PRZETWARZANIE JĘZYKA NATURALNEGO

Temat projektu: ***Text summarizer w Pythonie***

Dominika Borowska, Aleksandra Brymerska

Kraków, 23.05.2022

# Cel projektu

Celem projektu było stworzenie narzędzia do streszczania tekstu z artykułów dostępnych w Internecie. Narzędzie to powinno przyjmować tekst, który następnie powinien być streszczonych w określonej ilości zdań – w efekcie czego uzyskujemy podsumowanie całego artykułu.

# Zakres pracy

W zakres pracy wchodziło określenie wymagań potrzebnych do zrealizowania projektu. Pierwszym krokiem było zebranie zbioru danych potrzebnego do wytrenowania modelu metodą web scrapping’u. Drugim krokiem było przygotowanie krótkiego programu, który podsumowuje tekst nie znając już wyjściowego wyniku, czyli wykorzystanie uczenia nienadzorowanego. Ostatnim i głównym punktem projektu było stworzenie modelu korzystającego z uczenia nadzorowanego oraz transformatorów, który na wejściu przyjmuje dwie kolumny – tekst artykułu oraz nagłówek i po wytrenowaniu potrafi już streścić jakikolwiek inny tekst podany na wejściu (nie znając odpowiedzi).

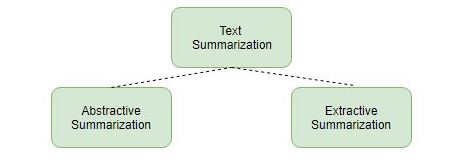
# Wstęp teoretyczny

Text summarization (streszczenie tekstu) w NLP to proces tworzenia streszczeń z dużych ilości danych przy zachowaniu istotnych elementów informacyjnych i wartości treści. Język streszczenia powinien być zwięzły i prosty, tak aby przekazywał czytelnikowi znaczenie. Według Statista do 2025 r. łączna ilość danych tworzonych, rejestrowanych, kopiowanych i zużywanych na całym świecie ma przekroczyć 180 zettabajtów. Większość tych danych należy zminimalizować do prostszych, zwięzłych podsumowań zawierających istotne szczegóły, aby łatwiej je przeglądać i analizować. Istnieje duże zapotrzebowanie na algorytmy uczenia maszynowego, które mogą szybko podsumowywać długie artykuły i dostarczać dokładnych informacji. Właśnie tam pojawia się podsumowanie tekstu.

Podsumowanie tekstu jest korzystne w przypadku kilku zadań NLP, w tym klasyfikacji tekstu, streszczeń tekstów prawnych, streszczeń wiadomości, generowania nagłówków itp.

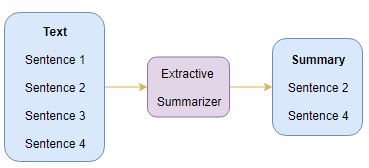
# Podejścia do streszczania tekstu

Istnieją dwa główne podejścia do podsumowywania tekstu – ekstrakcyjne i abstrakcyjne.



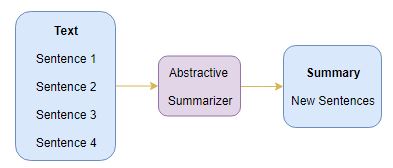
## Podsumowanie ekstrakcyjne

Podsumowywanie ekstrakcyjne polega na wyodrębnianiu podstawowych słów z oryginalnego dokumentu i łączeniu ich w celu stworzenia podsumowania. Wyodrębnianie podsumowań wykorzystuje mechanizm punktacji, aby uszeregować trafność fraz w celu wybrania tylko tych, które są najbardziej odpowiednie dla znaczenia dokumentu źródłowego. Ta technika wyodrębnia wymagany tekst zgodnie z określonymi kryteriami bez wprowadzania znaczących zmian w dokumentach. Ekstrakcyjna metoda podsumowania działa za pomocą algorytmów takich jak LexRank, Luhn, LSA itp., które są zaimplementowane przy użyciu bibliotek Pythona - Gensim lub Sumy.



