IMPLEMENTATION OF AI CHAT BOT USING PROGRAMMING LANQUAQE GO

Nikola Vukotić
Academy of Applied Technical and Preschool Studies, Nis, Serbia, nikola.vukotic@akademijanis.edu.rs
Slavimir Stošović
Academy of Applied Technical and Preschool Studies, Nis, Serbia,
slavimir.stosovic@akademijanis.edu.rs
Dušan Stefanović
Academy of Applied Technical and Preschool Studies, Nis, Serbia,
dusan.stefanovic@akademijanis.edu.rs
Aleksandar Cvetković
Academy of Applied Technical and Preschool Studies, Nis, Serbia, srt5820@vtsnis.rs

Abstract: In recent years, the field of technical and technological sciences has certainly been marked by the rapid development of artificial intelligence. Apart from this area, wide application has been shown in many other areas as well. The use of neural networks and deep neural networks has contributed to the fact that artificial intelligence algorithms work much better than before, and therefore their application has become wider than ever. The advancement of artificial intelligence algorithms has contributed to the rapid development of a lot of different software that are very useful virtual assistants and chat bots. This paper aims to introduce the reader to the current achievements in the field of natural language processing, using the Wolfram alpha system. A chat bot was developed that integrates into the very popular Slack application, which is used by many teams in their work. The possibilities of training the model used by the Chat bot are discussed and the results of the work in several different test scenarios are presented. The Go programming language, which is increasingly relevant in the field of Computer Science and has found wide application in many highly reliable systems, was researched and used.
Keywords: Chat bot, Artificial intelligence, NLP, Wolfram alpha, Go, CNN

1. INTRODUCTION

In recent years, the trend of developing machine learning (Machine Learning - ML) has become extremely significant both in academic circles and in industry. Machine learning is a field of science that gives computers the ability to learn without being explicitly programmed. Machine learning is applied in many fields. For example, the spam filter is a machine learning program that has learned to separate spam emails from ordinary ones, states Aurelien (2019). BBC research "Will a robot take my job?" says that there is a good chance that professions such as bartenders, waiters, accountants, receptionists and taxi drivers will be automated by 2035 (Oliver, 2017). But there is no need to worry. Research on planned automation should not be viewed with concern. Artificial intelligence is advancing rapidly, but its adoption is still full of unknown and unpredictable challenges. Obstacles like delays are inevitable. Machine learning is not a simple field, it works on the basis of static algorithms managed and monitored by data scientists and machine learning engineers.(Oliver, 2017; Stylianou et al, 2018) Machine learning (ML) is used in many areas:

- Electronic commerce (Personalized shopping in the form of a recommendation system, virtual seller for correspondence between buyer and seller, fraud prevention when using sensitive data)
- Education (Digitalization of educational content, video lessons, conferences, instructions for using literature, voice assistants, personalized learning and monitoring of learning progress)
- Everyday life (Autonomous vehicles, spam filters, facial recognition, recommendation systems within the most used applications)
- Navigation Recommendations (Finding shortest route)
- Robotics Transporting goods in hospitals or factories Cleaning offices and large equipment
- Health (Early detection of the disease)
- Video game characters (which are not controlled by the player, but with the help of machine learning they behave as closely as possible to people in the real world)
- Social Networks Advertisements (tailored to a person's interests)
- Marketing Ads (tailored to a person's interests)
- Economy and Finance Automated messages

ML integration is possible in the form of messaging software on apps like snapchat, discord, slack, trello, atlassian and many others. While machine learning is developing at a rapid pace, there are still some challenges for

developers. Developers must have quality data, otherwise they will have problems with data processing and features, collecting, extracting and finding data takes a lot of time. Due to the rapid growth of this field, there is a need for more experts who deal with this machine learning and is still not a well-defined goal for business problems (Biswal, 2023; Tutorials Point, 2019).

2. NATURAL LANGUAGE PROCESSING

Natural language processing (NLP) is a field of computer science that deals with human communication, more precisely interpreting and learning human language. It is sometimes called natural language understanding (NLU) and natural language generation methods (NLG). The need for algorithms that understand human language is constantly growing, and NLP is here to reduce that need. Traditional NLP methods are based on linguistics, more precisely on the basic semantic and syntactic elements of language such as part of a sentence (Otten, 2023). These are some of the popular NLP models:

- GPT-4 (Generative Pre-trained Transformer 4)
- GPT-3: Generative Pre-trained Transformer 3
- BERT: Bidirectional Encoder Representations from Transformers
- ELMO: Embeddings from Language Models
- RoBERTa: Robustly Optimized BERT approach
- T5: Text-to-Text Transfer Transformer
- ALBERT: A Lite BERT
- XLNet: eXtreme Language understanding Network
- GPT-2: Generative Pre-trained Transformer 2
- ULMFiT: Universal Language Model Fine-tuning (Otten, 2023)

