Bets Helper - A Sports Betting Chatbot

João Loureiro
j.pedro.loureiro@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal
May 2019

Abstract

The fast growth of Online Sports Betting around the world has led to the increase of the number of betters and, with them, the demand for betting information. In line with this demand, the objective of this project is to develop a sports betting chatbot, in other words, a sports betting personal assistant able to interact using natural language, being it via either voice or text-based conversations. The purpose of our chatbot is to collect the betting information that is spread throughout several platforms and provide it to the user on demand, in order for the users to avoid wasting time on manually searching for the information on the different platforms and so they get it as fast and with less work as possible. Also, there are no other chatbots available for the sports betting domain and no other platform supply a service similar to ours, so, in addition to enhancing the users’ betting experience, it is also an innovative system for users to use. As for the technology, we created our system using Google’s DialogFlow service for the chatbot logic and our own code for the information retrieval component.

Keywords: Online Sports Betting; Chatbot; Sports Betting Personal Assistant; Natural language; Betting Information; DialogFlow

1. Introduction

In this thesis, we will discuss the creation of a sports betting chatbot which will allow users to get information regarding sports betting, including betting tips, statistic probabilities and hour and date for matches, either via voice or text-based conversations.

Since its inception Online Sports Betting has had a steady growth but, in recent years, this growth got a lot faster and reached new heights, with an estimated global market worth of 56.7 billion dollars by 2018[9]. Such growth and size mean that this industry has a lot of potential users for new tools that improve people’s experience while betting.

Following this global fast growth is Portugal, especially after the first license for online sports betting was emitted to Betclic in May of 2016, one year after online sports betting was legalized in June, 2015[4].

Another big booster of sports betting popularity was the release of Placard, a game released by the Departamento de Jogos da Santa Casa da Misericórdia de Lisboa in September, 2015 which is very similar to online sports betting[16]. It served as the entry point to sports betting for many Portuguese people.

Figure 1 shows the percentile of bets for each sport in Portugal for the year of 2016[6]. It clearly shows football is the prime candidate for the Portuguese betters when choosing a sport to bet on, making it the main focus of our project.

In order to decide in which matches to bet, betters resort to a variety of platforms that each provide a different kind of information. In each of these platforms, users have to manually search for the information they want, none have a system to offer information to users on demand. That is where our system wants to improve. Plus, as will be explained on section 2, bets are quite time sensitive so a system that joins the information of several different types of platforms and provides it on demand is innovative and very helpful since it saves a lot of time for betters.

Figure 1: Online Sports Betting by sport
2. Background

There are a few important basic concepts when it comes to understanding betting, the most important being odds. The odds for an event reflect the probability of the outcome of that same event and, at the same time, determine the payout of a certain bet. For example, placing a bet of 1€ at 2.0 odds, in case of a win, pays 2€ and nothing in the case of a loss.

The two most common ways to represent odds are the decimal odds, shown as 1.22, and fractional odds, shown as 2/9, being the latter used mostly in the United Kingdom. Using fractional odds, placing a bet at, for example, 2 to 9 (2/9) odds represents a potential profit of 2€ for every 9€ bet. Meaning, a bet of 9€ at 2/9 odds, in case of a win, pays 11€ and nothing in the case of a loss.

There is also the concept of bookmaker, a platform used to place bets. Through bookmakers you can bet on the outcome of a certain event in a game for the offered odd. Bookmakers offer odds according to the probability they think is accurate for a certain outcome, which is why they change their offered odds over time and according to the circumstances.

The final concepts are sports tips, tipsters and tipster platforms. Sports tips are suggested bets for the likely outcome of a sports event. These are provided by tipsters through tipster platforms which are mostly websites where the tipsters, the people who come up with the tips, publish and share them for other people to follow, either with or without a match analysis as well.

3. Related Work

In this chapter we examine the most used types of platforms when it comes to sports betting in order to see what they have to offer and how they compare to our system. We also delve into the existing relevant scientific research related to our project.

3.1. Tipster Platforms

In this section we examine some of the most widely used tipster platforms, compare the features each one has to offer and explain our choice on which one to get the information from.

