UNIVERSITY OF Macedonia

Department of Applied Informatics

BookPal

Chatbot using Regular Expressions, written in Python

Konstantinos Vrazalis, ics22115

Lesson: **Computation theory**

Professor: **Yannis Refanidis**

Thessaloniki, April 2024

Table of Contents

1. Introduction

BookPal is a **chatbot** written in **Python**, which makes use of **REGULAR expressions in** order to recognize patterns in the user's sentences and respond accordingly. The theme of this chatbot is to provide information about **books**. The user can ask BookPal for information about the publication date, writing language, summary and online purchase link of a book. In order to provide broader functionality, Google's book Application Programming Interface (API) is used to provide corresponding information about any book the user wants. Thus, his primary question is whether there is a book with the particular title he has in mind. If BookPal answers in the affirmative, the user can formulate the above questions. The source code of the application, as well as the corresponding executables for Linux, Windows and macOS, can be found on the developer's GitHub.

2. Purpose of the application

The purpose of this paper is to familiarize the reader with the writing and recognition of regular expressions at the programming level. For this purpose the **module re**[1] of Python, which provides the appropriate functions to achieve this purpose. On a second level, an attempt is made to create a model graphical interface to simulate a messaging application with the **tkinter** graphics library[2], and finally an introduction to the use of APIs for retrieving information from the web is made, using the **Google Books API**[3].

3. Programme structure

3.1 Introduction

In order to make the code more readable, the program is divided into **three files**, each of which takes care of implementing the **API**, **recognizing REGULAR expressions** and **implementing the graphical interface**, respectively. These files are linked (via *import*), as shown in the following figure[4]:

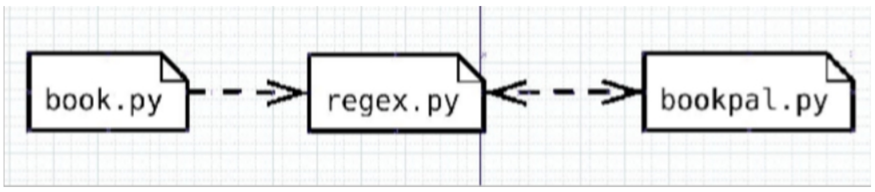


Figure 1: Structure of the program files

A detailed explanation of the functions of each file is given in the respective subsections. It is worth noting that each file uses the data received from the previous one as abstract tools and does not deal with the details of their implementation. Therefore, the end user only needs to run *bookpal.py* (the graphical interface file) and the program will be fully functional.

3.2 File book.py

This file is responsible for storing data for a book. To do this
this creates a **Book class**, containing a single method, which **RETURNS** a
a one-dimensional **string table**, where each element of the table is the type of information indicated by its name. Thus, the returned *book_info[]* table includes: title and author information, publication details, language information, summary information, and an online purchase link. This information is then added to the BookPal responses to the user, so that each response is personalized for a particular book.

In order to assign values to the above fields, the *requests* library is used.[5] Python library to make **HTTP** requests to https://www.googleapis.com/books/v1/volumes, with the title of the book entered by the user as a parameter. For simplicity, the variables are matched to the elements of only the first title match, as the *requests.get()* method returns all possible matches found by the API.

3.3 File regex.py

This file is responsible for **configuring** the answers to the user's questions. Its purpose is to identify regular expressions in the input string. For this reason it does *import re*, for the use of library functions. Furthermore, it imports the *book.py* file, for **enriching** the answers, as mentioned in the above subsection.

The regular **expressions**, together with their corresponding **answers,** are given in the table
patterns_responses[]. In the left column are the expressions, as defined in the way described in Section 4. In the right column are the initial parts of the sentences that BookPal will respond to.

In the *chatbot()* method, given the input string, the table is scanned sequentially and the *re.match()* method is called to **match** the user's sentence to a **pattern**, if detected. The existing responses are then enriched according to their position in the table and the position of the corresponding element in the *book_info[*] table. If there is no pattern match, BookPal responds in a general way that it does not understand the question. The method finally **RETURNS** a string with BookPal's **answer**, to be used by the next file.

