatbot assistant in the field of beekeeping field that integrates the NLP/NLU technologies to understand the beekeeper’s vocal and taking in consideration the specific working conditions (wearing the beekeeper’s suit, the possibility of not being able to hold the phone and use the hands while manipulating the app, etc.) by offering them a hands-free experience thanks to real-time hot-word detection.

In order to address these limitations, we propose the following system design and implementation.

III. Proposed Approach

“BeeKnote” aims to address the limitations of the other agent-based chatbot systems in the beekeeping domain. The system can be accessed by installing the “BeeKnote” Android App on a mobile device with internet access provided in order to do the cloud database storage and communication with the different developed and deployed APIs and AI models.

A. BeeKnote main functionalities

After being authenticated, “BeeKnote” offers mainly three principal features:

1) **Recording and Understanding the beekeepers’ vocal commands:** all the commands to be treated aim to store information related to the beehive being inspected at that time (which beehive?, which time? which value? which metric?), these commands fall into four areas: weight-related commands, humidity-related commands, internal temperature-related commands, and external

TABLE I. COMPARAISON TABLE

	Iot-AgriBot [5]	Chatbee [6]	API digital bot [7]	LINE [8]	Zhang et al. [9]	Symeonaki et al. [10]	Beeking [11]	AGRI-QAS [12]	ADANS [13]	**BeeKnote**
[2ex] Human language communication	✓	✓		✓	✓	✓	✓	✓	✓	✓
Mobile devices support	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Database writing	✓					✓	✓			✓
Beekeeping-specific chatbot		✓	✓				✓			✓
NLP/NLU-based learning	✓	✓		✓	✓	✓		✓	✓	✓
Independent from IOT devices		✓			✓	✓	✓	✓	✓	✓
Independent from other messaging applications				✓		✓	✓	✓	✓	✓
Interaction with the user	✓	✓	✓	✓	✓	✓		✓	✓	✓
Vocal inputs support	✓						✓			✓
Vocal inputs understanding	✓									✓
Vocal hot-word detection										✓

temperature-related command, after achieving the NLU of the command, it will be stored locally on-device waiting for the manual validation/rejection by the beekeeper, the NLU part will be well detailed below.

2) **Cloud Storage of the vocal intent after the validation:** once the command is validated by the user, the application will send an HTTP request to an API, which will store the command content (beehive name, value, metrics, etc.) in the cloud in order to assure the data preservation.
3) **Rejection of the command:** we see that offering the user a manual rejection of the command is important for the precaution and handling of the edge cases (misunderstanding of the command, saying false information, inappropriate record, etc.).

Additionally, and for the sake of improving the beekeeper's experience while using our Android app and taking into consideration their particular condition while working, a hand-free experience is provided where the user can communicate with the app by the pronunciation of predefined keywords.

B. BeeKnote global system architecture:

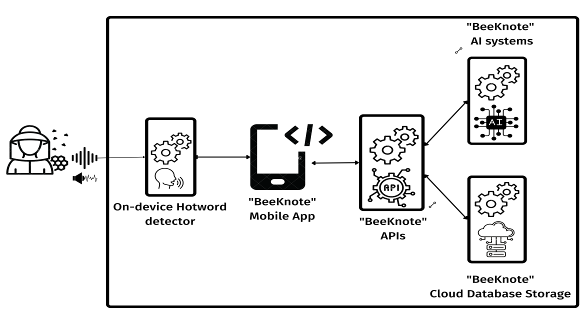

Figure 1. BeeKnote global system architecture

As shown in Figure 1 and in order to fulfill our three major features, the system is going to need a Mobile App with an integrated hot-word detector to assure the hand-free experience by toggling the recording, the validation, and the rejection by saying predefined keywords. Additionally, the mobile app will be connected to the developed APIs, which are going to be the portal to communicate with our AI systems to achieve the NLU and the cloud database storage to store the commands and their interpretations on the cloud in a foolproof manner.

C. BeeKnote System Components

We propose a complex system composed of:

1) Automatic Speech Recognition (ASR) System: "Speech Recognition" is the process of converting a speech signal to a sequence of words by means algorithm implemented as a computer program [14]. ASR is an important technology to enable and improve human-computer interactions [15].

