

Image Inquiry Simulation for PET/FIRST Test Exam

Stefania Zanetta (S4463362)

zanettastefania@outlook.it

1: Introduction

In this report it will be presented a chatbot to simulate an oral section for a PET/FIRST exam. A PET/FIRST oral exam is divided in many parts, one of which is described as “Describe one colour photograph, talking for about 1 minute”¹. This chatbot is meant to be used as a didactic tool to practice for this part of the exam. The main goal is to use object detection and caption generation with NLP tools to have a conversation about contents of images, and give feedback for possible grammatical mistakes.

2: Architecture

The chatbot’s architecture consists of different functional parts used for different purposes, put together to converse with the user. Given the educational purposes of the chatbot, conversational flow and responses are handled in a rule-based logic, to maintain control over all outputs provided to the user. The architecture consists of the following components:

- *Object Detection*: The goal of object detection is to detect all instances of objects from one or several known classes, such as people, cars, or faces in an image.² For this task I used YOLOv8, a model for object detection and image segmentation known for its accuracy and speed.³ In this project only its object detection functions will be used.
- *Image Captioning*: The goal of image captioning is to describe the content of an image using properly formed English sentences.⁴ In this project I used ViT-GPT-2 image captioning model, a Transformer-based model that combines Vision Transformer (ViT) as encoder and Generative Pretrained Transformer 2 (GPT-2) as a decoder trained on COCO dataset.⁵ The model used in the project was chosen for its speed and accuracy.
- *Grammar Check*: The goal is to find and correct any grammatical and/or typographical errors. For this project I decided to use Gramformer, being easy to use and for its speed. Although more official options like LanguageTool are

¹ Cambridge English. at: <https://www.cambridgeenglish.org/exams-and-tests/preliminary/exam-format/>

² Amit, Y., Felzenszwalb, P., Girshick, R. (2021). Object Detection. In: Ikeuchi, K. (eds) Computer Vision. Springer, Cham. https://doi.org/10.1007/978-3-030-63416-2_660

³ Sohan, Mupparaju & Ram, Thotakura & Ch, Venkata. (2024). A Review on YOLOv8 and Its Advancements. 10.1007/978-981-99-7962-2_39.

⁴ Vinyals, O. *et al.* (2015) *Show and tell: A neural image caption generator*, *arXiv.org*. Available at: <https://arxiv.org/abs/1411.4555>

⁵ Transformative Fusion: Vision Transformers and GPT-2 Unleashing New Frontiers in Image Captioning within Image Processing. (2023). International Journal of Innovative Research in Engineering & Management

available, Gramformer was chosen for its simplicity and speed, as the chatbot requires fast response times.

- *POS Tagging and Tokenization*: The goal of POS tagging and tokenization is to break down and analyse the text, identifying the grammatical parts of speech and segmenting the text into manageable units. For this project, I used the NLTK (Natural Language Toolkit) library for POS tagging and tokenization, and WordNet to obtain synonyms for further analysis of the user's responses.

3: Functions

After reviewing the components of the chatbot, this section will explain how each function and their role in the conversation. Due to the large number of functions, they have been divided into different subsections for clarity.

3.1: Image Loading and Processing

At the start of every conversation, the chatbot will ask which image the user would like to talk about. Once the user confirms the selected image, before asking any questions, the chatbot will process it to detect objects and get a caption:

- *is_url(string)*: To accommodate the user, the chatbot accepts both images in their device and images from the web. This function, given a path for the image the user wants to use simply returns TRUE if the path given is an url, FALSE otherwise.
- *load_image(image_path)*: After finding the type of path image passed, with the previous function *is_url*, the function will load and return the image. If the image isn't found returns FALSE.
- *show_image(image)*: Once the image is loaded, it is displayed to the user to see.
- *setup_yolo()*: This function sets up and returns the model to use for the object detection, as mentioned above, for this project I used YOLOv8. Many versions of model can be found, and mainly differ in accuracy, number of parameters and speed. The model chosen, yolov8x.pt, is the most accurate.⁶
- *detect_yolo(image, model)*: This function takes an image and the object detection model and returns the list of names and identification codes of objects found in the image.
- *get_caption(model, image_processor, tokenizer, image)*: Given pre-configured tokenizer, image processors and model, the function generates the caption. The image is processed, and the output is decoded by the tokenizer to produce a caption as a simple string.

⁶ Ultralytics (2024) YOLOv8, Ultralytics YOLO Docs Available at: <https://docs.ultralytics.com/it/models/yolov8/#performance-metrics>

3.2: User Input Interpretation

After the user provides the name of the image they want to discuss, the chatbot needs to understand their replies. The following functions are used to interpret the user inputs:

- *classify_input(input_str, start)*: This function is used to interpret the user's response, when asked a yes or no question. In addition to the user input, the function requires another argument, a Boolean *start*, which is used to differentiate the output message of the chatbot based on the part of the conversation where the function is called. The function returns TRUE if the user confirms the question, FALSE otherwise.
- *extract_number_from_text(text)*: This function extracts numbers from a sentence. Both simple numbers and non-specific quantities (eg. "few", "several") are converted and returned as numeric values.
- *pos_to_wordnet_pos(penntag, returnNone=False)*: This function converts a part of speech tag into the corresponding WordNet part of speech tag.
- *get_synonyms(word)*: This function returns all synonyms from WordNet based on the word and its part of speech.