3.4 File bookpal.py

This file is responsible for creating the user **interaction interface.** Its name is generic for ease of execution, as this is the file that needs to be executed for the entire program to work.

The *tkinter* library is imported to create the graphics and the *regex* file to download the BookPal answers. The process of generating the window elements will not be discussed in this document, as it is outside the scope of this work (the code contains detailed comments). In general, two **frames** are created, one for inserting the user's message into the corresponding input field and one for displaying the user's and BookPal's messages.

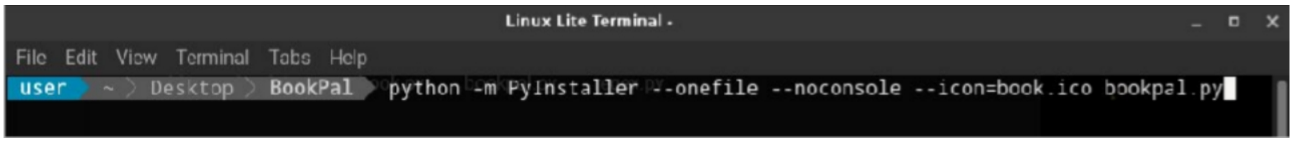
Substantial functionality is provided by the *send_message()* function, which displays the **message of the user** and then the **BookPal response** for the given input, via the
regex.chatbot(), which takes the input message as a parameter.

Essentially, this file takes care of passing the user's message to *regex.py to* detect the regular expression, and then displays the response returned by the *chatbot()* method. This process **is REPEATED** until the user **MANUALLY** terminates the program**,** as, for simplicity, there is no specific input statement that stops execution.

3.5 Notes on executable files.

To run the program, the user just needs to run the *python* **command** *bookpal.py* at
/path_to_folder/BookPal/, on a Linux or macOS **terminal** and in CMD/PowerShell on Windows or run the file graphically from its **IDE.** The application window will then appear. However, as it is possible that a user may not have some of the modules installed, he will get the corresponding **error message** and will have to install them via **pip**[5].

In order to avoid the above problem, but also to allow the user to run the program directly, Python allows **all files to be COMPRESSED** and **all libraries TO BE INCLUDED** in one executable file, via the **PyInstaller** module[6]. The BookPal folder contains executables for **Linux**, **Windows** and **macOS**, which were created by running the command:



```
user ~ Desktop BookPal  python -m PyInstaller --onefile --noconsole --icon=book.ico bookpal.py
```

Figure 2: Compression command in an executable file

In the image above:

a) **-m**: indicates the use of a Python module
b) **PyInstaller**: the name of the module
c) **-onefile**: indicates that the executable will be a single file
d) **-noconsole**: indicates that no terminal will be opened at runtime
e) **-icon**: denotes the icon[7] icon of the executable (the .ico file is included in the BookPal folder)
f) **bookpal.py**: declares the file to be compressed

Running the above command creates the **executable** file in the *BookPal/dist/* folder, depending on the type of operating system. This file can be run on any compatible computer, **without** necessarily having **Python installed.** Note that on **Windows 10** and above, as code that uses APIs is also requested to be compressed, the command will not be allowed to run as long as **Windows Defender** or a general **Antivirus is** enabled. Similarly, if an attempt is made to run bookpal.exe on a computer with one of the above, there will be a **security warning message**. In fact**, there IS no**
absolutely no risk of running the .exe file, as it comes from **open**, **safe code**.

4. Explanation of Normal Expressions

The following section discusses all the regular expressions recognized by BookPal, as found in the *regex.py* file:

a) hello|hi|hey

The user's welcome message. Any of the above words, followed by any combination of words (or the blank) is recognized. Thus, phrases such as "hi" or "hello there" or "hi friend" are recognized. BookPal responds by introducing itself.

b) (.*)(thank you|thanks|(good)?bye)

The user's goodbye message. Any combination of words (or the blank) is recognized, followed by the three alternatives in the second bracket (with "good" being optional) and then any combination of words. Thus, phrases such as "okay thanks" or "great goodbye" or "thank you bye" are recognised. BookPal says goodbye wishing you a good read.

c) (.*?)(know about|know|tell me about)(the| a)?(book named| book called)? (.*)