The accuracy of speech recognition models is measured by the “Word Error Rate” (WER), which is defined as: “The number of errors divided by the total words”. Fadel et al. [16] presented a table summarizing the evaluation of French speech recognizer systems by the WER metric. All the cited systems give a WER bigger than 9% (from 9.6% to 47.41%).

In “BeeKnote”, “Whisper” [17] is used, it is an open-source multilingual and multitask deep learning speech recognition system provided by the “Open AI” community launched officially in September 2022.

"Whisper" gives a WER equal to 8.3% in the "Fleurs" database for the French language [18], which proves his performance compared to the alternative ones.

2) Hot-word detection system: Hot-word detection is the application's constant waiting mechanism for a specific word(s) [19], when the predefined word is heard, the application recognizes it and activates a certain predefined process [20]

For “BeeKnote”, predefined words are needed to launch/stop the audio recording, the validation of the command, and the command rejections to assure a hands-free experience for the beekeepers, and for that, “Porcupine” is used [21].

"Porcupine" is a lightweight and production-ready hot-word detector that shows the biggest accuracy (91%) in "work word benchmark" compared to two famous hot-word detectors: “Snowboy” (68%) and "PocketSphinx" (52%) [21].

3) Text classification system: Text classification is the task of classifying a document under a predefined category. If d_i is a document of the entire set of documents D and $\{c_1, c_2, \ldots, c_n\}$ is the set of all the categories, then the role of the text classifier is to assign one category c_i for each document d_i [22].

We built a custom multi-classification NLP model that takes as input a text and classifies it according to its intent among four classes: "Weight", "Humidity", "Internal temperature", and "External temperature". For the training, an Excel sheet containing 75 commands and their corresponding classes filled manually by beekeepers is used.

After doing a pipeline of text preprocessing consisting of tokenization, sequence padding, and finally the Stanford Glove embedding, we passed the data into the model to do the training process, and in order to treat the whole sentence, which is composed of a set of words at once and in a single iteration, the model to train must be a sort of a Recurrent Neural Network (RNN) that evaluates the current input as well as what it has learned from past inputs thanks to its internal memory [24].

And for that, deep multi-classification bidirectional Global Recurrent Unit (GRU) [23] neural architecture was built and trained to do the training and the prediction of the commands topic.

The reason behind using GRU instead of a typical RNN or the more-sophisticated one: Long Short term memory (LSTM), is that the GRUs train faster and seem to perform better for small amounts of data, but LSTMs have greater computational complexity [24].

4) Named Entity Recognizer (NER) system: NER is the problem of locating and categorizing important nouns and proper nouns in a text. For example, names of persons, organizations, and locations are typically important. NER plays an important role in information extraction [25]. For “BeeKnote”, a NER system is crucial to do the information extraction from the beekeeper command, such as beehives names, values, and metrics to store or even the qualitative description, such as “heavy”, “hot”, “too much vapor”, etc. There are several pre-trained models, most are for the English language and they are trained for general-use entities (like the locations and the organizations). In our case, we defined 11 domain-specific entities that we need to extract from the beekeeper commands, Figure 2 shows the weight-related predefined entities as an example.

There were some efforts to design from scratch a deep RNN model that would train well in our NER data but unfortunately, it doesn’t show promising results, and for this reason, we used CamemBERT to train our custom NER model for the beekeeping domain in the French Language.

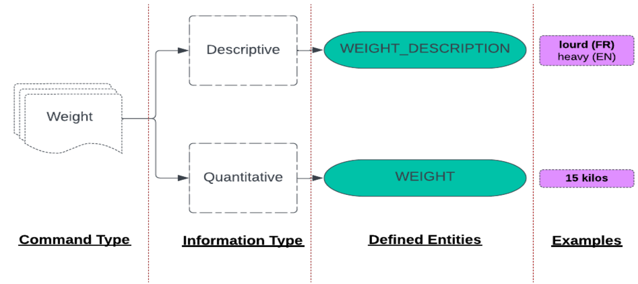

Figure 2. Predefined entities for the weight information

"CamemBERT" is based on RoBERTa, which itself is based on BERT. BERT is a multi-layer bidirectional Transformer encoder, it comes in two sizes: the "BERT-Base" and the "BERT-Large". The "BERT-Base" architecture is 3 times smaller, so it’s faster and easier to use while BERT-Large achieves generally better accuracy. "CamemBERT" is used because it achieves higher F1 scores than the traditional NER architectures [26].

