Task: --/02/2024

1. Text Summarizer
2. Speech-to-Text
3. Text-to-Speech
4. Translator
5. File converter

--------------------------------------------------------------------------

Progress:

1. Text Summarizer:
   * Type 1: Using nltk
   * Type 2: Using Hugging face model – “facebook/bart-large-cnn”

Both .py files are working, however the result of the summarization differs between the two cases. Depending on the user’s discretion one or both may be flagged unfit for use for official cases.

Example use cases, for the same input text:

* Type 1 output:

‘’’New York (CNN)When Liana Barrientos was 23 years old, she got married in Westchester County, New York. A year later, she got married again in Westchester County, but to a different man and without divorcing her first husband. Barrientos, now 39, is facing two criminal counts of "offering a false instrument for filing in the first degree," referring to her false statements on the

2010 marriage license application, according to court documents. After leaving court, Barrientos was arrested and charged with theft of service and criminal trespass for allegedly sneaking into the New York subway through an emergency exit, said Detective

Annette Markowski, a police spokeswoman. In total, Barrientos has been married 10 times, with nine of her marriages occurring between 1999 and 2002. Prosecutors said the immigration scam involved some of her husbands, who filed for permanent residence status shortly after the marriages.’’’

* Type 2 output:

'Liana Barrientos, 39, is charged with two counts of "offering a false instrument for filing in the first degree" In total, she has been married 10 times, with nine of her marriages occurring between 1999 and 2002. She is believed to still be married to four men.

1. Speech-to-Text:

By using an available library (i.e, speech\_recognition) and inbuilt functions of the library, the .py file works as one would expect.

Use case to shown by executing the .py file.

1. Text-to-Speech:

* Google’s text to speech lib has been used.
* Input is taken in any language.
* Output can be in any accent from the list of supported languages displayed upon executing the .py file
* NOTE: If the input text is in Japanese, the accent Japanese (ja) must be selected, otherwise the audio file produced will be gibberish.
* Similar cases may happen for certain languages. All cases have yet to be tested.

1. Translator:

* Made use of library – googletrans: translating and langdetect: detecting input language
* Result of translation may be inconsistent
* Subsequent translations from one text to the other may result in the loss of meaning.