The user's message to find a book. Any combination of words is recognized, followed by one of the alternatives in the second bracket, then "the" or "a" (but optionally), then optionally one of the options in the next bracket, and finally any string, which will be the book search key (which is why "(.*)" is placed as the contents of the last bracket are extracted). Thus, phrases such as "do you know a book named lord of the rings" or "can you tell me about the lord of the rings" or "I would like to know about lord of the rings", and many other combinations are recognized. BookPal responds that it will search for the string extracted from the sentence and then displays the search results via the API.

d) (.*)when(.*)published|(.*)publish date

Question about the date of issue. Any combination of words is recognized, followed by "when" followed by any combination of words, followed by the two alternatives about the version, followed by any string. Thus phrases such as "do you know when it was published?" or "can you tell me the publish date?". BookPal responds by giving the publication date and the publisher.

e) (.*)(what|which)?(.*) language

Question about the language of writing. Following a similar logic to the previous questions, phrases such as "do you know what language it is written in" or "do you know the language?" are identified. BookPal responds by indicating the language.

f) (.*)(give|provide|tell)(.*)(description|info|details|summary)

Question about the summary. Following a similar logic to the previous questions, phrases such as "could you provide more details" or "give me a summary" or "can you tell me some info" and many others are identified. BookPal responds by quoting the summary of the book.

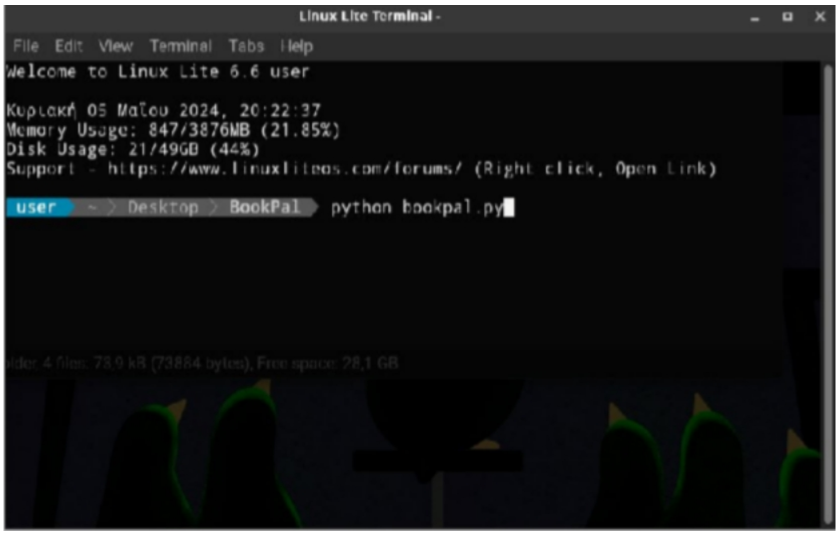
g) (.*)(buy|purchase) (link|it)

   Question about the online marketplace link. Following a similar logic to the previous questions, phrases such as "give me a buy link" or "can you tell me where I can buy it?" are identified. BookPal responds by giving the link, if it exists. If not, it indicates that it is unknown.

All of the above regular expressions ignore the existence of uppercase letters, as the *lower()* function is used in the user input. They are also designed to recognize as many sentences of the same kind as possible. Thus, from a regular expression containing many alternatives in many parentheses, all combinations of the elements involved can be derived.