To train our NER, “SpaCy” [27] is used as a software library platform where "CamemBERT" was manipulated as a custom spaCy pipeline to train it in our data.

5) First REST API (Flask): this API plays the gateway role by providing an HTTP portal to communicate with the three previous AI systems (ASR, classification, and NER), “Flask” is used as a technology to implement this API since the AI models can be easily plugged, extended and deployed as a web application there and it’s a lightweight server [28].

6) Cloud database storage: Cloud storage is crucial in our system to assure data preservation for the beekeepers, “MongoDB” is used as a technology to achieve data preservation due to its simplicity and its flexibility with the data structures, which will assure an easy future extension to our data schemas [29].

7) Second REST API (NodeJS): this API is the portal to communicate with the system cloud database by providing an HTTP interface for that, “NodeJs” is used as a technology since it’s well-known for being fast and it allows us to explore a dynamic range of data in real-time easily [30].

8) The Android Application (BeeKnote): our developed Android application will allow the beekeepers to exploit easily our three major functionalities previously mentioned by offering them an easy-to-use interface to communicate with our intelligent chatbot.

D. BeeKnote Workflow System Architecture

Figures 3 and 4 explain the interconnection between the different system components in order to achieve the recording, the understanding, and the validation of the beekeeper’s vocal commands.

1) Once our Hotword device detector hears the keyword “Bee note”, the mobile app starts the audio recording.

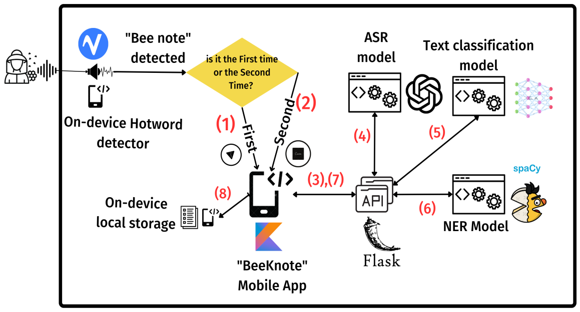

Figure 3. Audio Recording and Understanding Workflow

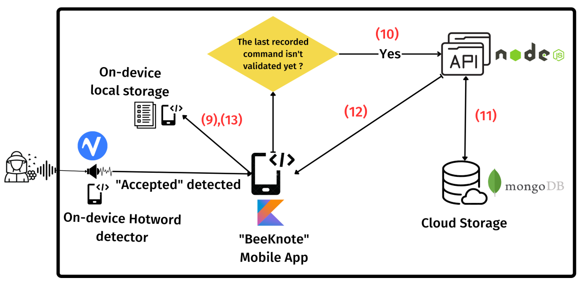

Figure 4. Command validation workflow

2) The record continues until the beekeeper says the same keyword again.
3) The audio will be recorded locally on the device in an MP3 file and it will be sent to the Flask API.
4) The API Transmits the .mp3 file to the ASR system, which will send back the generated textual transcription as a response.
5) The API transmits the generated transcription to our text classifier system, which will send back as a response the predicted class of the said command.
6) Then, the API transfers the transcription to our NER system, which will send back as a response an array of the detected entities and their types.
7) The Android app will receive a response from the API call in 3) the generated transcription, the predicted class, and the detected entities.
8) After confirming the harmoniousness between the predicted class and the detected entities, we store the transcription, the predicted class, and the detected entities in the local storage, waiting for the beekeeping validation.
9) Once the validation keyword has been said “Oui Bee Note”, the Android App searches inside its local storage for the latest recorded command.
10) If it’s not validated yet, it will be transmitted to the NodeJS API.
11) The NodeJS API sends the appropriate query to MongoDB to store the command interpretation on the cloud.
12) The Android App receives a response from the request sent in 10) a confirmation of the cloud storage with success.
13) The last recorded command status will be changed from "pending" to “validated”.