*Academia das Apostas* ¹ - *Academia das Apostas* is a Portuguese website that includes analysis from the editors for some of the most popular matches, articles about betting and betting news, competitions for tipsters to win money prizes and a forum for users to discuss tips and other betting related subjects. It also allows users to follow tipsters so newly inserted tips appear as notifications on the website. Notifications are not sent by email so the users are required to visit the website to view them.

*Bet Advisor* ² - *Bet Advisor* has a blog updated by the editors with articles about betting, sport news and match analysis for a few important games. They allow both free and paid tipsters. In order to have access to paid tipster tips, users are required to either pay a monthly subscription and have access to all the tips or buy tips individually and only have access to the tips bought. It is not possible to follow free tipsters, only paid tipsters. Notifications are received in real time but are also a paid service, included in the monthly subscription to a tipster.

*Betshoot* ³ - *Betshoot* provides analysis from the editors for a few games while also offering a section where users can comment on this analysis. They run a monthly competition with money prizes for the top tipsters but do not allow users to subscribe to tipsters.

*Bettingexpert* ⁹ - *Bettingexpert* runs a monthly competition for tipsters with a money prize for the best ones. They allow users to subscribe to tipsters and receive notifications when new tips are added.

*Blogabet* ⁸ - *Blogabet* does not have competitions for tipsters. They allow both free and paid tipsters and it’s possible to follow both categories. Paid tipsters are allowed to post some free tips but in order to have access to all tips, users need to pay a monthly subscription. Notifications by both email and phone are available, but are paid services, otherwise notifications require the user to check the website.

*OLBG* ⁴ - *OLBG* features a forum divided by sport where users can discuss betting subjects related to that sport. They run a wide range of monthly tip competitions for several categories and for specific competitions (e.g. Football World Cup). It is possible to follow tipsters and receive notifications when new tips are inserted via email.

When we examine these platforms features we can see that articles and match analysis are not widely offered by these platforms, being available on only two and three platforms respectively. These can be very useful features but are very hard to implement through a chatbot because of the amount of information we need to convey, so we are not going to focus on these.

Also, we can see that only three of the platforms provide external notifications, as in email or text.

---

¹www.academiadasapostas.com
²www.betadvisor.com
³www.betshoot.com
⁴www.olbg.com
message, to users that follow tipsters. Of those three, only OLBG\textsuperscript{15} offers this as a free service, the other two require payment. Also, Bet Advisor\textsuperscript{11} is the only platform that offers real-time notifications, which are included on their paid tipster subscriptions.

Finally, we can see tipster platforms are divided into two different models when it comes to rewarding tipsters. Either they have competitions that reward the best tipsters on the platform or they allow tipsters to be paid for their tips. To collect the tip information we relied on web-scraping, which is explain on Section 4.4. We choose to scrape this information from Bettingexpert\textsuperscript{9} because it was the only platform which offered a tip summary, simplifying our job, and also because it was one of the easiest platforms to scrape information from.

The main difference between these platforms and our project is that, even tho we offer similar information, none of these platforms offer the information to users on demand, the user has to manually search for the relevant information. Our chatbot improves on this model by allowing users to get the information they want via voice or text request, without the need to specifically look for it. It is also innovative in the sports betting area since none of the existing platforms do anything similar.

3.2. Match Prediction Platforms
Besides tipster platforms, another popular service among betters is match prediction platforms. These platforms make predictions for the outcome of matches based on statistical data, which may include previous direct confrontations between the two teams, injured players, among others.

We explored two of these platforms, ProSoccer\textsuperscript{5} and Statarea\textsuperscript{6}, in order to examine the features they have to offer.

Both platforms offer their prediction services for free. ProSoccer\textsuperscript{9} has the advantage of using a computational intelligence software to generate their predictions while Statarea\textsuperscript{10} does not specify how they generate theirs, but Statarea\textsuperscript{10} offers a wider variety of predictions and allows users to give their opinion and feedback.

Our project improves upon these platforms by offering the same available information on demand, via voice or text request, so users do not have to manually search the website for the desired information.

To collect this information we relied on web-scraping, which is explain on Section 4.4. We could have scraped the information from any of the two websites but in the end we chose Statarea because of the wider selection of available predictions and also because it was easier to scrape than the alternative.