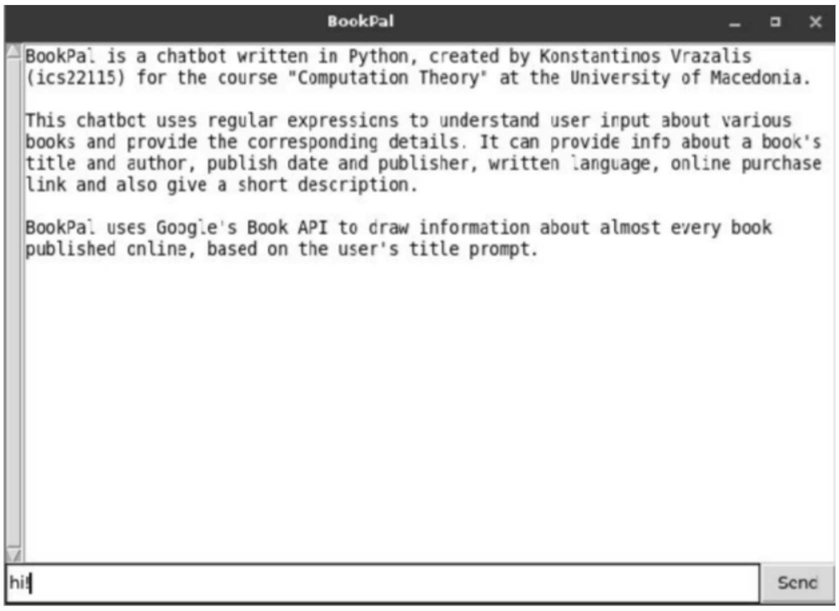
5. Demonstration of Application Functionality

This section shows snapshots of the application execution, where all possible types of queries that the user can formulate are performed, plus the case where the input is not understood. The first image captures the chosen mode of execution of the application (as mentioned in subsection 3.5, this is not the only one).



```
Welcome to Linux Lite 6.6 user

Κυριακή 05 Μαΐου 2024, 20:22:37
Memory Usage: 847/3876MB (21.85%)
Disk Usage: 21/49GB (44%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

user  ~ > Desktop > BookPal  python bookpal.py
```

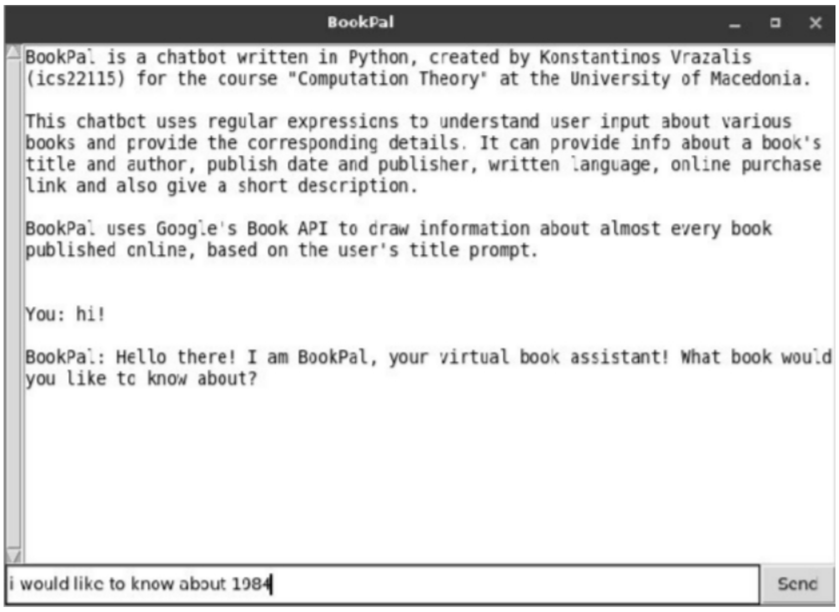
1: Running the application.



```
BookPal is a chatbot written in Python, created by Konstantinos Vrazalis
(ics22115) for the course "Computation Theory" at the University of Macedonia.

This chatbot uses regular expressions to understand user input about various
books and provide the corresponding details. It can provide info about a book's
title and author, publish date and publisher, written language, online purchase
link and also give a short description.

BookPal uses Google's Book API to draw information about almost every book
published online, based on the user's title prompt.
```

```
hi!
```