3.3: Question Generation and Evaluation

After the image is selected and processed, the chatbot will begin asking questions and analysing the user's replies. There are different functions that will handle this part of the conversation:

- *generate_questions(detected_list)*: This function manages the creation of the first three questions. Passing the list of detected objects from the image, a random object from the list is selected to generate a question. For the remaining two questions, objects not present in the image are chosen. The function returns a list of questions, along with a Boolean indicating if the question is about a real object in the image and the corresponding identification code of the detected list.
- *check_answer(question_tuple, user_input, detected_list)*: This function checks user's answer to the first three questions, and determines the chatbot's response. It requires three different arguments: the question with all needed information, as described in the previous function, with the Boolean and the element of the detected list, the input of the user and the list of detected objects.
First, the function extracts any number given in the reply, using the function *extract_number_from_text*. Then from the list of detected objects, it counts the actual number of the object of the question present in the image. Based on whether the question is about a real object in the image or not, the function compares the user's answer to the actual count, determining its correctness. Along with the chatbot response, a Boolean is also returned to indicate if the reply was correct.
- *calculate_score(questions_list, user_answers, detected_list, feedbacks, final_answer, corrected_answers, desc)*: After the user answers all questions, this function evaluates the responses given, counting grammatical errors and incorrect

answers, to give a final mark to the user. The first three questions and the final question are graded differently; the final mark will be an average between the two results.

For the first three questions each answer can reach a maximum of five points; three points are given if the answer is correct and two additional points can be awarded based on the length of the reply. For each grammatical error detected one point is deducted. At the end of the evaluation the mark is normalized to the range of 0 to 10.

The evaluation of the final answer is calculated on similar parameters. A longer description, more than five words, starts from a base of six points, while a shorter one starts from four points. For each word that matches the objects detected, the description generated by the model or one of its synonyms, one point is added to the score. In the end, half a point is deducted for each grammatical error found. The function outputs the results and returns the average of the two marks.

- *questions(user_input, detected_list, desc)*: This function is responsible for handling input and output of all the questions. In addition to calling the other functions in this section, it checks if the user would like to reset the chatbot, with the input “stop” or quit the conversation, with the input “quit”.

It also calls for the function *gf.correct* of the previously mentioned library called Gramformer, to store the corrected versions of the response given by the user.

After gathering the first three answers, it asks one last question equal for all images, and not created by the function *generate_questions*, for a description of the image submitted.

In the end, after the final score is provided, the user will be given the opportunity to quit the chatbot or start a new conversation.

3.4: Chatbot

ChatBot() is the main function that needs to be executed to start the chatbot. As mentioned in previous sections, the chatbot can be restarted by typing “stop” or closed by typing “quit” at any time.

When the chatbot starts, the first question asks the user to specify the image for the conversation. After receiving the image from *load_image*, the function displays the image for the user to see, with *show_image*, and ask for confirmation, offering the possibility to change image if necessary. The image remains displayed for the rest of the conversation for the user to check and to help the user replying to the questions.

After the user confirms the selected image, it is first processed by the image captioning model *get_caption* then, by the object detection model with *detect_yolo*.

Once both operations are complete, it calls the function *questions* with the list of detected objects and image description.

When the conversation ends, the *questions* function returns either “quit” or “stop”. Based on the output, the chatbot then either will stop or reset the conversation asking for another image from the user.

4: Conversation Flow

In this section, an example of conversation is presented, showing different responses given by the chatbot and how replies by user are handled.

When the chatbot is started, the user will be asked to specify the image they would like to use for the conversation. Further information is provided about the types of images that the chatbot accepts as an input.

```
-----[New Chat]-----
[Stop chatbot: 'quit'] ChatBot: Hello! Let's start! Which image would you like to talk about?
[Stop chatbot: 'quit'] ChatBot: You can type the path of images in you device eg. image.jpg
[Stop chatbot: 'quit'] ChatBot:Or you can paste url of image on the web eg. https://example.website.com/image.jpg
You: 
```

If given wrong paths or broken URLs, the chatbot will inform that the image wasn't found and will ask again for another image.

```
You: non_existing_image.png
[Stop chatbot: 'quit'] ChatBot: Sorry I could not find the image: non_existing_image.png
[Stop chatbot: 'quit'] ChatBot: Which image would you like to talk about?
You: 
```