3.3. Scientific Research
In this section we delve into the existing relevant scientific research related to our project.

3.3.1. Chatbots: Are they Really Useful?
In this paper [18] Shawar & Atwell investigate areas where chatbots could be useful, such as education, information retrieval, business and e-commerce. They also present several chatbots with useful applications found in daily life, such as help desk tools.

They start by explaining that a chatbot system is a software program that interacts with users using natural language. The authors also give a brief historical introduction to chatbots and explain how the technology evolved, starting when it began in the 1960’s with the aim of mimicking human conversation and amuse the user.

In the rest of the paper, the authors talk about using chatbots for four different purposes:

As a Tool of Entertainment - When talking about chatbots as tool of entertainment, the authors explain that initial aim of building chatbot systems was to mimic human conversation and amuse users. They give the example of ELIZA\textsuperscript{21}, the earliest example of a chatbot in this paper and which was created in the 60’s by Joseph Weizenbaum to emulate a psychotherapist in clinical treatment.

As a Tool to Learn and Practice a Language - In order to see if chatbots could help users learn and practice a language the authors converted a Corpus of Spoken Afrikaans to the AIML format files and then re-train ALICE to create two Afrikaans chatbots. They then encouraged open-ended testing and feedback from users in South Africa. During these tests, users found that many responses were not related to the topic or were nonsense but they still found the chatbot an interesting tool to practise the language and enjoyed chatting with it.

After analysing similar experiments\cite{13, 8}, the authors found that chatbots could be useful for students learning new languages or practicing old ones for several reasons, including: (a) Chatbots being able to repeat the same material with students several times without being boredom; (b) Many chatbots using both text and speech mode when responding, which provides an opportunity to practice reading and listening skills; and (c) Chatbots being a

\begin{itemize}
\item \textsuperscript{5}www.prosoccer.gr
\item \textsuperscript{6}www.statarea.com
\end{itemize}
new trend, which improves the students motivation towards learning.

**As an Information Retrieval Tool** - The authors give several examples of chatbots used as an information retrieval tool. The leading example being FAQChat, which was a retrained version of ALICE after the author’s program was adapted to the FAQ (Frequently Asked Questions) of the School of Computing at the University of Leeds. FAQChat provides answers about topics that include the Unix operating system, teaching and research resources.

**Chatbots in Business and other Fields** - Finally, the authors give three examples of chatbots in other domains: (a) There is Happy Assistant[1], a chatbot that helps users access e-commerce sites to find relevant information about products and services; (b) Sanelma, a chatbot that impersonates a fictional person that users can talk with in a museum and that provides background information concerning a certain piece of art; and (c) Rita (real time Internet technical assistant)[19], used in the ABN AMRO Bank to help customers do financial tasks such as money transfers.

After having surveyed several chatbot systems which succeed in practical domains like education, information retrieval, business, e-commerce, as well as for amusement, the authors conclude that in general, the aim of chatbot designers should be to build tools that help people, facilitate their work and their interaction with computers using natural language, but not to replace the human role totally, or imitate human conversation perfectly.

We found this paper relevant for our project first because it gave us an historical introduction to chatbots, it showed us how they worked and what they were used for since the creation of the technology. It also showed us that chatbots can be successful in several domains, including information retrieval. This is a positive sign for our project because with the growing popularity and more widespread use of chatbots in several areas, a chatbot for sports betting should be well received among users.

Also, our chatbot works basically as a information retrieval chatbot since it collects information from several sources when the user requests, so this paper gave us an assurance that it would be a viable solution.

**3.3.2. Case Study: Building a Serverless Messenger Chatbot**

In this paper[14] Lehvä, Mäkitalo & Mikkonen present a two-month case study on building and designing a Facebook Messenger chatbot, for a media company, that could help the users follow up on the latest news of their interests. The goal of the project was to design a scalable, modern architecture for a chatbot that follows liquid software principles.

The authors start by introducing and explaining the technologies they used to create the chatbot.

The interface for the chatbot is provided by the Facebook Messenger, a service for instant messaging with Facebook friends. To start a conversation with the chatbot, the user simply has to search for the chatbot by the name of the Facebook page they are linked to on the Messenger.

To build the backend, the authors used the Amazon Web Services (AWS), which is a cloud service platform built by Amazon the offers a wide range of services for developing software.

Next, the authors explain the motivation behind using a serverless computing approach and why they decided to use a serverless architecture, implemented using AWS, for their project.