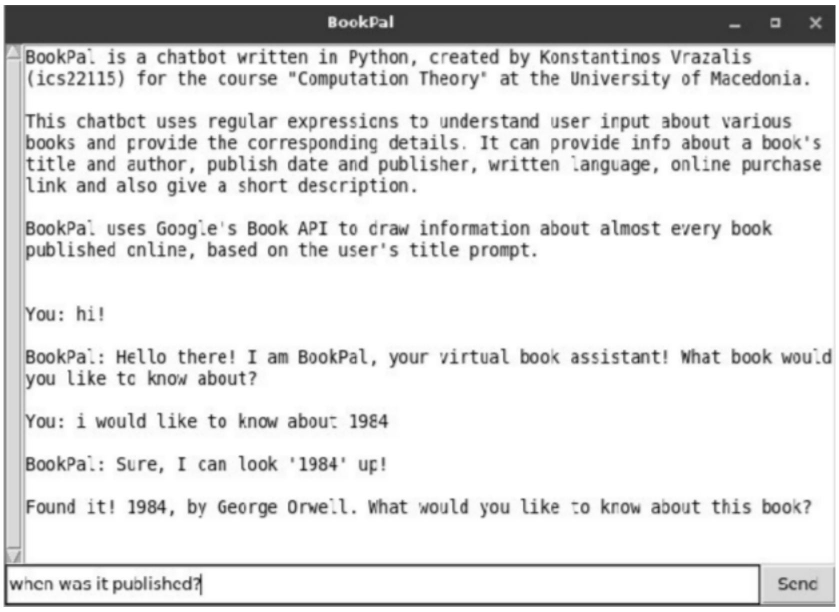
2: Greeting message.



```
BookPal is a chatbot written in Python, created by Konstantinos Vrazalis
(ics22115) for the course "Computation Theory" at the University of Macedonia.

This chatbot uses regular expressions to understand user input about various
books and provide the corresponding details. It can provide info about a book's
title and author, publish date and publisher, written language, online purchase
link and also give a short description.

BookPal uses Google's Book API to draw information about almost every book
published online, based on the user's title prompt.


You: hi!

BookPal: Hello there! I am BookPal, your virtual book assistant! What book would
you like to know about?
```

```
i would like to know about 1984
```

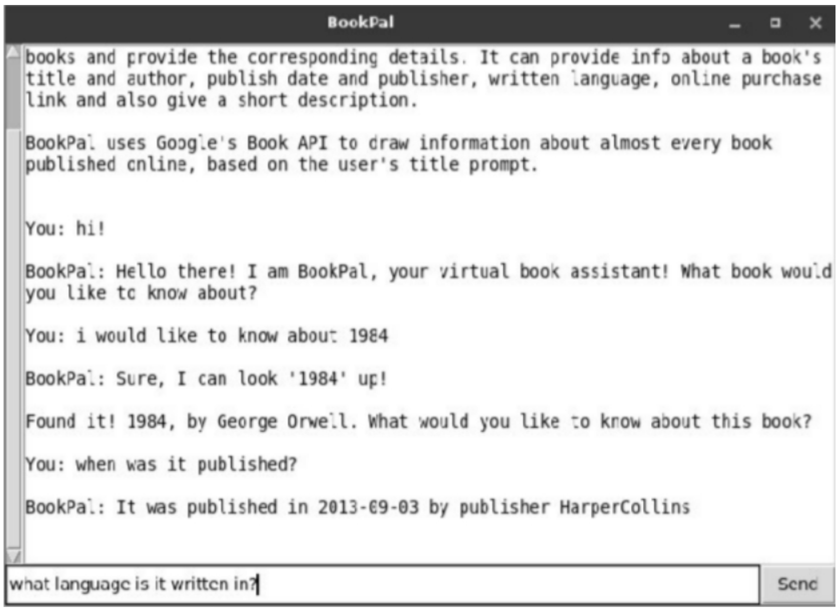
3: Search for a book.



```
BookPal is a chatbot written in Python, created by Konstantinos Vrazalis
(ics22115) for the course "Computation Theory" at the University of Macedonia.

This chatbot uses regular expressions to understand user input about various
books and provide the corresponding details. It can provide info about a book's
title and author, publish date and publisher, written language, online purchase
link and also give a short description.

BookPal uses Google's Book API to draw information about almost every book
published online, based on the user's title prompt.


You: hi!

BookPal: Hello there! I am BookPal, your virtual book assistant! What book would
you like to know about?

You: i would like to know about 1984

BookPal: Sure, I can look '1984' up!

Found it! 1984, by George Orwell. What would you like to know about this book?
```

```
when was it published?
```