IV. Results and Discussion

In order to measure the accuracy of our developed system, we have built a test dataset composed of 26 vocals (8 for weight, 10 for temperature, and 8 for humidity) where we covered every desired pattern that our system is supposed to understand accurately. This dataset contains 56 entities to detect (15 weight-related, 25 temperature-related, and 16 humidity-related).

A. Unit Testing

We have done a unit accuracy test of each component independently before doing an integration test where we measure the accuracy of the whole system by connecting the whole system’s components.

1) ASR System results: as it’s shown in Table II, three versions of Whisper were used in our test database, while the three have varying percentages of accuracy, they have a convergent execution time, and that’s why we picked the ‘Large’ version for "BeeKnote", which has 76.92% of accuracy

TABLE II. ASR results

Version	Perfect Transcriptions	Avg. Exec (s)
Base	42.3%	**0.328**
Medium	61.53%	1.03
Large	**76.92%**	1.49

2) NER System results: Two versions of CamemBERT were used to train our own NER System and two metrics to measure the accuracy of these two versions were defined and used:

- Detection rate: it measures the ratio of the detected entities inside the commands transcription.
- Accuracy rate: it measures the ratio of the detected AND well-classified entities inside the commands transcription.

In this unit test, we suppose that the ASR system transmits the perfect transcription to our NER.

TABLE III. NER camemBERT-base VS camemBERT-large results

Type	Detection		Accuracy		Avg Exec (s)	
	Base	Large	Base	Large	Base	Large
Weight	100%	100%	**100%**	93%	**0.12**	0.32
Temperature	92%	**96%**	88%	**96%**	**0.10**	0.27
Humidity	**100%**	93%	81%	**87%**	**0.10**	0.27
Total	96%	96%	89%	**92%**	**0.10**	0.33

Table III shows that although camemBERT-large is thrice slower, it has shown just a little improvement in the accuracy rate (3% more accurate).

3) Text Classification system results: Table IV shows that our best classifier has an accuracy of around 70-75% for the three classes.

 ISBN: 978-1-68558-051-3

TABLE IV. TEXT CLASSIFICATION SYSTEM:

Type	Classified correctly
Weight	75%
Temperature	70%
Humidity	75%
Total	73.07%

B. Integration Testing

Since we noticed that our NER system performs so well that it doesn't need the classification system to assure its consistency, in addition to that, we want to improve the beekeeper experience by allowing him to say a mixture of entities that don't have to be on the same class (for example, allow the beekeeper to store information about the weight and the temperature of a certain hive in the same command), so the transcription generated by the ASR system will be passed only to the NER system in order to get at the end the different entities to store in the cloud.

In this test, we will pass the transcription generated by 'Whipser-large' to our NER systems "camemBERT-large" and "camemBERT-base".

The results in Table V shows that the integration of Whipser-large with our pre-trained camemBERT-large or camemBERT-base gives the same global accuracy (80.35%) with a slight increase in the detection rate for the camemBERT-large (+3.5%) even though it was shown that the camemBERT-large has better accuracy, this can be explained by the camemBERT-large's fault intolerance with the inaccurate transcriptions.

TABLE V. WHISPER-LARGE INTEGRATION WITH CAMEMBERT-BASE VS WHISPER-LARGE INTEGRATION WITH CAMEMBERT-LARGE RESULTS

Type	Detection		Accuracy		Avg Exec (s)	
	Base	Large	Base	Large	Base	Large
Weight	86%	80%	80%	80%	**+0.13**	+0.32
Temperature	92%	**92%**[1]	80%	**84%**	**+0.11**	+0.38
Humidity	**93%**	87%	**81%**	75%	**+0.10**	+0.52
Total	**91%**	87%	80%	80%	**+0.11**	+0.40

[1] There was a false positive detected entities here: 92% are true positive ones while there was additional 16% detected considered as false positive.

We conclude finally that the integration of "camemBERT-base" with "Whipser-large" is the best to avoid the false positive detected entities and to improve the overall system performance by reducing the response latency to assure a more fluid experience for the beekeepers.