Using a serverless computing approach, the developer does not need to manage servers. Instead, the developer can simply just upload the code to the serverless environment and the code gets executed when it is triggered by some event. Also, the serverless environment does the scaling automatically depending on the workload. Furthermore, serverless computing can be very cost efficient as the customer only pays when the code is being executed and also because, due to not requiring system administration work, the serverless approach can also lead to cost savings in operational management[15].

We found this paper relevant for our project because it shares some important similarities to ours. The most important being:

- Both chatbots are used for information retrieval, albeit for different themes: sports betting on our case and news providing for theirs. This gave us an example of how these chatbots work which helped us get a better understanding for building ours.

- Both rely on a serverless approach, even tho with different providers: they used AWS while we rely on services provided by Google. This paper gave us a very good insight on both the advantages and disadvantages of using a serverless approach in order to build a chatbot, since the authors go in depth into the analysis of this theme.

In conclusion, this paper gave us a real example of what we were trying to accomplish, a information retrieval chatbot using a serverless approach, gave us an idea of the difficulties we could face, as well as possible solutions to these problems, and, more importantly, showed us that it would be possible.
3.3.3. YourQA

In this paper [17], Quarteroni & Manandhar report their experience on the design, implementation and evaluation of a chatbot-based dialogue interface for an open-domain question answering (QA) system, while showing that chatbots can be effective in supporting interactive QA.

The system is named YourQA and it is able to provide both factoid and complex answers such as definitions and descriptions. It takes the top 20 Google results for a question, retrieves the corresponding Web pages and analyzes them to extract answers and rank them by relevance to the question.

Dialogue interfaces pose complex evaluation challenges and so are often evaluated using qualitative metrics such as user satisfaction and perceived time of usage [20]. So, when it came to evaluate their system, the authors designed three scenarios where users had to look for two different items of information relating to the same topic.

After that, users were asked to fill in a questionnaire about their experience. It was composed of nine questions where question one and two assessed the performance of the system, question three and four focused on interaction difficulties, five and six related to the overall satisfaction of the user, seven and eight focused on the system performance and question nine assessed which of the versions the users preferred.

In the end, the authors drew optimistic conclusions on the feasibility of chatbot based interactive QA. This is relevant for our project because it is another example of a successful information retrieval chatbot we learned from.

Most importantly, this paper was relevant because we used it to learn how to evaluate our system. It explained the difficulties faced when evaluating a system like this and gave us an understanding on the options we had to evaluate our system, including introducing us to the user evaluation approach, which they used and that we ended up using as well. Also, it provided us with additional material we learned from for this same matter.

3.3.4. System for providing real time sports betting information

In this paper and in a different vein from all the previous papers, we have a patent [2] for a method and system for providing real time sports betting information and where Corbo explains the motivation behind building a system for online sports betting and describes the system in question, including examples of several different screens of the system.

The system is built to provide sports betting information to users in real time, in order to allow users to make more intelligent decisions regarding their bets. Its features include:

- Providing the user with a schedule of all sporting events for the given time period, the different bets that can be made on each game and the respective odds.
- Providing the user with a means to log and track bets for a complete total of winnings and losses.
- Alerting the user when a predetermined occurrence takes place in connection with the odds. These alerts include one for a predetermined line becoming available and another for a “major line move” between the various sports books and casinos. For example, if the user wishes to place a bet at specific odds, he can set the line seeker alert, and that will notify him when any sports book offers those particular odds. The major line move alert can be set to notify the user if a predetermined amount of sports books change their lines, on a specific game, within a predetermined amount of time.

Regarding our project, this paper allows us to compare our project to an existing sports betting information retrieval system, and even though it is not a chatbot, it still a relevant system in the same domain as our own. It also gave us an idea of which sports betting information might be useful for users when it comes to making the right decisions for their bets.

4. Proposed Solution

In this chapter we will go into further detail on the features and implementation of the system. We will also describe the chosen architecture, meaning we will explain each part that constitutes the system, why we chose them and how they work together.

4.1. Functionality

Our system is first and foremost a personal assistant or, in other words, a chatbot. That means it is capable of interacting with users using natural language, either via voice or text-based conversations. In this case, the chatbot works in English.