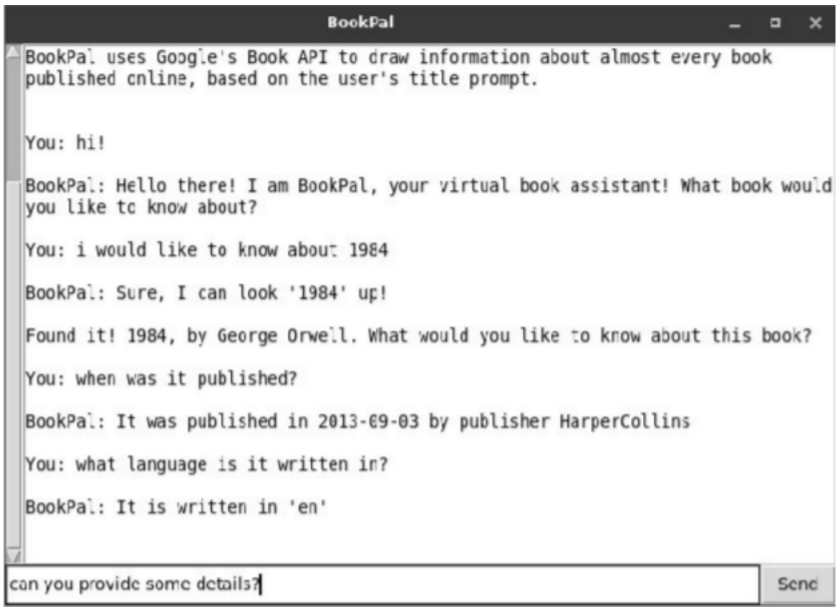
4: Question about date of issue.



```
books and provide the corresponding details. It can provide info about a book's
title and author, publish date and publisher, written language, online purchase
link and also give a short description.

BookPal uses Google's Book API to draw information about almost every book
published online, based on the user's title prompt.


You: hi!

BookPal: Hello there! I am BookPal, your virtual book assistant! What book would
you like to know about?

You: i would like to know about 1984

BookPal: Sure, I can look '1984' up!

Found it! 1984, by George Orwell. What would you like to know about this book?

You: when was it published?

BookPal: It was published in 2013-09-03 by publisher HarperCollins
```

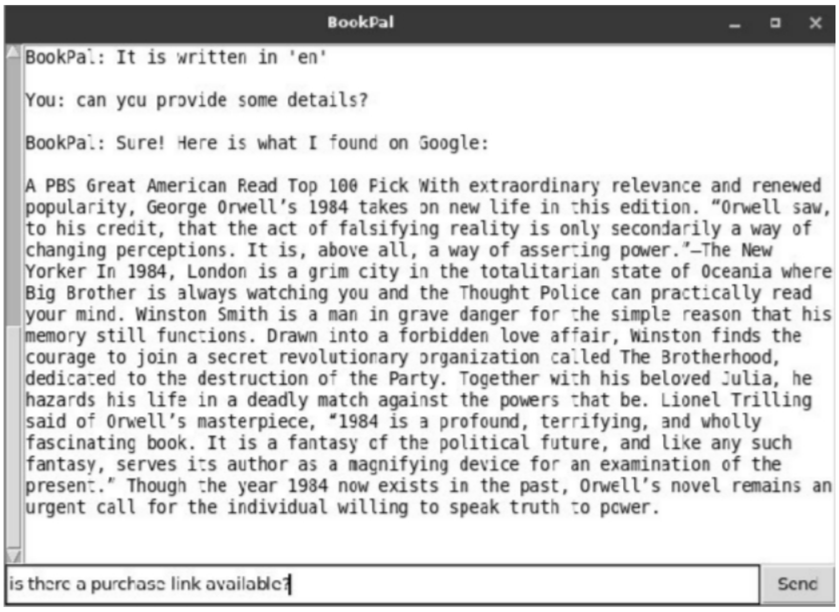
```
what language is it written in?
```

5: Question about language.



