Jan Proniewicz, 297989, Informatyka - Data Science

Języki i Biblioteki Analizy Danych - Projekt zaliczeniowy, pt. “Kategoryzacja notatek prasowych / artykułów internetowych”

Ogólny opis systemu:

Omawiany projekt działa na zasadzie aplikacji konsolowej. Tzn. że nie posiada graficznego interfejsu i wszystkie funkcjonalności odbywają się za pośrednictwem konsoli. Działanie systemu testowano za pośrednictwem środowiska PyCharm, ale powinien również być możliwy do uruchomienia w wierszu poleceń (należy wtedy uruchomić plik *Menu,py*). Na potrzeby zajęć, powstał on w języku Python (autor korzystał z wersji 3.6).

Celem niniejszego programu jest umożliwienie użytkownikowi pobranie z internetu “najnowszych” notatek prasowych lub artykułów (dla uproszczenia, pojęcia te będą stosowa), ich kategoryzację, względem odgórnie określonych kategorii oraz ich szybki przegląd. Źródłem tych notatek są wybrane przez autora strony internetowe portali prasowych. W tym przypadku, są to Wyborcza (*wyborcza.pl*), Onet (*wiadomosci.onet.pl* i *sport.onet.pl*) oraz Wirtualna Polska (*wiadomosci.wp.pl* i *sportowefakty.wp.pl*). Informacje są zbierane z różnych zakładek tych stron internetowych, np. świat, kraj, Kraków, Śląsk itd.. Oczywiście, każdy z tych portali ma swój własny zestaw zakładek. Ograniczono się tylko do niektórych z nich. Poprzez “najnowsze” artykuły, autor ma na myśli te, które są dostępne na pierwszej stronie portalu (pierwsza stronica, bez rozwijania) oraz takie, które nie zostały opublikowane wcześniej, niż określonego dnia (o takiej dacie “granicznej” decyduje użytkownik; dokładniej to zostanie wytłumaczone później).

W ramach niniejszego systemu, każda notatka zawiera: tytuł artykułu, krótki opis na wstęp (tzw. lead), datę publikacji, odsyłacz URL do oryginalnego artykułu oraz nazwę zakładki, z której ta notatka pochodzi. Pobrane notatki są później zapisywane do

pliku *press_notes.json*. Dzięki temu, pobrane ówcześnie dane nie zostaną utracone, a program będzie miał do nich dostęp nawet po ponownym uruchomieniu.
Jak już wspomniano wcześniej, kategoryzacja pobranych notatek prasowych opiera się na ich przydziale do odgórnie określonych przez system kategorii, tj. “polityka”, “sport”, “piłka nożna”, "zimowy sport", "tenis", "sport motorowy", "kolarstwo", “koronawirus”, “medycyna”, “nauka i technika”, “gospodarka”, “kultura”, “religia i Kościół”, “kryminalne i wypadki” i “inne” (kategoria domyślna, w której znajdują się artykuły, niepasujące do żadnej z wymienionych grup). Aplikacja nie wyklucza sytuacji, w której jedna notatka prasowa mogłaby się znaleźć w różnych kategoriach. Omawiany proces jest dokonywany na podstawie zakładki, z której dana notatka pochodzi oraz treści, wynikającej z tytułu oraz lead’u. Wyniki klasyfikacji zostają później zapisane w pliku *notes_classified.json*, co usuwa konieczność ponownej kategoryzacji, jeśli program zostałby później uruchomiony ponownie. Warto również zaznaczyć, że artykuły w obu wymienionych plikach są posortowane wg daty publikacji, w kolejności od najnowszego do najstarszego, w celu ułatwienia czytelności, w razie otwarcia tych plików.
Posegregowane artykuły można wyświetlić w konsoli, co umożliwia specjalnie do tego utworzone menu. W czytelny sposób pokazuje pobrane informacje na ich temat, tj. tytuł, lead, data publikacji oraz odsyłacz do pełnej treści oryginalnego artykułu. Ze względu na charakter aplikacji (tj. konsola), notatki są wypisywane w kolejności od najstarszej do najnowszej, żeby użytkownik nie musiał “przewijać” okna konsoli na samą górę, by zobaczyć najnowsze wiadomości.
System pozwala również na usunięcie z “bazy” starych notatek prasowych, w celu zwolnienia pamięci. Należy jednak pamiętać, iż jest to proces nieodwracalny.