Being a sports betting oriented chatbot means it provides information that helps users chose the right bets to make. We built our system to provide five different types of information:

**Main Odds** - Users can ask for the odds for a match, in which case the system returns the main odds for the indicated match, which include odds for: (a) the full-time result\(^7\); (b) the half-time result\(^8\); (c) over and under 0.5, 1.5

\(^7\)A full-time result can be either home team win, away team win or draw

\(^8\)A half-time result refers to the result at a match’s half-time which can be a draw, home team winning or away team winning
and 2.5 goals; (d) both teams to score or not; (e) double chance\(^9\); and (f) draw no bet\(^{10}\).

**Statistic probabilities** - Users can ask for statistically predicted probabilities, in that case the system returns probabilities for: (a) the full-time result\(^{13}\); (b) the half-time result\(^{14}\); (c) over 1.5, 2.5 and 3.5 goals; and (d) both teams to score. Beyond these, the system also returns a tip based on this probabilities.

**Betting tips** - Users can ask for betting tips for a given match in either of two ways: (a) If they specify a tipster, the system returns that tipster’s tip for that match; or (b) if they do not, the system returns a tip summary made from all the tips tipsters posted for the specified match.

**Date** - Users can ask for the date of a match between two teams, in this instance the system returns the date for the next time the two specified teams play.

**Team Next Matches** - Users can ask for the next matches of a team, in which case the system returns the next five matches of the specified team, in chronological order.

The chatbot is made to be user friendly, so we added several features in for it to better steer the conversation in the right direction and to help the users get the information they want in the easiest way possible:

1. When the user inputs only a match with no information request, the chatbot asks the user what type of information they would like to know about that match.

2. When a user requests information about a match, after answering that request, the chatbot asks if the user would like to know anything else about that match. This way the user does not have to input the same match several times in a row.

3. When the chatbot can not find the match a user requested information about, it provides several suggestions of matches that the user might be talking about. These come in the form of each team’s next match.

4. The chatbot has a help command with several options. When asked for help in general, the chatbot explains it’s purpose, what commands are available and how to use them to get information. When asked for help for a specific command, the chatbot explains what that command does and how to use it.

---

\(^9\)A double chance is a bet that consists of betting on two possible outcomes, meaning either home team win or draw, away team or draw and either team to win

\(^{10}\)A draw no bet refers to betting in a team winning but the bet money is returned if the game ends in a draw

### 4.2. Architecture Overview

To help better understand the architecture of our system we created a simple diagram, as can be seen in Figure 2. It shows our system is divided into three parts:

**Google Assistant** - Which acts as the user interface. Google Assistant receives the user input and passes it as text to Dialogflow. The opposite happens for the output, the response is received as text and converted to speech for the user.

**Dialogflow** - Which contains the chatbot logic. Dialogflow parses the text input in order to understand what the user wants and passes this information to Google Cloud Functions. After receiving the response, Dialogflow passes it to Google Assistant.

**Google Cloud Functions** - Which hosts the code of the system. It receives the request from Dialogflow, scrapes the needed information and returns the final processed response to Dialogflow.

---

The first step for building our system was the creation of the Dialogflow agent, followed by writing, hosting and connecting the code and finally integrating Google Assistant with Dialogflow. Each of these parts is going to be further individually explained in the following sections.

### 4.3. Dialogflow

Dialogflow is a service owned by Google that makes it easier for users to create a chatbot by providing a platform that allows developers to create agents, which house all the logic needed for the human-computer interaction, while incorporating Google’s machine learning expertise[3].

We found Dialogflow was, by far, the easiest and least time-consuming way to create a chatbot since it saved us the work of creating all the logic needed to handle the human-computer interaction. We started by creating a Dialogflow agent, BetsHelper.

The base of each agent are Intents. These represent the different subjects the agent is supposed to be able to discuss.

Within each of these Intents we specified several Training Phrases, which are examples of possible
input phrases specified by the developer. These are needed so the agent can determine the Intent each input phrase belongs to. It does this by determining the closest match between the input phrase and the Training Phrases of all Intents.

The agent also needs to know what information is useful within each input phrase. To do this, developers specify Entities which are key concepts needed to answer the user's request. Dialogflow includes a few predefined system entities, like the date Entity for instance.