V. CONCLUSION AND FUTURE WORK

This paper presents "BeeKnote", a voice chatbot assistant for beekeepers. This system is able to record and understand the beekeeper's vocal inputs to extract relevant mentioned information related to the beehive state. The system is in a microservices shape, which makes every component easily replaceable by an alternative one, this also assures the system flexibility and the ability to reuse it across different areas of business. The future version of "BeeKnote" may address the limitations of the current version by including an offline version where all the AI models are deployed on the device and the cloud storage of the content of the commands will be done once the mobile is connected to the internet, this would increase the system availability by assuring the chatbot-beekeeper interactions in areas with poor internet coverage. Additionally, and due to the lack of training data, the "Bee-Knote" current version handles semi-structured instructions, where the beekeeper's commands should have a similar format to the commands used in the learning dataset, otherwise, the beekeeper may face inaccurate processing of its command, so assuring that aspect of flexibility can be interesting also.

REFERENCES

[1] Warre, E. (2006) ["L'apiculture pour tous: L'apiculture facile et reproductive"] (Beekeeping for all: Easy and reproductive beekeeping). Revest-Saint-Martin: Coyote ed.Accessed: May 30, 2023. [Online]. Available: http://www.apiculture-warre.fr/.

[2] A. Zacepins, A. Kviesis, V. Komasilovs, V. Brusbardis, and J. Kronbergs, "Status of the Precision Beekeeping Development in Latvia," Rural Sustainability Research, vol. 45, no. 340, pp. 86–92, Aug. 2021, doi: 10.2478/plua-2021-0010.

[3] J.-C. Huet, L. Bougueroua, Y. Kriouile, K. Wegrzyn-Wolska, and C. Ancourt, "Digital Transformation of Beekeeping through the Use of a Decision Making Architecture," Applied Sciences, vol. 12, no. 21, p. 11179, Nov. 2022, doi: 10.3390/app122111179.

[4] ["Enquête «Les apiculteurs et le numérique» | Article | ITSAP."] (Survey "Beekeepers and digital technology" | Article | ITSAP.) https://itsap.asso.fr/articles/enquete-les-apiculteurs-et-le-numerique (accessed May 30, 2023).

[5] E. Symeonaki, K. Arvanitis, P. Papageorgas, and D. Piromalis, "AI-Based Chatbot System Integration to a Social Media Platform for Controlling IoT Devices in Smart Agriculture Facilities," 2021, pp. 193–209. doi: 10.1007/978-3-030-84156-0_10.

[6] A. A. de Oliveira Filho, "Chatbot as Rural Extensionist for Small Beekeepers," in Anais do XIV Encontro Unificado de Computação do Piauí (ENUCOMPI 2021), Sociedade Brasileira de Computação, Nov. 2021, pp. 200–206. doi: 10.5753/enucompi.2021.17772.

[7] D. Ferreira Mojaravscki, ["Apicultura Digital, a Transformação Tecnológica Da Apicultura,"](Digital Apiculture, the Technological Transformation of Apiculture,) in Agrárias: Pesquisa e Inovação nas Ciências que Alimentam o Mundo Vol.I, Editora Artemis, 2020, pp. 23–32. doi: 10.37572/EdArt_0643006203.

[8] R. Gunawan, I. Taufik, E. Mulyana, O. T. Kurahman, M. A. Ramdhani, and Mahmud, "Chatbot Application on Internet Of Things (IoT) to Support Smart Urban Agriculture," in 2019 IEEE 5th International Conference on Wireless and Telematics (ICWT), IEEE, Jul. 2019, pp. 1–6. doi: 10.1109/ICWT47785.2019.8978223.

[9] D. Zhang, X. Chen, Y. Zhang, and S. Qin, "Template-based Chatbot for Agriculture Related FAQs," Jul. 2021, Accessed: May 30, 2023. [Online]. Available: https://arxiv.org/abs/2107.12595v2

[10] E. Symeonaki, K. Arvanitis, D. Piromalis, and M. Papoutsidakis, "Conversational User Interface Integration in Controlling IoT Devices Applied to Smart Agriculture: Analysis of a Chatbot System Design," 2020, pp. 1071–1088. doi: 10.1007/978-3-030-29516-5_80.