Przewodnik użytkownika:

W momencie uruchomienia programu (a mówiąc ściślej - pliku *Menu.py*), konsola powinna wyświetlić “menu główne”, w którym są wylistowane możliwe funkcjonalności systemu, tak jak na rysunku poniżej.

```
C:\Python\python.exe C:/Users/Jan/Documents/python/JanProniewicz_Projekt/program_files/Menu.py
=============================================================================================================
System klasyfikacyjny internetowych notatek prasowych:
=============================================================================================
Wybierz opcję:
1 - Pobierz więcej notatek z internetu (strony: wyborcza.pl, onet.pl, wp.pl);
2 - Usuń 'stare' notatki prasowe;
3 - Przeprowadź klasyfikację notatek prasowych;
4 - Pokaż notatki prasowe wg kategorii;
5 - Wyjdź
```

Użytkownik może pobrać najnowsze artykuły ze wszystkich trzech podanych portali prasowych, usunąć stare notatki, sklasyfikować pobrane notatki, wyświetlić ich spis wg kategorii lub po prostu wyjść. Wybór opcji następuje poprzez wpisanie w konsoli odpowiedniej liczby.

Chcąc pobrać więcej notatek prasowych, system w pierwszej kolejności poprosi użytkownika o podanie roku, miesiąca i daty (liczbowo). Będą one reprezentowały tzw. “minimalną datę”, która będzie stanowić punkt przerwania pobierania, tzn. program nie będzie brał pod uwagę artykułów opublikowanych wcześniej, niż tego dnia. Po zakończeniu procesu, powinna się wyświetlić liczba notatek, pobrana z każdego portalu (w sumie, z bazy, nie w danym momencie) oraz stosowny komunikat. Niestety, ze względu na złożoność algorytmu, funkcja ta działa bardzo wolno (szczególnie jeżeli postanowi się pobrać nowe notatki po bardzo długim czasie), dlatego wykonując ją, należy uzbroić się w cierpliwość. Użytkownik ostatecznie powinien mieć widok taki, jak na rysunku.

```
=============================================================================================
Podaj 'najwcześniejszy' dzień (pełna data), do którego notatki będą pobierane:
Rok: 2021
Miesiąc: 1
Dzień: 5
Liczba notatek prasowych w bazie:
Wyborcza: 585
Onet: 886
Wp: 755
Razem: 2226
Pobieranie zakończone pomyślnie!
```

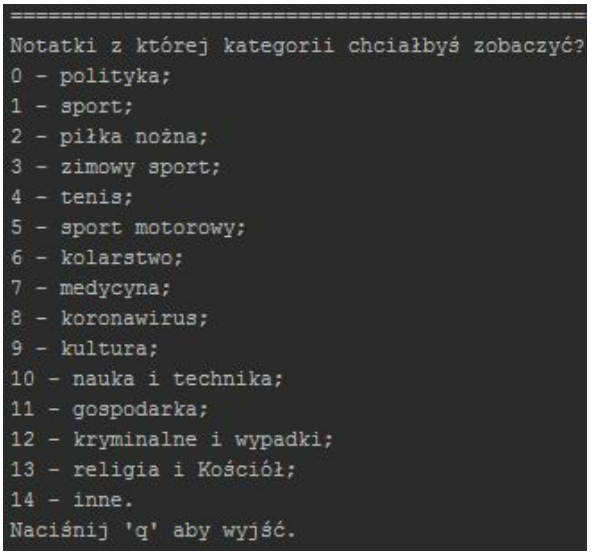
Usuwanie ze strony użytkownika opiera się na podobnej zasadzie, tzn. proces prosi o podanie roku, miesiąca i dnia. Tym razem, oznaczają one “datę maksymalną”, stanowiącą punkt przerwania usuwania. Program usunie tylko te artykuły, które zostały opublikowane przed danym dniem. Po wykonaniu funkcji, system wyświetli liczbę usuniętych notatek oraz stosowny komunikat, jak na rycinie poniżej.

```
================================================================================
Podaj 'najpóźniejszy' dzień (pełna data), do którego notatki będą usuwane:
UWAGA! Proces nieodwracalny!
Rok: 2021
Miesiąc: 1
Dzień: 5
Usunięto: 28
Usuwanie 'starych' notatek prasowych zakończone pomyślnie!
```