The last part we had to set up on Dialogflow was the Fulfillment. While the Intents identify what the user wants, the Fulfillment is the code that is hosted elsewhere, connected to Dialogflow through a webhook and that handles the response.

When Dialogflow receives an user input, it matches it to an Intent and sends an HTTP POST request to the webhook with a JSON object containing the information about the Intent and the Entities of that same input. The answer also comes in the form of a JSON object containing the answer ready to be delivered to the user. The code and what it does is explained in Section 4.4.

The second big advantage of using Dialogflow is the ease of training the agent. Using Google's machine learning technology, an agent learns both from the examples provided in the training phrases and from the language models developed by Dialogflow. Based on this data, it builds a model for making decisions on which intent should be triggered by a user input and what data needs to be extracted. It then keeps improving though real conversations with users.

It is also possible to view the past conversations logs and if necessary manually assign certain inputs to a specific intent. We had to do this in very few occasions in order to fine tune the agent.

4.4. Code
The code we wrote for our system has the main purpose of formulating a response for the user query using the information retrieved via webscraping. We choose to write our code in the Python programming language mainly because of Beautiful Soup, a Python library for pulling data out of HTML and XML files which makes webscraping a lot simpler and easier. We had to resort to web scraping, which is the act of extracting data from websites, because there are no available APIs that provide the information we require.

When Dialogflow sends the request, the code handles it and extracts the Intent and the Entity parameters. It then checks the Intent in order to know what information the user wants since each Intent relates to a feature, as explained in Section 4.3. Regardless of which Intent is used, the Entity parameters derived from the user’s input are used as a base to webscrap the information. Finally, using the acquired information, we formulate the response the chatbot will give the user and send it to Dialogflow.

4.5. Google Cloud Functions
We chose to host our code using Google’s Google Cloud Functions service, which is an event-driven serverless compute platform part of Google Cloud Platform.

We opted for a serverless approach because it allows us to simply deploy our code to Google. Google Cloud Functions runs the code and dynamically scales it to match the usage. This shifts the worry about server management and security patches away from the developer[11, 10, 15, 5, 12].

We chose to use Google Cloud Functions because being an serverless platform, it enjoys the benefits explained previously but also because it provided a webhook which we needed to set up the Dialogflow Fulfillment. It was the easiest and least time consuming way of setting up the code hosting and the Fulfillment part of our project.

Google Cloud Functions also provides logging and monitoring for our code. There are very important features because they allow us to debug our function and track its performance.

4.6. Google Assistant
Google Assistant is a virtual assistant developed by Google, in other words, it is a software agent that can perform tasks or services for an individual. It is able to communicate with users via both voice and text, meaning it interprets human speech and responds via synthesized voice.

We chose this platform for the user interface. This means Google Assistant is the visual part of our system through which a user interacts. It receives the user input and passes it as text to Dialogflow while the opposite happens for the output.

We decided to use Google Assistant as the user interface instead of developing a new one because the integration between Dialogflow and Google Assistant is immediate. Within Dialogflow we have the option to export the agent to Google Assistant, requiring just a few other additions, which include device selection and submission for approval from Google, for the agent to be operational through it. Also it is customizable and, since it is owned by Google, we can expect good functionality and high design standards.

This decision saved us the time and work of building a user interface from scratch and is the much easiest and least time consuming way when compared to the available alternatives.
5. Work Evaluation

In order to evaluate our system, since dialogue interfaces pose complex evaluation challenges, we decided to test its performance, usability and user satisfaction through the users test evaluation approach[7]. This approach consists of creating several tasks that encompass the overall usage of the system, having the testers do them and examining how they perform. This allows us to register metrics such as the time of usage which we use to evaluate the system’s usability. Following the tests, we also had the testers fill a questionnaire in order to measure their satisfaction with the system.

5.1. Testing

For the user testing we started by developing the use cases, which are tasks that encompass all the functionalities and that reflect the intended typical usage of the system. For this purpose we came up with the following four tasks:

T1 - Find Tottenham next matches and the date for the Chelsea vs Tottenham match;
T2 - Get the probabilities for the Chelsea vs Tottenham match;
T3 - Get the odds for the Chelsea vs Tottenham match;
T4 - Get the tips for the Chelsea vs Tottenham match and the tip by “MisterTips” for that same match.