## Podsumowanie abstrakcyjne

Podsumowanie to koncentruje się na najważniejszych informacjach w tekście oryginalnym i tworzy nowy zestaw zdań do podsumowania. Nowe zdanie może nie być częścią tekstu źródłowego. Podejście to jest całkowitym przeciwieństwem podejścia ekstrakcyjnego, ponieważ to ostatnie generuje podsumowanie na podstawie dokładnego tekstu oryginalnego. Technika ta obejmuje identyfikację kluczowych elementów, interpretację kontekstu i odtworzenie ich w nowy sposób. Gwarantuje podanie najistotniejszych informacji w możliwie najkrótszych słowach. Abstrakcyjna metoda podsumowania działa dobrze z modelami głębokiego uczenia, takimi jak model seq2seq, LSTM itp., a także z popularnymi pakietami Pythona (Spacy, NLTK itp.) i frameworkami (Tensorflow, Keras).



# Transformer Simple T5

T5 (Text-To-Text Transfer Transformer) to model transformatora (transformera), który jest szkolony w sposób kompleksowy z tekstem jako wejściem i zmodyfikowanym tekstem jako wyjściem, w przeciwieństwie do modeli w stylu BERT, które mogą wyprowadzać tylko etykietę klasy lub rozpiętość wejścia. To formatowanie tekstu na tekst sprawia, że ​​model T5 nadaje się do wielu zadań NLP, takich jak podsumowanie, odpowiadanie na pytania, tłumaczenie maszynowe i klasyfikowanie.

Model jest szkolony wykorzystując podejście MLM (masked language model). MLM to zadanie polegające na wypełnianiu pustych miejsc, w którym model maskuje część tekstu wejściowego i próbuje przewidzieć, jakie powinno być zamaskowane słowo. T5 zastępuje wiele kolejnych tokenów jednym słowem kluczowym maski.

Firma Google wydała wstępnie wytrenowane modele struktury text-to-text T5, które są szkolone na nieoznakowanym dużym korpusie tekstowym o nazwie C4 (Colossal Clean Crawled Corpus) przy użyciu głębokiego uczenia. C4 to oczyszczone dane tekstowe pobrane z sieci o pojemności 800 GB. Proces czyszczenia obejmuje reduplikację, odrzucanie niepełnych zdań oraz usuwanie obraźliwych lub hałaśliwych treści. T5 oczekuje prefiksu przed tekstem wejściowym, aby zrozumieć zadanie zadane przez użytkownika. Na przykład „summarize:” w przypadku podsumowania, „cola sentence:” w przypadku klasyfikacji, „translate English to Spanish:” w przypadku tłumaczenia maszynowego itp.

Diagram

Description automatically generated

Źródło: https://turbolab.in/abstractive-summarization-using-googles-t5/

W naszym projekcie model T5 wykorzystany został do uczenia nadzorowanego (simpleT5 jest stworzony tylko do tego). Jednak możliwe jest również wykorzystanie transformatora T5 do uczenia nienadzorowanego.

# Technologia

W projekcie wykorzystano język Python oraz Jupyter Notebook. Skorzystano przede wszystkim z następujących bibliotek:

- *BeautifulSoup* - biblioteka Pythona do wyciągania danych z plików HTML i XML. Wykorzystana została do scrapping’u danych ze strony internetowej z codziennymi wiadomościami.

- *requests* – biblioteka Pythona obsługująca żądania i zapytania HTTP.

- *pandas* i *numpy* – jedne z najpopularniejszych bibliotek Pythona – wykorzystane do ogólnej obsługi danych.

- *SpaCy* – biblioteka Pythona przeznaczona do zaawansowanego przetwarzania języka naturalnego.

- *simpleT5* – biblioteka umożliwiająca korzystanie z transformatora simpleT5,

oraz wiele innych dodatkowych bibliotek.

# Web scraping

Zdecydowano się na pobranie artykułów ze strony <https://www.dailymail.co.uk>. Strona ta nie blokuje dostępu do pełnych artykułów i nie wymaga logowania. Przez 3 dni pobrano po 150 artykułów ze strony głównej, co dało łącznie około 500 artykułów. Do pobierania danych wykorzystano wcześniej już opisane biblioteki BeautifulSoup oraz requests.

Text

Description automatically generated

Nagłówki newsów oznaczone są tagiem HTML „h2”, dlatego wyszukano wszystkie takie tagi na stronie głównej witryny.