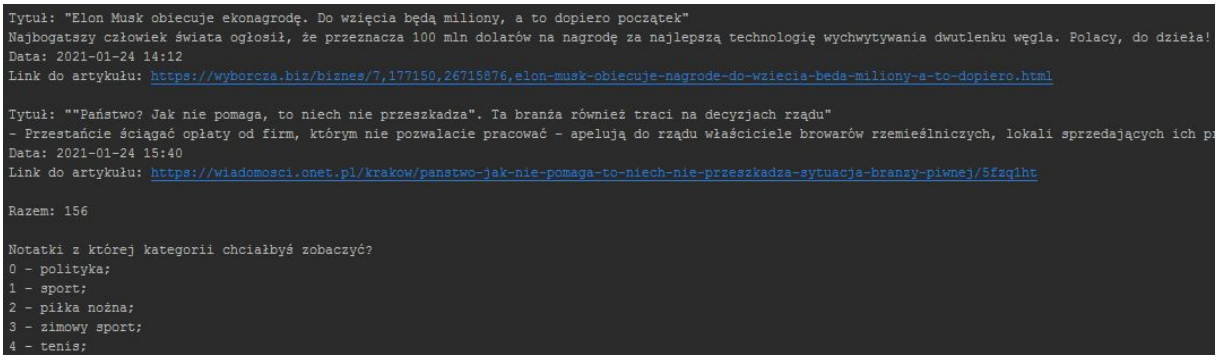
Klasyfikacja od użytkownika będzie wymagała podjęcia jednej decyzji. Mianowicie, czy użytkownik będzie chciał zresetować zbiór sklasyfikowanych notatek, tj. opróżnić plik *notes_clssified.json* i sklasyfikować wszystko od nowa (*y* na “Tak”), czy tylko podzielić nowo pobrane notatki, a “te stare zostawić w spokoju” (*n* na “Nie”). Pierwsza metoda pozwala na rekonfigurację “bazy” skategoryzowanych artykułów, jeśli program przeszedł przez znaczące zmiany, a druga - jest w pewnym sensie bardziej oszczędna. Mimo to, ponownie, ze względu na złożoność algorytmu, funkcja ta może działać stosunkowo wolno. Po zakończeniu procesu, system wyświetla liczbę notatek prasowych w każdej z kategorii, jak poniżej.

```
==================================================
Czy chcesz skategoryzować notatki od nowa?
Tak (y) / Nie (n)y
Liczba artykułów/notatek w każdej z kategorii:
Polityka: 332
Sport: 904
Piłka nożna: 306
Zimowy sport: 262
Tenis: 146
Sport motorowy: 127
Kolarstwo: 29
Medycyna: 270
Koronawirus: 447
Kultura: 94
Nauka i technika: 95
Gospodarka: 156
Kryminalne i wypadki: 303
Religia i kościół: 33
Inne: 235
Kategoryzacja notatek prasowych zakończona pomyślnie!
```

Po wyborze opcji wyświetlania posegregowanych notatek prasowych, system przekieruje użytkownika do osobnego menu, w którym ma po kolei wylistowane kategorie i odpowiadające im indeksy. Żeby wyświetlić artykuły dla któreś z nich, należy wprowadzić odpowiedni indeks. Można również wprowadzić ‘q’, aby wrócić do menu głównego.



Po wyborze indeksu, program wypisuje wszystkie notatki z danej kategorii w kolejności od najstarszej do najnowszej, w celu ułatwienia czytelności dla użytkowników konsolowych. Dzięki temu, użytkownik może z łatwością zobaczyć aktualne, ogólne wiadomości z danej dziedziny. Jeśli jakiś temat go zainteresuje, może również kliknąć odpowiedni odsyłacz, który przekieruje go w przeglądarce do pełnego artykułu (Wyborcza niestety wymaga wykupienia prenumeraty, żeby mieć pełny dostęp do treści artykułów). Następnie, system wraca do menu kategorii. Poniżej przedstawiono widok (jego fragment) końca wizualizacji listy dla notatek prasowych z kategorii “gospodarka”.