These tasks were subjected to pilot tests in order to guarantee that they were well formulated and that made sense to testers. Also, each task was also given a clear definition of when it was successfully completed so we could accurately record the usability metrics. For example, the third task would be successfully completed as soon as the chatbot presented the odds.

Next we choose our usability metrics, which were the metrics we recorded in order to evaluate the system. We settled for the following three:

**Task Duration Time** - The time the tester took to complete the task;

**Number of Errors** - The number of errors the tester did during the task, with an error being a mistake that required the task to be restarted from the starting point;

**Task Completeness** - If the tester was capable of finishing the task with success.

With both the tasks and the metrics defined, we then proceeded on to the actual testing. We started by giving a short introduction with less then five minutes which explained to the testers what is a chatbot and how to communicate with our system, what is sports betting and what kind information is possible to obtain from our system. After this explanation the users were allowed to use the system for 2 to 3 minutes.

Following this introduction, we moved on to the tasks. As stated before, the users were asked to complete four tasks while we recorded the time they took and the number of errors committed. Each tester did the four tasks twice, first using our system and then using the existing alternatives. This was done so we could compare both metrics and conclude which platform is best for the users.

Since no other platform offers all the services our system offers, different tasks had to be compared to different platforms. So task 1 was compared to Google\(^{11}\), task 2 to Statarea\(^{12}\), task 3 to Betclic\(^{12}\) and task 4 to Bettingexpert\(^{16}\).

We collected the times from the tests with thirty different users and calculated the average and standard deviation of the time they took to complete each task. The following Tables 1 to 3 shows both this times, the total number of errors the users made in each task and if they were able to complete the task or not.

<table>
<thead>
<tr>
<th>Task</th>
<th>Bets Helper</th>
<th>Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19.29 ±2.47</td>
<td>32.06 ±6.09</td>
</tr>
<tr>
<td>2</td>
<td>11.51 ±3.02</td>
<td>35.90 ±23.05</td>
</tr>
<tr>
<td>3</td>
<td>12.83 ±4.26</td>
<td>19.46 ±4.70</td>
</tr>
<tr>
<td>4</td>
<td>20.50 ±10.75</td>
<td>40.79 ±19.22</td>
</tr>
</tbody>
</table>

Table 1: Tasks Average Time Table

<table>
<thead>
<tr>
<th>Task</th>
<th>Bets Helper</th>
<th>Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 2: Tasks Errors Table

<table>
<thead>
<tr>
<th>Task</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3: Tasks Completed With Success Table

As can be seen from the results, our system is faster in all tasks, with the alternatives taking, in some cases, more than double the time of our system to provide the same information. Also, the testers were more prone to mistakes using the alternatives than when using our system.

The errors that occurred when using our system due to the user inputting mistaken information (the

\(^{11}\)www.google.com

\(^{12}\)www.betclic.pt
wrong match date, for example). As for the alternatives, both the platforms used during the second and fourth task, Statarea\textsuperscript{12} and Bettingexpert\textsuperscript{5} respectively, can be hard to understand if never used before, which caused the larger number of errors.

Also, all users were able to finish all four tasks with success.

To better examine the results, we also calculated averages dividing the results between experienced and inexperienced betters. As is better explained in the next Section 5.2, out of the thirty testers, twenty-one were considered inexperienced betters while the remaining nine were considered experienced betters.

From the results we found that our system is faster for both groups but the inexperienced betters are considerably faster using our system. Also, inexperienced betters are more prone to mistakes than experienced betters when using the alternatives.

Our system seems like an improvement over the alternatives for all users but mostly for inexperienced betters, mostly because it is easier to get information by speaking than by using platforms never used before and because our system is made to be user friendly.

5.2. Questionnaire
After finishing the tasks, each tester was asked to fill a user satisfaction questionnaire. The questionnaire was comprised of eight questions and was designed to be simple and fast to answer, as not to over saturate the testers and so the tests did not take too much of their time.

The first three questions of the questionnaire were multiple choice questions and aimed to characterize the tester. The first question was to determine the tester’s age with the answers being twenty-four male and the six female.

As for the second question, which was to determine the tester’s age, two had an age between 18 and 21, twenty between 21 and 27, three between 28 and 35, two between 36 and 41 and three had an age above 41.