These are some of the NLP models with a free solution:

- GPT-3 (Generative Pre-trained Transformer 3)
- BERT (Bidirectional Encoder Representations of Transformers)
- Hugging Face
- AllenNLP
- Wolfram|Alpha (Mishra, 2023)

2.1. Transformer models

In recent years, a new model has emerged in the world of artificial learning that is becoming increasingly popular. It is a transformer model. A transformer model is a neural network that learns context and thus meaning by tracking relationships in sequential data such as the words in this sentence. This model applies mathematical techniques called "attention" or "self-attention" and they reveal the subtle ways in which even distant data elements in a sequence influence and depend on each other. For the first time, the public learned about this term in 2017 in the article "Attention is All You Need". The application of this model is wide. The Transformer model translates text and speech in near real time, opening up meetings and classrooms for the hearing impaired. They help researchers understand the chains of genes in DNA and the amino acids in proteins in ways that can speed drug design. People use the transformer model every time they search for something on the Google search engine. Transformers in many cases replace CNN and RNN neural networks, which until a few years ago were considered the most popular types of deep learning. About 70% of arXiv articles and papers on artificial intelligence published in the last few years mention transformers. Like most neural networks, transformer models are basically large blocks of decoders that process the data. Attention is so crucial to transformers that Google researchers almost used the term as the name of their model. However, individuals felt that the attention network did not sound very convincing, so in the end Jakob Uszkoreit came up with the name tramsformer. The GPT model series falls under transformers (Merritt, 2022).

3. WOLFRAM | ALPHA

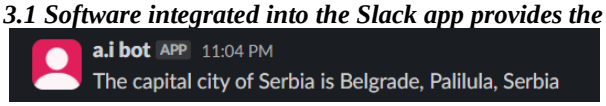
Wolfram|Alpha is a computerized knowledge system. It does not search for answers on the Internet, but finds answers in Wolfram's knowledge base. It also uses its own NLP model and belongs to the group of NLP with a free solution. Application Programming Interface (API) is a mechanism that allows communication between two software components. Wolfram Research, Inc. It has different number of NLP api (Wolfram|Alpha, 2023) and the most important one used in the work will be briefly explained below.

3.1 Short Answers API

This API returns the result of a single plain text. This text is taken directly from the Wolfram|Alhp results module. The API is designed to provide the shortest and simplest possible answers. If the answer cannot be short, the query will fail (Wolfram|Alpha, 2023). Figure 3.1 shows the answer of the software to the question "What is the capital of

Serbia?". Interaction with the software is done through the Slack (Slack, 2023) application. It was this API that was used in the practical implementation.

Figure 3.1 Software integrated into the Slack app provides the answer



4. WIT.AI

Wit.ai enables training of models powered by Wolfram. Training is machine learning from experience through examples (sentences) given by the programmer to the software, states Aurelien (2019). Wit.ai is an NLP modeling tool that helps developers easily embed an NLP model into messaging software to obtain structured data from messaging or voice (Gupshup, 2023). After the training, performance is measured to see the results of the training, more precisely, whether the software understands human language better. If the software understands the sentence over 90%, it is a good result and there is no need for additional training.

In order to train the software, you need to create an account on the wit.ai website, after which you need to create an application. The user will then be shown the following elements:

- Understanding (In this part examples are given to the NLP model)
  - Understanding review - checking the given examples in small series in order to optimize the model's understanding of the language
  - Utterance - Enter here the sentence that the software will study as an example
  - Intent - the goal or purpose that the user has in the context of the conversation with the software
  - Out of Scope - Training the application to recognize examples that should not be learned
  - Entity – a certain type of information or data that the user wants to extract from the user's input
  - Roles – the role of the entity
  - Resolved value – taken value that will be associated with the entity
  - Confidence - how percentage the software understands the given example/sentence.
  - Trait – additional information that can be associated with entities. Some of the traits are:
    - wit/bye (represents the entity for the intention of saying goodbye or ending the interaction)
    - wit/greetings (represents greetings or greeting messages from the user)
    - wit/on_off (used for entities related to turning something on or off)
    - wit/sentiment(used for sentiment-related entities in user messages
    - wit/thanks (refers to user expressions of gratitude)

Figures 4.1, 4.2, 4.3, 4.4, 4.5 and 4.6 show the results of understanding the software before and after the training. The greatest progress of software learning is shown in Figures 4.5, 4.6. The software could understand the question even with 69%, however, this can already confuse the software and there is a possibility that the answer will not be correct. After training, the software could understand the question 100%. It is important to note that the software can understand the questions better after training, but if the software does not have the answer to the question in its knowledge base, the software will say that the result is not available. Figure 3.5 shows an example where the software has no result.