Jeżeli w jakimkolwiek momencie użytkownik poda dane, które system nie rozpozna, np. losowy ciąg znaków, wyświetli on stosowny komunikat o błędzie i wróci do menu głównego. Można to uznać za sposób anulowania operacji. Należy mieć na uwadze, że w przypadku opcji 1 i 2 system zareaguje dopiero wtedy, gdy wszystkie parametry dla daty zostaną wypełnione (prawidłowo lub nie). Aby zakończyć działanie aplikacji, wystarczy w menu głównym podać cyfrę 5.

Pełny opis systemu:

Zasadniczo, program, realizujący wymienione zadania, dzieli się na sześć plików systemowych (*.py*) i dwa, reprezentujące “bazę” pobranych i posegregowanych notatek (*.json*). Pliki systemowe to:

- *Menu.py* - menu główne oraz plik inicjujący aplikację;
- *PressNotesDownloader.py* - klasa pobierająca notatki prasowe z Wyborczej, Onetu i WP;
- *PressNotesDeleter.py* - klasa usuwająca “stare” notatki prasowe;
- *PressNotesClassifier.py* - klasa kategoryzująca pobrane notatki prasowe;
- *PressNotesVisualizer.py* - klasa generująca w konsoli notatki prasowe z wybranych kategorii;
- *JsonOperations.py* - metody pomocnicze, pobierające treść z plików *.json* i zapisujące nowe informacje do nich.

Natomiast wśród plików “bazy” można wyróżnić:

- *press_notes.json* - tutaj są zapisywane pobrane notatki prasowe, z podziałem na portale, z których pochodzą, tj. “wyborcza”, “onet” i “wp”;
- *notes_classified.json* - jak sama nazwa wskazuje, jest to zapis skategoryzowanych notatek prasowych z podziałem na kategorie: “polityka”, “sport”, “piłka nożna”, "zimowy sport", "tenis", "sport motorowy", "kolarstwo", “koronawirus”, “medycyna”, “nauka i technika”, “gospodarka”, “kultura”, “religia i Kościół”, “kryminalne i wypadki” i “inne”.

Podzbiory notatek prasowych (dla portali i kategorii) są słownikami, w których każda notatka jest reprezentowana przez swój tytuł (jest on kluczem notatki). Z kolei same artykuły również są słownikami, o kluczach: “Tytuł”, “Opis” (lead), “Data”, “URL” i “Zakładka”.

Menu.py:

Samo menu opiera się na zasadzie pętli *while*, która nie zostanie przerwana, dopóki użytkownik nie zdecyduje się wyjść (opcja nr 5). Sprecyzowane zostały w niej instrukcje warunkowe, realizujące odpowiednie funkcje z pozostałych plików systemowych, w zależności od *inputu* użytkownika. Obsługuje także wyjątki *ValueError* i *IndexError*, na wypadek, gdyby użytkownik wprowadził niezgodne dane w jakimkolwiek momencie. Wraca wtedy na początek pętli do menu głównego.

PressNotesDownloader.py:

Do pobierania informacji o notatkach prasowych, zastosowano dość niekonwencjonalną, aczkolwiek interesującą metodę, polegającą na tzw. *Web Scraping'u*. Polega ona na wydzielaniu określonych fragmentów kodu html wybranej strony. W pythonie, taką technikę umożliwia specjalna biblioteka o nazwie *beautifulsoup4*. Jej metoda *BeautifulSoup* potrafi zinterpretować zawartość strony internetowej, pobraną wcześniej np. przy pomocy biblioteki *requests* i przetworzyć ją na czytelny kod html. Następnie, dzięki innym metodom, wchodzących w skład omawianej biblioteki, można np. wyselekcjonować konkretne fragmenty tego kodu na podstawie typu i atrybutów, pobrać wartości tych atrybutów lub pobrać tekst, znajdujący się w danym "bloku". Żeby móc z łatwością wyselekcjonować fragmenty, których program rzeczywiście potrzebuje, warto na oryginalnej stronie internetowej wybrać opcję "Zbadaj" po wciśnięciu prawego przycisku myszy, aby wyświetlić jej kod html a następnie sprawdzić, które jego fragmenty dotyczą konkretnych elementów strony. Mając to na uwadze, można przejść do pewnego pisania algorytmu. Należy również pamiętać, że każda strona internetowa ma inną budowę, więc jeden program, stworzony dla jednej konkretnej strony może nie działać poprawnie dla wszystkich z nich. Dlatego, niezbędnym było utworzenie osobnych metod pobierania notatek prasowych dla Wyborczej, Onetu i WP. Dla dwóch ostatnich portali należało te metody jeszcze dodatkowo podzielić na funkcje wewnętrzne, ze względu na różny charakter niektórych zakładek / stron.