On Table 4 we can see the answers for question three, which determined the academic qualifications of the testers.

Our system is built to be user friendly and to be used by both new and experienced betting users so it was important to test it with both kinds of user in order for us to compare it to the alternatives and to identify the different difficulties each group had.

The last three questions assess the testers opinion on the system which is essential for us to examine the system usability and user satisfaction. Questions 6 and 7 were both answered using a scale from 1 to 5, where on question 6 1=“Very Hard” and 5=“Very Easy” and on question 7 1=“Not at all” and 5=“Very”. Question 8 was a multiple choice question. Tables 7 to 9 show us the collected answers for this questions.

We tried to get a diversity of testers from different backgrounds because our system is supposed to be used by a large demography which involves many different kinds of people, so our tests aimed to encompass as many different people as possible.

The following two questions were meant to assess the sports and betting knowledge of the testers. They were answered using a scale from 1=“Never” to 5=“Every Day”, where users had to choose one as the answer.

Tables 5 and 6 show the count of answers for this questions. From this table we can see that most testers have a fairly decent following when it comes to sports, with twenty-five of the answers being either 3 or 4, and that one third of the testers have a good knowledge when it comes to betting, with answers 3 or above, which we considered experienced betters, while the other two thirds had very little knowledge of sports betting with answers of 2 or below, which we inexperienced betters.

<table>
<thead>
<tr>
<th>4. How often do you follow sports?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>Average</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5. How often do you engage in sports betting?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>Average</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. How easy was it to obtain the desired information?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7. Did you enjoy using the system?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

We tried to get a diversity of testers from different backgrounds because our system is supposed to be used by a large demography which involves many different kinds of people, so our tests aimed to encompass as many different people as possible.

The following two questions were meant to assess the sports and betting knowledge of the testers. They were answered using a scale from 1=“Never” to 5=“Every Day”, where users had to choose one as the answer.

Tables 5 and 6 show the count of answers for this questions. From this table we can see that most testers have a fairly decent following when it comes to sports, with twenty-five of the answers being either 3 or 4, and that one third of the testers have a good knowledge when it comes to betting, with answers 3 or above, which we considered experienced betters, while the other two thirds had very little knowledge of sports betting with answers of 2 or below, which we inexperienced betters.
8. When comparing Bets Helper to the alternatives, which one was your favorite?

<table>
<thead>
<tr>
<th>Bets Helper</th>
<th>Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 9: Question Eight Table

From the tables we can see the count of answers, with questions 6 and 7 also having the average with standard deviation. We can see the answers are very positive, with 29 and 30 of all testers rating 4 or above on both questions 6 and 7, which translates to an answer average of 4.23 for both questions.

Question 6 was meant to assess the system usability while question 7 was meant to assess the user satisfaction with the system, so this kind of feedback is a very positive sign for our system since it means that the system is good at supplying the desired information and that the major part of users like using it.

Also, question 8 shows that every tester preferred using our system to the alternatives. We should note that some experienced betters stated that, while they preferred using our system for these specific tasks, for normal betting some of the alternatives may be have to be used occasionally for specific situations. But they also stated that having access to all the different kinds of information our system offers is very useful and saves a lot of time when betting that would be otherwise spent alternating between platforms.

6. Conclusion

In recent years, online sports betting has had a very fast growth and reached a new peak in popularity. We took advantage of this growing industry to create a system that would improve the betting experience for both the new and experienced betters using an innovative technology in the field.

We used a Google platform named Dialogflow to create the chatbot with great results and our own written code to get the information needed for our system to work, which in the end worked in a fast and efficient manner. Using a combination of these two elements we managed to build a good and very solid system.

To test and evaluate the system we opted for the users test evaluation approach with thirty users, using tests that reflect the real world usage of the system. The results we collected from these tests were very positive, the system performed better than the alternatives, all while providing several types of information that previously required four different platforms to be consulted.

Also, the system was made to be user friendly and the results reflected that. The users easily adapted to the system, found it easy to use and enjoyed using it. The use of the chatbot technology was a good choice in this matter since it is a type of technology easy for users to adapt to. It was especially good for users with very small betting experience because it is more intuitive than the alternatives, which required much more experience to use effectively.

References