

Maintenance Manual

Evaluating GPT-3.5-turbo for Action Item Extraction in Meeting Transcripts

Atanas Komsijski

April 26, 2024

1 Installing the dependencies

Install the dependencies needed to execute the code by navigating to the main folder in Terminal and running the following command:

```
pip install -r requirements.txt
```

2 Hardware requirements:

- internet connection to use OpenAI's API
- 64-bit processor
- 4GB of RAM or higher

3 Software requirements

- Python 3.8 or higher
- the dependencies included in `requirements.txt`
 - ipython 8.21.0
 - jupyter_core 5.7.1
 - matplotlib 3.8.3
 - sentence-transformers 2.6.1
 - tiktoken 0.6.0
 - pandas 2.2.1
 - etc.

4 Installation

As the code is presented in Jupyter notebooks it does not require any specific installation beyond the initial dependencies as specified above. Those notebooks can be viewed using either VSCode and the Jupyter add-on or the basic Python distribution of `jupyter`, which can be installed using the pip package manager by running `pip install jupyter`. And executing the notebooks by navigating to the folder and running:





```
jupyter notebook notebook.ipynb
```


5 Directory structure

```
| README.txt
| requirements.txt
| MRTtoTXT.ipynb
| Chat.ipynb
| GPT_action_items_v1.xml
| GPT_action_items_v2.xml
| GPT_action_items_v3.xml
| Human_action_items.xml
| ROUGE_score.ipynb
| BERT_score.ipynb
| Sentence_pairing_with_BERTScore_Not_Used.ipynb
|
----ICSI_original_transcripts
|   index.html
|
|   ----doc
|   |       all.blp
|   |       file.tbl
|   |       icsi1.spk
|   |       Meeting1_annotated.dtd
|   |       Meeting1_annotated_dtd.html
|   |       naming.html
|   |       naming.txt
|   |       overview.txt
|   |       seatingchart.txt
|   |       trans_guide.txt
|   |
|   ----transcripts
|   |       75 .mrt files
|   |
|   |   ----chosen_transcripts
|   |   |       25 .txt files
|   |   |
|   |   ----txt_transcripts
|   |   |       75 .txt files
```





6 List of source code files and their role

Jupyter Notebooks:

 MRTtoTXT.ipynb	Converts the .mrt files provided in the ICSI dataset to human-readable .txt files.
 Chat.ipynb	Performs the long text chunked linear segmentation and feeds the transcript into the GPT model.
 ROUGE_score.ipynb	Computes the ROUGE scores between the GPT-generated action items and the human reference.
 BERT_score.ipynb	Computes the BERTScore between the GPT-generated action items and the human reference. Also contains standard deviation calculations for the study of consistency between iterations of the automatically generated action item lists.

The  Sentence_pairing_with_BERTScore_Not_Used.ipynb file is also provided, however it is not used in our evaluation. It contains two versions of an attempt at creating a metric using a more granular approach matching sentences instead of words using an approach similar to BERTScore to calculate Precision, Recall and F1. It is solely provided for potential future development and to show the attempted technique, although it did not prove reliable.

XML files:

 GPT_action_items_v1.xml	Contains the action items extracted by the GPT model for prompt version 1.
 GPT_action_items_v2.xml	Contains the action items extracted by the GPT model for prompt version 2.
 GPT_action_items_v3.xml	Contains the action items extracted by the GPT model for prompt version 3.
 Human_action_items.xml	Contains the ground truth list of action items; the combined list of all lists extracted by our annotators.

7 File Pathnames

Already pre-processed into .txt files meeting transcripts are located in the dataset's folder:





\ICSI_original_transcripts\transcripts\txt_transcripts.

The subset of 25 transcripts we chose to use in this study can be found at:

\ICSI_original_transcripts\transcripts\chosen_transcripts

The XML files containing the GPT-generated action items lists and the XML file containing the human-annotated action item lists can be found directly in the base directory of the folder:

❏ komsijski_atanas

-  GPT_action_items_v1.xml
-  GPT_action_items_v2.xml
-  GPT_action_items_v3.xml
-  Human_action_items.xml

8 Main methods

<pre>def MRTtoTXT(xml_file)</pre>	Reads an <code>.mrt</code> file and cleans up the additional XML-format information, only keeping the speaker and their uninterrupted speech segment. Writes the cleaned up, human-readable content to a <code>.txt</code> file.
<pre>def process_folder(input_folder, output_folder)</pre>	Takes as parameters an input folder and an output folder, used to process the whole folder of <code>.mrt</code> meeting files from the ICSI corpus into <code>.txt</code> files.
<pre>def num_tokens_from_string(string, encoding_name: str)</pre>	Accepts a string and an encoding name, returns the number of tokens contained within the string.
<pre>def split_into_chunks(text)</pre>	The function takes in a text to be segmented and outputs a list of segments, applying linear chunked segmentation.
<pre>def extract_action_items (list_of_chunks, temperature, max_tokens)</pre>	Takes in the list of chunks generated by the <code>def split_into_chunks(text)</code> function and makes an API call to the OpenAI GPT model, passing it the instruction to extract action items and each chunk of text one after the other.
<pre>def process_text_file(file_path, file_name, temperatures: dict, max_tokens: int)</pre>	Function iterating over the list of files we want to process, constructing the XML output files and cleaning up the GPT output.
<pre>def read_xml(file_path)</pre>	Function that reads the contents of our XML files and saves it in a nested list for easier access.
<pre>def compute_rouge(gpt_meetings, human_meetings)</pre>	Computes the ROUGE score (n-gram overlap) between the GPT's output and the human reference. Returns a dictionary of the scores.
<pre>def compute_bert_score (gpt_meetings, human_meetings)</pre>	Computes the BERTScore metrics between the GPT's output and the human reference. Returns a dictionary of the scores.

9 Direction for future improvements

The attempted development of a scoring metric inspired by BERTScore, located in the `Sentence_pairing_with_BERTScore_Not_Used.ipynb` file may be explored further. Another future improvement that can be implemented is potentially creating a visual interface for the GPT action item extraction or developing an online application using our functions to allow people to automatically extract action items from their meeting transcripts.