Projekt semestralny: Programowanie w języku Python II

Temat: Aplikacja do automatycznego generowania podsumowań tekstów.

**RAPORT**

Wstęp – modele językowe

Modele językowe to rodzaj sieci neuronowych (inspirowanych ludzkim mózgiem), które opierają się na algorytmach głębokiego uczenia i mogą wykonywać szereg zadań związanych z przetwarzaniem języka naturalnego. Każdy model językowy wymaga tzw. wytrenowania na dużym zbiorze danych – czyli zapisu wzorców i struktur językowych. Odpowiednio wytrenowany model językowy będzie potrafił skutecznie przetworzyć kontekst logiczny zapisu ludzkiej mowy na interpretowalne relacje logiczne między obiektami oraz dokonywać predykcji językowej, czyli rozumieć pełny kontekst.

Modele językowe mogą służyć do generowania tekstów na dowolny temat, tłumaczeń, streszczania, kategoryzacji tekstów czy rozpoznawania nastroju tekstu (np. tzw. mowy nienawiści).

Przed przystąpieniem do realizacji projektu wykonano research odnoszący się do informacji o dostępnych metodach realizacji zadania i możliwych sposobach osiągnięcia założonego celu. Wybrano drogę wykorzystania dostępnych modeli językowych, dla których najważniejsze repozytorium stanowi strona: <https://huggingface.co/>

Dla osiągnięcia najlepszych rezultatów przetestowano dziesiątki modeli językowych opartych na : **BART** (Bidirectional and Auto-Regressive Transformers), który został udostępniony przez FACEBOOK AI , **BERT** (Bidirectional Encoder Representations from Transformers), **T5** (Text-To-Text Transfer Transformer), który został udostępniony przez Google , **PEGASUS** , **GPT-2** , czy **GPT-3** (Generative Pre-trained Transformer).

Ostatecznie wybór padł na model oparty na BART : slauw87/bart\_summarisation. Model ten, co prawda, nie radzi sobie z tekstami w języku polskim, jednak lepiej niż inne dokonuje podsumowań w języku angielskim. Dlatego zmieniono pierwotne założenia i przyjęto że tekst wejściowy dla aplikacji powinien być właśnie w języku angielskim.

Następnie podjęto decyzję o rozszerzeniu funkcjonalności tej aplikacji dla lepszego poznania tematyki modeli językowych – poprzez stworzenie modelu i wykonaniu jego wytrenowania w oparciu o zbiór danych w języku polskim.

Cel numer dwa, jaki postawiono w celach ćwiczeniowych to dokonanie kategoryzacji, czyli przyporządkowania tekstu do kategorii. Stworzono zatem pokaźny zbiór tekstów (w oparciu o treści książek, artykułów prasowych, własną inwencję twórczą, czy źródła takie jak wikipedia), który jako moduł zaimportowany do mojego skryptu – posłużyć miał do wytrenowania nowego modelu językowego. Ten cel udało się zrealizować.

Schemat działania aplikacji

1. Wykonano projekt aplikacji zgodnie z tematem.
2. Aplikacja działa według następującego schematu:
   1. Import potrzebnych bibliotek i modułów
   2. Definicja funkcji, które zostaną użyte: **summary** (w oparciu o metodę *pipeline* z biblioteki **transformers**) , **translate** (tłumaczenie tekstu na potrzeby rozszerzenia funkcjonalności aplikacji) w oparciu o metodę *translator* z biblioteki **googletrans** oraz innych funkcji związanych z użyciem GUI aplikacji
   3. W oparciu o bibliotekę **scikit-learn** stworzony został nowy model, który wytrenowano na danych w języku polskim (funkcja train\_test\_split )
   4. Stworzono funkcję *classify\_text* , która w oparciu o nowy model (pipeline z biblioteki scikit-learn) dokonuje predykcji językowej i klasyfikuje tekst.
   5. Stworzono funkcję, która wczytuje i przetwarza dane wejściowe (plik pattern.txt) – summarize+translate+category
   6. W oparciu o bibliotekę tkinter stworzono GUI, które składa się z 4 elementów :
      1. Pola do wprowadzania nowego tekstu
      2. Przycisku wyzwalającego funkcję „on\_process\_click”, która przywołuje te same czynności (summarize+translate+category) dla nowych danych

Źródła danych

Do wykonania tego projektu wykorzystano dziesiątki najrozmaitszych źródeł danych. Dla zrozumienia sensu i składni użytych tutaj metod, posiłkowano się wieloma repozytoriami github.com, na przykład:

- <https://github.com/oliver0221/Text-Summarizer-pytorch-easy-run>

- <https://github.com/tpai/summary-gpt-bot>

- <https://github.com/dongjun-Lee/text-summarization-tensorflow>

- <https://github.com/DwarakJ/TextSummarizer>

Obejrzano także dziesiątki tutoriali i opracowań w internecie, na przykład :

- kurs HUGGING FACE NA YOUTUBE: <https://youtube.com/playlist?list=PLo2EIpI_JMQvWfQndUesu0nPBAtZ9gP1o&si=zH-95s5gqV7x9cL->

- kanał : <https://www.youtube.com/@rupert_ai/videos>

- <https://www.youtube.com/watch?v=3afcOiQ-RII>

- <https://www.ovhcloud.com/pl/learn/what-is-large-language-model/>

WNIOSKI:

Realizacja indywidualnego projektu semestralnego może być niezwykle pouczająca, ponieważ pozwala na gruntowne zapoznanie się z tematem. Samodzielne osiągnięcie zakładanych efektów po wielu godzinach prób i błędów sprawia natomiast ogromną satysfakcję.