Sam proces pobierania notatek prasowych opiera się na utworzeniu obiektu klasy *PressNotesDownloader* i wykorzystaniu jej metod. W momencie jej utworzenia, zapisuje sobie wspomnianą "najwcześniejszą datę" (jej elementy są podawane jako argumenty) i przekształca ją na format *datetime* (do tego użyteczny jest moduł *datetime*). Oprócz tego, pobiera on zawartość pliku *press_notes.json*, aby móc później do niego zapisać nowo pobrane notatki oraz przygotowuje sobie spis zakładek dla każdego z portali (dla Wyborczej dodatkowo przygotowuje ciągi cyfr, będące charakterystycznymi częściami URL kolejnych zakładek).

Jak wspomniano wcześniej, klasa zawiera osobne metody, wykonujące Web Scraping dla poszczególnych portali:

- *scrape_wyborcza* - wykonująca Web Scraping dla Wyborczej (*wyborcza.pl*).

- *scrape_onet* - wykonująca Web Scraping dla Onetu, tj. *wiadomosci.onet.pl* i *sport.onet.pl*. Dzieli się na trzy osobne metody wewnętrzne. Jedna została stworzona dla stron, gdzie artykuły są wylistowane, druga - dla zakładek dot. miast i regionów (tam artykuły są ułożone jak “kafelki”), a trzecia z kolei pobiera niezbędne informacje z kolejnych artykułów.
- *scrape_wp* - wykonująca Web Scraping dla Wirtualnej Polski (*wiadomosci.wp.pl* i *sportowefakty.wp.pl*). Ze względu na całkowicie odmienny wygląd tych dwóch stron, dla każdej z nich utworzono osobne metody wewnętrzne.

Każda z wyżej wymienionych funkcji opiera się z grubsza na tych samych założeniach. Mianowicie, wykonują one odpowiednie operacje, iterując po kolejnych, przypisanych im kategoriach. W pierwszej kolejności, “budują” adres URL wybranej zakładki i za sprawą biblioteki *requests* - używają go do pobrania zawartości odpowiedniej strony internetowej. Następnie, przy pomocy *BeautifulSoup*, przetwarzają zawartość strony na kod html i wyszukują odpowiednich danych. M. in. wybierają “blok” kodu, zawierający informacje o artykułach/notatkach prasowych, dzielą go na poszczególne, osobne “bloczki” notatek, a potem, dla każdego z nich, wybierają niezbędne dane, opisane we wcześniejszych paragrafach, np. tytuł, lead itd. Wszystkie te informacje oraz nazwa przeglądanej zakładki są zbierane do kolejnych słowników, które następnie są dodawane do zbioru (słownika) artykułów z odpowiedniego źródła. Przy użyciu funkcji *save_press_notes* obiekt *PressNotesDownloader* zapisuje wszystko z powrotem do pliku *press_notes.json*.

Funkcje te obsługują również *AttributeError*, na wypadek, gdyby aktualnie przeglądana strona miała inny format, niż obsługiwany przez program (gdyż np. pochodzi z innego URL i *BeautifulSoup* nie mogłoby znaleźć szukanych elementów). Wtedy po prostu pomija dany artykuł i bada następny.

W trakcie pobierania notatek z internetu wykonywane są również procedury pomocnicze, np. weryfikacja, czy notatka o danym tytule się już pojawiła (jeśli tak, to najpewniej następne też będą się powtarzać, więc program przejdzie do analizy kolejnej zakładki), czy data publikacji nie jest “za stara” w porównaniu do ustalonej “minimalnej” oraz metoda sortująca pobrane już notatki prasowe wg daty publikacji od najnowszej do najstarszej (dla lepszej czytelności i zachowania spójności w algorytmach; tu przydatane są metody *datetime* i *OrderedDict* kolejno z modułów/bibliotek *datetime* i *collections*).