Once the image is successfully found, it will be displayed, and the chatbot will ask confirmation before continuing the conversation. If the user gives a negative response, the chatbot will ask for the image path again.

```
[Stop chatbot: 'quit'] ChatBot: Which image would you like to talk about?
You: https://images.pexels.com/photos/927022/pexels-photo-927022.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500
```



```
[Stop chatbot: 'quit'] ChatBot: Is this the image you would like to talk about?
You: 
```

In case of confirmation, the chatbot begin processing the image. This process could take some time. During the wait, the user will receive a writing tip. The message not only aims to motivate the user, but also hints the user on how to take a good mark, since longer answers can earn additional points. Once processing is complete, the chatbot will ask for the first question.

```

You: yes
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Tip: Try to write as much as possible! Practicing is the key to progress!
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Ok! Let's start with some questions!
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Are there any sheep in the image? If so, how many sheep are in the image?
You: 

```

For each question reply the chatbot will give feedback. In case any grammatical errors are found, a corrected version of the answer will be given. The following image shows the response for answers with grammatical errors, incorrect responses and correct responses without errors. It can be noticed that only answers with grammatical errors receive two types of feedback: one for grammatical mistakes and another indicating whether the question was answered correctly.

```

[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Are there any sheep in the image? If so, how many sheep are in the image?
You: No there eren't any sheeps
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Corrected sentence: {"no there aren't any sheep."}
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Correct! There are no sheep in the image.
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Are there any laptops in the image? If so, how many laptops are in the image?
You: no there aren't
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Incorrect. There are 3 laptops in the image.
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Are there any vases in the image? If so, how many vases are in the image?
You: No there aren't any vases
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Correct! There are no vases in the image.

```

Once the first three questions have been answered, the chatbot will ask for a description of the image as final question before giving an evaluation. As communicated by the chatbot the description given should be less than 240 characters, if given more any excess will be cut off and not counted toward the score.

After the user response their final question, the chatbot will first give feedback on any grammatical errors found, then it will print the marks for all the replies given.

```

[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Could you describe what's happening in the picture? [max 240 number of characters]
You: a man working in an office. He is holding a book in his hand and he is sitting on a chair
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Your mark for the answering the questions: 5.3/10
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Your mark for the description is: 10/10
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Your final mark is: 7.7/10
[Change image: 'stop' -> Stop chatbot: 'quit'] ChatBot: Would you like to analyze another image?
You: 

```

In the example provided, points were deducted from the first three questions, since there were both grammatical errors and an incorrect response. For the description, no errors were found, and the detailed answer earned additional points.

At this point the user will be asked if they want to discuss about another image. If confirmed, the chatbot will restart. As it can be noticed in all chatbot messages, the user can quit or restart the conversation at any time.

5: Conclusion

In this project, a chatbot was developed to help users practicing conversations about images, basing this system on a simulation of an oral exam for the PET/FIRST. By integrating different areas of computer vision, with object detection model YOLOv8 and an image captioning model ViT-GPT-2, the chatbot is able to extract information from the image and start a conversation with the user. Moreover, the use of grammar-checking libraries, such as Gramformer, allows the chatbot to provide feedback on grammatical errors and misspellings.

However, the system could benefit from some adjustments. One being the previously mentioned Gramformer. While being lightweight and giving quick responses, it could benefit from fine tuning to improve the accuracy. Alternatively, a rule-based approach for grammatical correction could offer more control over the quality of the output.

Another potential improvement involves the chatbot's ability to extract context from the user's responses, especially in the final question. Name Entity Recognition could be used to better understand the user's description and ensure that the answer is coherent with the image content.

Lastly, although not directly related to Natural Language Processing, using a unified model, for both object detection and image captioning, could optimize computational efficiency and produce more coherent outputs. For example, it is common for some objects detected by YOLOv8 to not be mentioned in the generated caption. A unified model could address this issue and reduce waiting time for the user.

References

- [1] Cambridge English. at: <https://www.cambridgeenglish.org/exams-and-tests/preliminary/exam-format/>
- [2] Amit, Y., Felzenszwalb, P., Girshick, R. (2021). Object Detection. In: Ikeuchi, K. (eds) Computer Vision. Springer, Cham. https://doi.org/10.1007/978-3-030-63416-2_660
- [3] Sohan, Mupparaju & Ram, Thotakura & Ch, Venkata. (2024). A Review on YOLOv8 and Its Advancements. 10.1007/978-981-99-7962-2_39.
- [4] Vinyals, O. et al. (2015) Show and tell: A neural image caption generator, arXiv.org. Available at: <https://arxiv.org/abs/1411.4555>
- [5] Transformative Fusion: Vision Transformers and GPT-2 Unleashing New Frontiers in Image Captioning within Image Processing. (2023). International Journal of Innovative Research in Engineering & Management
- [6] Ultralytics (2024) YOLOv8, Ultralytics YOLO Docs. Available at: <https://docs.ultralytics.com/it/models/yolov8/#performance-metrics>