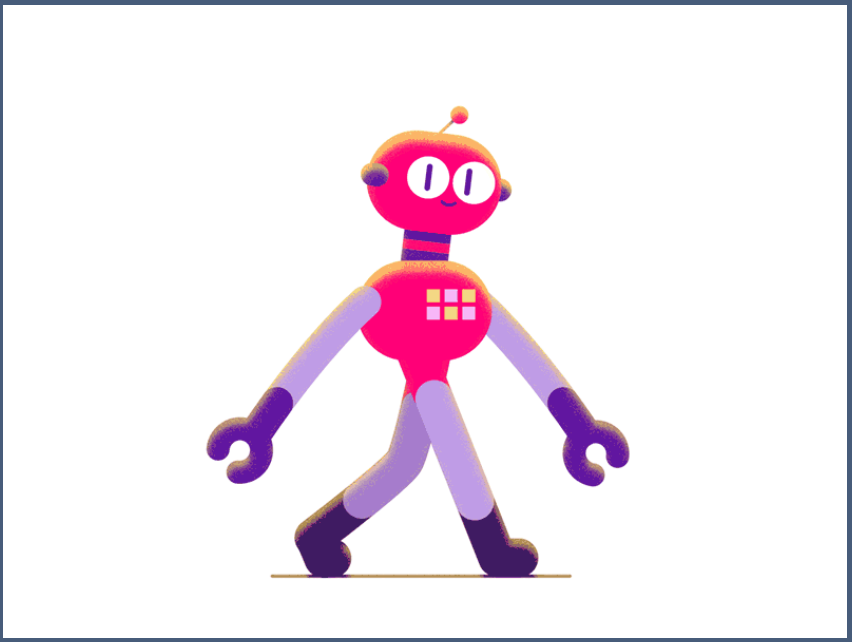
Bilbo the bot



A fancy NLP project by

@leosanchezsoler

The Bridge: Digital Talent Accelerator

March 2021

Contents

Prelude

Almost every Data Scientist has felt that sense of stress and insignificance when trying to inquire into a specific topic.

The vast amount of resources available on the net can be overwhelming for anyone who is trying to internalize complex concepts without dying in the process.

That kind of feeling marked the beginning of something that ended up in Bilbo, a recommendation engine fuelled with thousands of Data Science articles from prominent academic institutions from all around the globe.

Thanks to Python, Amazon Web Services (AWS) and a slight grasp of Natural Language Processing (NLP) it's been possible to bring Bilbo to life, feeding it, and deploying it to production.

In the following pages, the whole process will be analyzed in detail: from the tools used, to the evaluation of the project.

ONLY THING TO DO! ON WE GO!

- Bilbo Baggins -

Goals

The main goal was to display all the accumulated expertise achieved from the past months in order to create a useful and innovative tool from scratch.

In order to achieve it, it was mandatory to tackle the task with a holistic perspective in which every step had to be in harmony with the rest. That's the reason why *Trello* played such an important role within the first days.

Knowledge is at the core of Bilbo the bot, and just being able to shed light to aspiring Data Scientists can be considered as a success.

Although this project has been very rewarding and passed with merits, it is true that not deploying it into a message app as Telegram or Discord tarnishes the final outcome. Nevertheless, these features will be included in the second phase.

Specifications

Hardware

Almost every computer can run this program, even though some devices may last more than expected.

Here are the details of the device used for the project:

- **Model name:** ASUS Rog Strix G513QR_G513QR
- **Processor**: AMD Ryzen 7 5800H with Radeon Graphics, 3201
- **Memory:** 1 TB SSD
- **RAM Memory:** 16 GB
- **Graphic Card:** Nvidia GeForce RTX 3070 Laptop GPU

Software

Python 3.8.8. has been the main software used.

Requirements

To execute the program, all tools and datasets are inside the program, in order to make things as easy as possible for the user.

Steps

1. Research the context

In this particular case, there was no need to do an exhaustive research, as recommendation engines documentation is easy to find on the Internet.

2. Get the data

The data was obtained from this Github repository: https://github.com/NeelShah18/arxivData. The dataset contains thousands of Data Science papers that were collected using web scraping.

Despite the fact that some variables were not properly formatted, after inspecting its structure it was considered valid for training a model.

3. Data Wrangling

For this project, Data Wrangling took some time to be done. As the info was collected using web scraping, most of them still had the html structure. In order to structure the data, it was necessary to iterate through dictionaries and extract the relevant tags.

It was also curious that some columns looked like lists at first sight, but were classified as objects.

4. Data Mining/Data Cleaning

The main dataset did not suffer much Data Mining due to its structure. Removing some unnecessary columns was the only change applied.

5. Text preprocessing

Text formatting had been one of the major issues within the project. NLP implies taking many steps in order to obtain a proper text format. The preprocessing phase steps were as follows:

- Remove stopwords: use Natural Language Toolkit (nltk) to drop stopwords. By doing this, the source text only contains relevant words for better performance while training.
- Tokenize: assign a unique id for each word in the text.
- Encode labels: Bilbo is taught to suggest articles based on tags.
- Embedding.

6. Training the model

Training was quite an arduous task, because of the computational demand for fitting the model. However, the wait was worthwhile. After hours of training, the net was ready to predict.

7. Deploying to production

Finally, the time to deploy the model arrived. By using Flask it was possible to create a site in which users could learn more about the

project, get the cleaned data and last but not least, search for Data Science articles by typing a sentence.

After the model was tested locally and proved to perform as expected, it was deployed to an AWS remote server.

Since then, Bilbo was ready to accompany aspiring Data Scientists in their journey.

Conclusions

Once the project was finished, some mandatory questions were formulated.

Code conclusions

The idea of dealing with so many different areas of coding has been a major force against the deadlines. In spite of all the difficulties that came across the project, it's been possible to tackle every aspect of the process.

Training a neural network was the main goal that fuelled every effort and every extra hour working on bringing Bilbo to life.

Personal conclusions

As it happened with the previous projects made during the time in The Bridge, they always teach something new and push students to the next level. With such a level of exigency, it encourages them to do their best and create vast projects from scratch, exceeding previous expectations most of the time.

Sources

Data Source

- NeelShah18/arxivData
  - https://github.com/NeelShah18/arxivData/blob/master/arxivData.json

Solve code doubts

- Stack Overflow
  - https://stackoverflow.com/
- Tensorflow Documentation
  - https://www.tensorflow.org/guide
- Github
  - https://www.github.com/
- Flask Documentation
  - https://flask.palletsprojects.com/en/1.1.x/
- Stackexchange
  - https://stackexchange.com/

Organization Tools

- Trello
  - https://trello.com/
- Creately
  - https://creately.com/
- Google Drive

    - https://drive.google.com/drive/
- Discord
    - https://discord.com/

Git repository

- Github
    - https://github.com/