Figure 4.1 Result without training 1

Figure 4.2 Result with training 1

Figure 4.3 Result without training 2

Figure 4.4 Result with training 2

Figure 4.5 Result without training 3

Figure 4.6 Result with training 3

5. WOLFRAM INTEGRATION WITH SLACK APPLICATION

The integration of Wolfram with the Slack application was done in the Go programming language. Go is a language that was created in September 2007 and was created by Robert Griesemer, Ken Thompson and Rob Pike, developers from Google. Go is used in Cloud & Network Services, Command-line interfaces (CLIs), Web programming, and other areas of programming. Packages contain programming code from other developers that is public and available for use (Go, 2023).

5.1 Project structure

The project directory contains 4 documents:

- **.env**
  - This document contains the necessary data used in the main program to integrate Wolfram with the Slack application.
- **go.mod**
  - This document declares the module, tells the computer which version of the Go programming language is used for this project, and declares the minimum required versions for packages to be imported
- **go.sum**
  - This document checks all sums to ensure that all packages have been successfully implemented.
- **main.go**
  - The file contains the main program, required variables and methods. This code allows the software to receive commands from Slack, use Wit.ai to analyze messages and the Wolfram Alpha API to get answers to user questions, and then send the answer back to users. Various packages are imported that are required for the program to function, including communication with Slack, Wit.ai for text analysis, the Wolfram Alpha API for getting answers to queries, and other required libraries. Variables are defined, wolframClient is initialized as a variable of type *wolfram.Client to enable communication with the Wolfram Alpha API. The printCommandEvents function is defined and is used to print events that the software monitors. It also receives the channel (<-chan) analyticsChannel through which the slacker receives.

Below are the important parts of the main program and their descriptions:

- godotenv.Load(".env")
  - Loads environment values (eg API keys) from an .env file.
- Slacker.NewClient(...)
  - Creates a new instance of slacker.Client to communicate with Slack, using SLACK_BOT_TOKEN and SLACK_APP_TOKEN.
- Witai.NewClient(...)
  - Creates an instance of witai.Client to work with the Wit.ai service, using WIT_AI_TOKEN
- wolframClient := &wolfram.Client{AppIDos.Getenv("WOLFRAM_APP_ID")}
  - Initializes the wolframClient with the AppID to access the Wolfram Alpha API.
- Go printCommandEvents(...)
  - Starts the go routine to print command events.
- Bot.Command(...)
  - Defines a command that expects the user to address the bot with a "query" message.
- Handler

- Processing this command. It extracts the question from the message, uses Wit.ai to analyze the message and extract relevant entities and intents. It uses the Wolfram Alpha API to get the answer to the question asked and sends the answer back to Slack.

- Bot Listener:
  - Using bot.Listen(ctx), the bot listens to events from Slack and responds to user commands.

6. RESULTS

This section presents the results of the research. Experiments were performed with different questions, and below are nine questions and Chat bot answers. Figures 5.1 – 5.10 show the questions and answers the bot answered. Every query should stard with "@a.i bot query - ", follower by a question.

By analyzing the results, it can be determined that the Chat bot can give short answers to short questions, as the idea of the model itself dictates. It can also be found that the bot is limited by its knowledge base, which can be seen in Figure 6.8 where the answer was not found, although it was previously determined that the model fully understood the question.

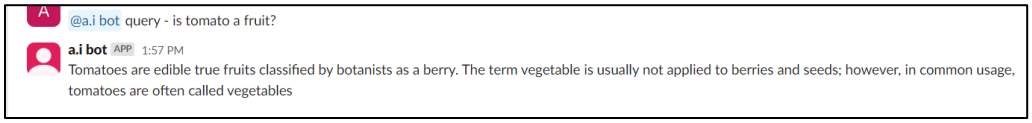


Figure 6.1 The result of the question "Is a tomato fruit?"

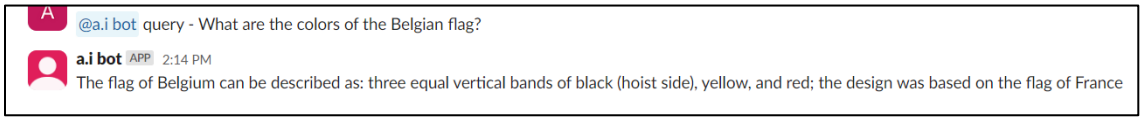


Figure 6.2 The result of the question "What are the colors of the Belgian flag?"