Przykład nagłówka pierwszego z dnia 21.05.2022:

A picture containing scatter chart

Description automatically generated

Większa część kodu scrapera prezentowana jest poniżej. Jego głównym zadaniem jest znalezienie linku do artukułu, tytułu newsa oraz całego tekstu. Wszystko to wykonywane jest dzięki użyciu tagów HTML oraz ich wyszukiwaniu. Ponieważ paragraf tekstu znajduje się w tagu „p”, tak też jest on wyszukiwany. Ostatecznie dane zostały zapisane w dwóch zbiorach: pierwszym, zawierającym wyłącznie sam tekst artykułu (wykorzystany zostanie tylko do porównania dwóch metod streszczania tekstu) oraz drugim, zawierającym nagłówek oraz główny tekst, który zostanie wykorzystany do stworzenia głównego modelu.

Table

Description automatically generated

Przykład newsów z dnia 21.05.2022.

Ostatecznie, dane zostały zapisane w odpowiednich plikach CSV.

Text

Description automatically generated

# Streszczenie artykułów z wykorzystaniem SpaCy oraz liczności słów

Zamiast rozumieć tekst, podsumowanie ekstrakcyjne, które zostało tutaj wykorzystane, opiera się na metrykach ilościowych skonstruowanych na podstawie samego tekstu. Podejście to składa się z poniższych kroków:

- spojrzenie na częstotliwość używania określonych słów,

- zsumowanie częstotliwości w każdym zdaniu

- uporządkowanie zdania na podstawie tej sumy.

Oczywiście zakładamy, że użycie słowa o wyższej częstotliwości implikuje bardziej „istotne” znaczenie. Może się to wydawać zbyt uproszczone, ale takie podejście często daje zaskakująco dobre wyniki.

Do tego podejścia wykorzystano jeden z już gotowych kodów dostępnych w Internecie. Zmodyfikowano go o drobny preprocessing danych, czyli usunięcie odnośników.   
  
Wykorzystano, już wcześniej opisaną, bibliotekę SpaCy do zaimportowania wstępnie wytrenowanego potoku NLP, aby pomóc w interpretacji struktury gramatycznej tekstu. Do pozbycia się słów stopu wykorzystano zbudowane już w SpaCy STOP\_WORDS, a także pozbyto się interpunkcji (biblioteka punctuation). Użyto również funkcji nlargest, aby wyodrębnić procent najważniejszych zdań.

Zasada działania algorytmu jest dość prosta i została opisana poniżej:

- tokenizacja tekstu z wykorzystaniem SpaCy.

- wektoryzacja i normalizacja liczby wystąpień – policzenie ile razy słowo zostało użyte w zdaniu (w tym wypadku pojedynczym wierszu). Słowa używane częściej mają wyższą wartość.

- obliczenie sumy znormalizowanej liczby dla każdego zdania.

- wyodrębnienie procentu najwyżej ocenionych zdań, które stanowią podsumowanie.

Przykład tekstu przed podsumowaniem: Text

Description automatically generated

Podsumowanie dla 10 procent:

Text

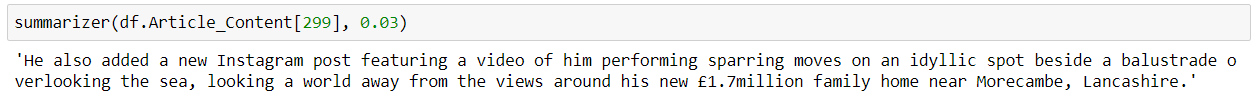
Description automatically generated with low confidence

Podsumowanie dla 40 procent:

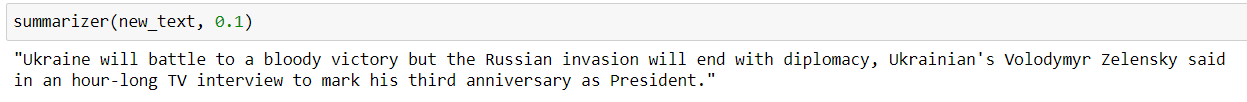
Text, letter

Description automatically generated

Podsumowanie dla 3 procent:



Podsumowanie dodatkowego tekst:



# Streszczenie tekstu z wykorzystaniem transformatora i uczenia nadzorowanego

Do tej części projektu wykorzystano simpleT5. Jest to model zbudowany na bazie technologii PyTorch-lightning i Transformers, co pozwala szybko trenować modele T5.