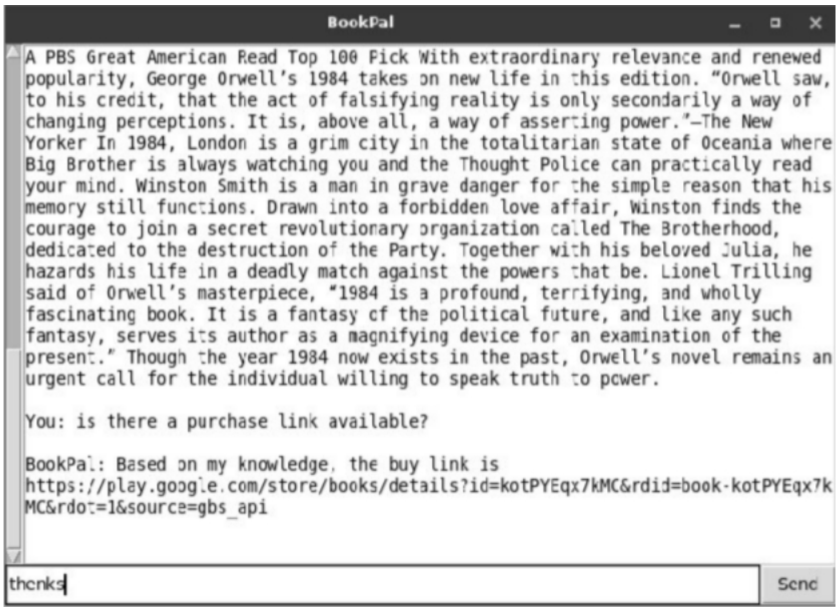
```
BookPal uses Google's Book API to draw information about almost every book
published online, based on the user's title prompt.


You: hi!

BookPal: Hello there! I am BookPal, your virtual book assistant! What book would
you like to know about?

You: i would like to know about 1984

BookPal: Sure, I can look '1984' up!

Found it! 1984, by George Orwell. What would you like to know about this book?

You: when was it published?

BookPal: It was published in 2013-09-03 by publisher HarperCollins

You: what language is it written in?

BookPal: It is written in 'en'

can you provide some details?
```

6: Question for summary.



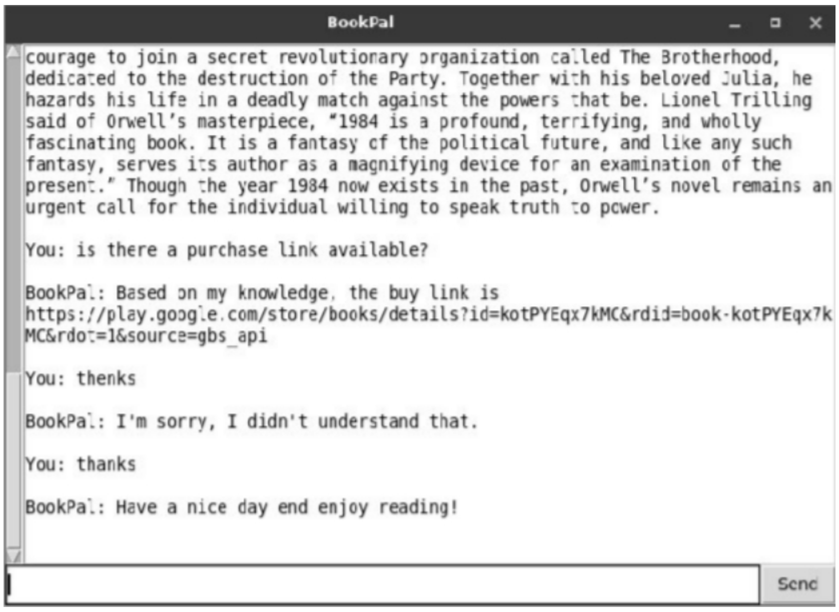
```
BookPal: It is written in 'en'

You: can you provide some details?

BookPal: Sure! Here is what I found on Google:

A PBS Great American Read Top 100 Pick With extraordinary relevance and renewed
popularity, George Orwell's 1984 takes on new life in this edition. "Orwell saw,
to his credit, that the act of falsifying reality is only secondarily a way of
changing perceptions. It is, above all, a way of asserting power."—The New
Yorker In 1984, London is a grim city in the totalitarian state of Oceania where
Big Brother is always watching you and the Thought Police can practically read
your mind. Winston Smith is a man in grave danger for the simple reason that his
memory still functions. Drawn into a forbidden love affair, Winston finds the
courage to join a secret revolutionary organization called The Brotherhood,
dedicated to the destruction of the Party. Together with his beloved Julia, he
hazards his life in a deadly match against the powers that be. Lionel Trilling
said of Orwell's masterpiece, "1984 is a profound, terrifying, and wholly
fascinating book. It is a fantasy of the political future, and like any such
fantasy, serves its author as a magnifying device for an examination of the
present." Though the year 1984 now exists in the past, Orwell's novel remains an
urgent call for the individual willing to speak truth to power.

is there a purchase link available?
```