[11] "BeeKing - your digital beekeeper app." https://beeking.eu/en/index.html (accessed May 30, 2023).

[12] S. Gaikwad, R. Asodekar, S. Gadia, and V. Z. Attar, "AGRI-QAS question-answering system for agriculture domain," in 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, Aug. 2015, pp. 1474–1478. doi: 10.1109/ICACCI.2015.7275820.

[13] M. Devi and M. Dua, "ADANS: An agriculture domain question answering system using ontologies," in 2017 International Conference on Computing, Communication and Automation (ICCCA), IEEE, May 2017, pp. 122–127. doi: 10.1109/CCAA.2017.8229784.

 ISBN: 978-1-68558-051-3

[14] S. K. Gaikwad, B. W. Gawali, and P. Yannawar, “A Review on Speech Recognition Technique,” Int J Comput Appl, vol. 10, no. 3, pp. 16–24, Nov. 2010, doi: 10.5120/1462-1976.

[15] D. Yu and L. Deng, “Automatic Speech Recognition,” 2015, doi: 10.1007/978-1-4471-5779-3.

[16] W. Fadel, I. Araf, T. Bouchentouf, P.-A. Buvet, F. Bourzeix, and O. Bourja, “Which French speech recognition system for assistant robots?,” in 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), IEEE, Mar. 2022, pp. 1–5. doi: 10.1109/IRASET52964.2022.9737976.

[17] “Introducing Whisper.” https://openai.com/research/whisper (accessed May 30, 2023).

[18] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” Dec. 2022, Accessed: May 30, 2023. [Online]. Available: https://arxiv.org/abs/2212.04356v1

[19] M. Skorikov, K. N. J. Omar, and R. Khan, “Voice-Controlled Intelligent Personal Assistant,” 2022, pp. 57–65. doi: 10.1007/978-3-030-92905-3_6.

[20] K.-P. Yang, A. Jee, D. Leblanc, J. Weaver, and Z. Armand, “Experimenting with Hotword Detection: The Pao-Pal,” European Journal of Electrical Engineering and Computer Science, vol. 4, no. 5, Sep. 2020, doi: 10.24018/ejece.2020.4.5.246.

[21] “Porcupine Wake Word Detection & Keyword Spotting - Picovoice.” https://picovoice.ai/platform/porcupine/ (accessed May 30, 2023).

[22] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, “Text classification using machine learning techniques.,” WSEAS Transactions on Computers, vol. 4, no. 8, pp. 966–974, 2005.

[23] K. Cho et al., “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 1724–1734. doi: 10.3115/v1/D14-1179.

[24] A. Shenfield and M. Howarth, “A Novel Deep Learning Model for the Detection and Identification of Rolling Element-Bearing Faults,” Sensors, vol. 20, no. 18, p. 5112, Sep. 2020, doi: 10.3390/s20185112.

[25] B. Mohit, “Named Entity Recognition,” in Natural Language Processing of Semitic Languages, Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 221–245. Accessed: May 11, 2023. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-45358-8_7

[26] L. Martin et al., “CamemBERT: a Tasty French Language Model,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 7203–7219. doi: 10.18653/v1/2020.acl-main.645.

[27] “spaCy · Industrial-strength Natural Language Processing in Python.” https://spacy.io/ (accessed May 30, 2023).

[28] P. Singh, “Machine Learning Deployment as a Web Service,” Deploy Machine Learning Models to Production, pp. 67–90, 2021, doi: 10.1007/978-1-4842-6546-8_3.

[29] C. Gyorodi, R. Gyorodi, G. Pecherle, and A. Olah, “A comparative study: MongoDB vs. MySQL,” 2015 13th International Conference on Engineering of Modern Electric Systems, EMES 2015, Jul. 2015, doi: 10.1109/EMES.2015.7158433.

[30] S. Tilkov and S. Vinoski, “Node.js: Using JavaScript to build high-performance network programs,” IEEE Internet Comput, vol. 14, no. 6, pp. 80–83, Nov. 2010, doi: 10.1109/MIC.2010.145.

 ISBN: 978-1-68558-051-3