T5 to model z koderem i dekoderem. Konwertuje wszystkie problemy NLP, takie jak tłumaczenie języka, podsumowania, generowanie tekstu, odpowiadanie na pytania, na zadanie zamiany tekstu na tekst.

Przygotowano klasy pomocnicze takie jak:

- ustawienia – klasa wskazująca ogólne elementy, takie jak rodzaj modelu, kolumny, wykorzystanie technologii CUDA (na gogle collab), wskazanie ilości epok, wielkości zbioru testowego oraz do treningu.   
  
Graphical user interface, text

Description automatically generated  
  
Text

Description automatically generated with medium confidence  
  
Model T5 wymaga wskazania słowa, które wskazuje na tekst będący podsumowanie. W naszym przypadku będzie to po prostu słowo summarizer. W następnych krokach zostanie ono dodane przed nagłówkiem w każdym wierszu.  
  
- preprocessing – podstawowe operacje na zbiorze danych będące preprocessingiem, czyli pozbycie się linków, specjalnych znaków, dodatkowych spacji oraz \n. Następnie zmiana nazw kolumn (simpleT5 oczekuje od zbioru aby posiadał dwie kolumny: "source\_text" i "target\_text") oraz dodanie wcześniej wymienione słowa summarizer na początku nagłówków.  
  
A picture containing text

Description automatically generated

- klasa model – wskazująca na model T5

- klasa Train – tutaj inicjowane są wszystkie wcześniejsze klasy oraz następuje proces trenowania danych. Dodatkowo tutaj dodane zostały wszelkie ostrzeżenia oraz informacje drukujące się podczas trenowania.   
  
Text

Description automatically generated with medium confidence

Po odpowiednim trenowaniu wyniki zapisują się w folderze outputs. Następnie najlepsze wagi wykorzystujemy do podsumowania nowego kawałka tekstu, tym razem wziętego ze strony CNN.

Ukraine will battle to a bloody victory but the Russian invasion will end with diplomacy, Ukrainian's Volodymyr Zelensky said in an hour-long TV interview to mark his third anniversary as President. "We did not start this war. But we have to finish it," Zelensky said in an interview that was recorded Friday and broadcast Saturday. "Victory will be bloody in battle. But the end will be in diplomacy. We want everything back. Russia does not want to give anything away," Zelensky said. The broadcast also included an interview with Zelensky’s wife Olena Zelenska, who said the war had not changed her husband. "I can't say that he has changed. As he was a reliable husband and man, so he is," she said. But she lamented not being able to see him for months. "Our family, like all Ukrainian families, is now torn. We didn't see each other for two and a half months, we only talked on the phone. Thank you for this opportunity [this interview], because we are spending time together now -- dating on TV," she said.

Nagłówek ze strony CNN: After bloody battles, Ukraine war will end with diplomacy, Zelensky vows in anniversary interview

Podsumowanie wykonane przez model: Ukraine will fight to a bloody victory but the Russian invasion will end with diplomacy, Ukrainian President Volodymyr Zelensky says  
  
Porównując to z podsumowaniem stworzonym przez poprzedni model: Ukraine will battle to a bloody victory but the Russian invasion will end with diplomacy, Ukrainian's Volodymyr Zelensky said in an hour-long TV interview to mark his third anniversary as President.

# Podsumowanie

Oba streszczenia są bardzo podobne do siebie i całkowicie oddają sens artykułu. Obie metody, zarówno ekstrakcyjna jak i abstrakcyjna bez problemu radzą sobie z podsumowywaniem artykułów. Metoda abstrakcyjna jest o wiele bardziej czasochłonna ponieważ w naszym przypadku długo zajmuje trenowanie modelu, jej zaletą jest jednak możliwość odpowiedniego dostosowania parametrów. Metoda ekstrakcyjna bazowała wyłącznie na zliczaniu częstości słów w zdaniu.