7: Question about market link.



```
A PBS Great American Read Top 100 Pick With extraordinary relevance and renewed
popularity, George Orwell's 1984 takes on new life in this edition. "Orwell saw,
to his credit, that the act of falsifying reality is only secondarily a way of
changing perceptions. It is, above all, a way of asserting power."—The New
Yorker In 1984, London is a grim city in the totalitarian state of Oceania where
Big Brother is always watching you and the Thought Police can practically read
your mind. Winston Smith is a man in grave danger for the simple reason that his
memory still functions. Drawn into a forbidden love affair, Winston finds the
courage to join a secret revolutionary organization called The Brotherhood,
dedicated to the destruction of the Party. Together with his beloved Julia, he
hazards his life in a deadly match against the powers that be. Lionel Trilling
said of Orwell's masterpiece, "1984 is a profound, terrifying, and wholly
fascinating book. It is a fantasy of the political future, and like any such
fantasy, serves its author as a magnifying device for an examination of the
present." Though the year 1984 now exists in the past, Orwell's novel remains an
urgent call for the individual willing to speak truth to power.

You: is there a purchase link available?

BookPal: Based on my knowledge, the buy link is
https://play.google.com/store/books/details?id=kotPYEqx7kMC&rdid=book-kotPYEqx7k
MC&rdot=1&source=gbs_api

thanks
```

8: Wording of wrong message.



```
courage to join a secret revolutionary organization called The Brotherhood,
dedicated to the destruction of the Party. Together with his beloved Julia, he
hazards his life in a deadly match against the powers that be. Lionel Trilling
said of Orwell's masterpiece, "1984 is a profound, terrifying, and wholly
fascinating book. It is a fantasy of the political future, and like any such
fantasy, serves its author as a magnifying device for an examination of the
present." Though the year 1984 now exists in the past, Orwell's novel remains an
urgent call for the individual willing to speak truth to power.

You: is there a purchase link available?

BookPal: Based on my knowledge, the buy link is
https://play.google.com/store/books/details?id=kotPYEqx7kMC&rdid=book-kotPYEqx7k
MC&rdot=1&source=gbs_api

You: thenks

BookPal: I'm sorry, I didn't understand that.

You: thanks

BookPal: Have a nice day end enjoy reading!
```

9: Farewell message.

6. Conclusions

This paper was an introduction to the use of regular expressions with the Python programming language. On a secondary level, it also provided the opportunity to become familiar with the graphical interface of Python programs, and the use of APIs to retrieve information from the web. Future extension of the application could include optimization of existing or addition of new regular expressions, but also improvement of the graphical interface to make it more like a messaging environment. Finally, it was concluded that the use of APIs can significantly extend the functionality of an application to make it a truly useful tool.

7. Bibliography

1. Documentation of the library re: https://docs.python.org/3/library/re.html
2. Documentation of the tkinter library: https://docs.python.org/3/library/tkinter.html
3. Google Books API: https://developers.google.com/books
4. For the drawing of Figure 1 the software Dia was used: http://dia-installer.de/doc/index.html
5. Library documentation requests: https://pypi.org/project/requests
6. Documentation pip: https://pypi.org/project/pip
7. PyInstaller documentation: https://pyinstaller.org/en/stable
8. Application icon link: https://icon-icons.com/download/25711/ICO/512/