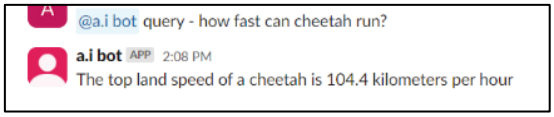


Figure 6.3 The result of the question "How fast can a cheetah run?"

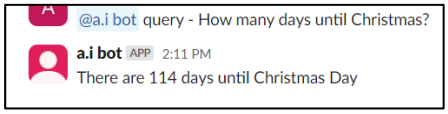


Figure 6.4 The result of the question "How many days until Christmas?"

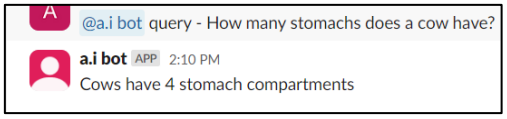


Figure 6.5 The result of the question "How many stomachs does a cow have?"

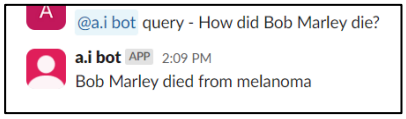


Figure 6.6 The result of the question "How did Bob Marley die?"

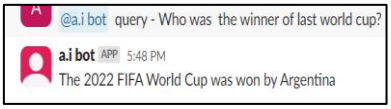


Figure 6.7 The result of the question "Who is the winner of the last world cup?"

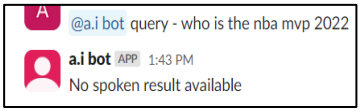


Figure 6.8 The result of the question "Who is the nba mvp 2022?"

7. CONCLUSION

Through this work, Slack software was developed that uses various technologies and APIs to allow users to ask questions and receive answers on the Slack platform. Integration with Wolfram Alpha and Wit.ai services enables obtaining relevant information based on the questions asked. Various concepts and techniques were applied, including the use of the Go language for backend development, working with APIs, parsing JSON data, as well as integration with the Slack platform. Using libraries such as slacker, wit-ai/wit-go, and krognol/go-wolfram, software is enabled to process commands and interpret user messages to provide relevant responses. By training, the software is trained to better understand the questions it is asked, but it cannot be trained to deliver results. In the questions and answers on the wolfram|alpha site, they say that it is best to ask the software a question in as few words as possible.

This way the software will understand the question better. There is also a possibility that the software has no results for the question asked. The Wolfram team says that there is a possibility that the question is not well worded and the question should be reformulated, or that the answer is still not in Wolfram's knowledge base (Merritt, 2022). This project represents a successful step towards creating a useful and interactive environment within the Slack platform. Further development may include extending functionality, adding optimizations and personalization to meet the specific needs of users and teams.

REFERENCES

Aurelien, G. (2019). *Hands-On Machine Learning with Sckit-Learn, Keras, and TensorFlow*, O'Reilly Media, Inc.

Biswal, A. (2023, August 21). *Top 18 Artificial Intelligence Applications in 2023*. Retrieved from https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/artificial-intelligence-applications#why_is_ai_used

Gupshup (2023, August 31). *Introduction to Gupshup and Wit.ai.* Retrieved from https://www.gupshup.io/developer/docs/bot-platform/guide/introduction-to-gupshup-wit

Go (2023, August 31). *The Go Programming Language.* Retrieved from https://go.dev/

Mishra, P. (2023, January 10). *Top 9 Open-Source NLP Projects*. Retrieved from https://medium.com/mlearning-ai/top-8-open-source-nlp-projects-5b2e4138118a

Merritt, R. (2022, March 25). *What Is A Transformer Model*. Retrieved from https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/

Oliver, T. (2017). *Machine Learning For Absolute Beginners*.

Otten, N.V. (2023, April 18). *Top 20 Most Powerful Large Language Models For NLP Tasks & Transfer Learning In 2023*. Retrieved from https://spotintelligence.com/2023/04/18/large-language-models-nlp/

Stylianou, N., Nurse, T., Fletcher, G., Fewster, A., Bangay, R. & Walton, J. (2015, September 11). *Will a robot take your job?*. Retrieved from https://www.bbc.com/news/technology-34066941

Slack (2023, August 31). *Where work happens | Slack*. Retrieved from https://slack.com/intl/en-gb/

Tutorials Point (2019). *Machine Learning with Python.* Retrieved from https://www.tutorialspoint.com/machine_learning_with_python/index.htm

Wolfram|Alpha (2023, August 31*). Wolfram|Alpha APIs: Computational Knowledge Integration*. Retrieved from https://products.wolframalpha.com/api