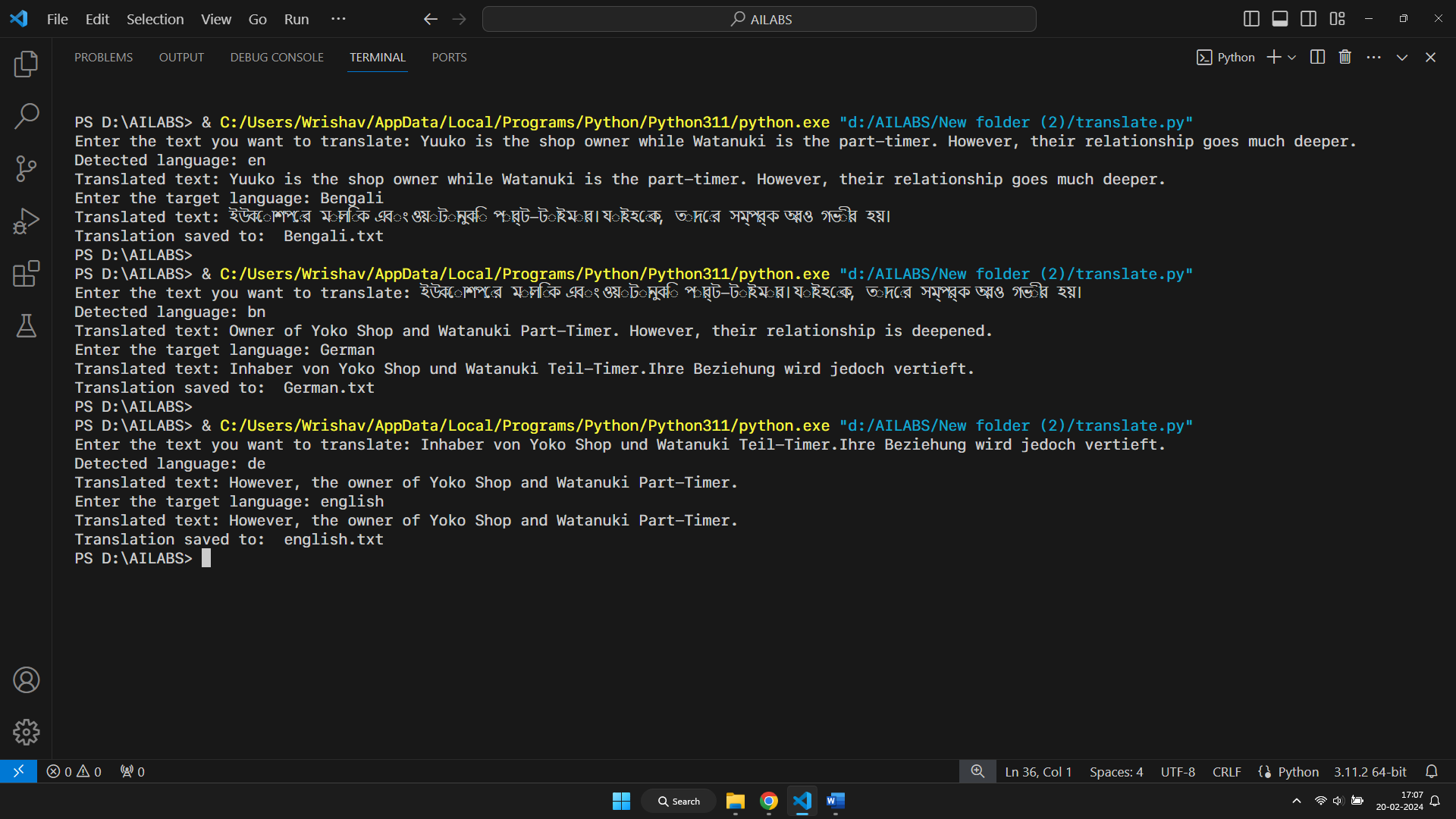
Example: **“Yuuko is the shop owner while Watanuki is the part-timer. However, their relationship goes much deeper.”**

Convert 1: to Bengali

Convert 2: from Bengali to German

Convert 3: from German to English

Final output: “However, the owner of Yoko Shop and Watanuki Part-Timer.”

For this example, neither the content nor the context is preserved.

While using google translate however, not only is the meaning, but also the original sentence is preserved.

1. File conversion:

NOTE: All observations are for my personal use case and for the individual libraries that have been used. May vary depending on modifications made to the .py executable.

* 1. DOCX:
     1. TXT: Bullets have not been preserved
     2. PDF: Formatting except new line has not been preserved

Test: Check for font colours, size, bold.

* 1. TXT:
     1. DOCX: The original data has been preserved
     2. PDF: Original formatting seems to preserved
  2. PDF:
     1. DOCX: Font style & font size seem to be compromised. Other changes if any yet to be observed
     2. TXT: Same as the case above. Formatting is lost, text is preserved as it is
  3. XLSX
     1. CSV: Works perfectly and preserves all cell contents. Whether or not text formatting such as ‘bold’, ‘italics’, or the likes are preserved is yet to been observed.
     2. TXT: Perfectly converted while preserving all data with perfect tab spaces to separate values.
     3. PDF: Data is preserved, text formatting however needs to be deliberated on. (size, bold, italics, etc)
  4. CSV:
     1. XLSX: All formatting is preserved, no errors observed
     2. PDF: Data is preserved, formatting however needs to be deliberated on.
     3. TXT: Perfectly converted while preserving all data with perfect tab spaces to separate values.
  5. TXT:
     1. XLSX: WIP
     2. CSV: WIP
     3. PDF: WIP
  6. WIP