PressNotesDeleter.py:

W trakcie tworzenia obiektu klasy *PressNotesDeleter* program zapisuje w nim wspomnianą wcześniej “maksymalną datę” (jej elementy są podawane jako argumenty) oraz pobiera zawartość obu plików *.json*. Proces usuwania “starych” notatek obiera się na wywołaniu utworzonej metody *delete_old_press_notes*, która iteruje po kolejnych notatkach z każdego zbioru (dla konkretnych portali lub kategorii) od najstarszej do najnowszej. Jeżeli data publikacji artykułu jest wcześniejsza niż “maksymalna”, notatka zostanie usunięta. W przeciwnym wypadku, funkcja dobiega końca. Proces jest powtarzany każdego z dwóch plików .json (po zakończeniu operacji, można użyć *modified_press_notes*, aby zapisać nowy stan notatek prasowych do odpowiednich plików).

PressNotesClassifier.py:

Nietypowa sytuacja również prezentuje się w przypadku procesu kategoryzacji notatek prasowych. Ponieważ system umożliwia przydział notatki prasowej do kilku kategorii naraz, typowa klasyfikacja nie może zostać zastosowana. Dlatego, niniejszy algorytm opiera się na dwóch sposobach:

1. Przydział na podstawie zakładki, z której pochodzi dana notatka. W tym celu, program będzie sprawdzał wartość klucza “Zakładka” w każdej z notatek i sprawdzał, czy taka zakładka jednoznacznie wskazywałaby na jedną z ustalonych kategorii. Także, postanowiono, że jeżeli artykuł pochodzi z zakładki dla określonej dyscypliny sportu, to powinien być także w kategorii “sport”, a jeśli wystąpił w zakładce “koronawirus”, to znajdzie się również w “medycynie” (zazwyczaj notatki z zakładki “koronawirus” są ściśle powiązane z tematami medycznymi).
2. Przydział wg liczby wystąpień określonych form w tytule i opisie (lead’ie). Podczas pobierania informacji o notatkach prasowych nie koncentrowano się na pełnych treściach artykułów, ponieważ z jednej strony mogłoby to być zbyt obciążające dla programu. Też z drugiej strony, tytuł i lead powinne teoretycznie zawierać wszystko, co trzeba wiedzieć o treści artykułu, więc klasyfikację oparto tylko na nich. Liczby wystąpień form obliczano za pomocą biblioteki *re* (*regex*), które wykrywają obecności pewnych ciągów znaków w tekście. Żeby uniknąć błędnych przyporządkowań notatek prasowych do

niektórych kategorii, w związku z przypadkowymi wystąpieniami form, za “optymalną” liczbę wystąpień ustalono ‘3’ (należy też mieć na uwadze, że tytuły i lead’y są stosunkowo krótkie). Jednakże, nie eliminuje to całkowicie błędów w klasyfikacji. Dla niektórych kategorii, takich jak “koronawirus”, czy dyscypliny sportowe, postawiono na sprawdzenie, czy chociaż jedno z wyrażeń kluczowych występuje (ponieważ jest to tematyka bardziej specyficzna / węższa). Dodatkowo, jeśli program uzna, że artykuł powinien się znaleźć w kategorii dla jakiejś dyscypliny, to powinien być też w kategorii “sport” (system nie robi czegoś podobnego dla “koronawirusa”, gdyż samo wspomnienie o “pandemii” nie świadczy o tematyce medycznej). Jednakże, w związku z tym rośnie ryzyko, że w kategorii “sport” znajdą się artykuły, zupełnie nie związane z tą tematyką (bo np. przypadkowo pojawił się w notatce “rowerzysta”). Ale to prędzej wynika z braku wiedzy autora na temat sportu i powiązanych z nim słów (w szczególności nazwisk mało znanych zawodników, drużyn czy wydarzeń).

Przy tworzeniu obiektu klasy *PressNotesClassifier*, system pobiera zawartość obu plików *.json* (z *press_notes.json* są “wyciągane” nowo pobrane notatki, a do *notes_classified.json* będą one zapisywane po klasyfikacji) oraz przygotowuje listę ustalonych odgórnie kategorii tematycznych oraz odpowiadających im listy “wyrażeń klucz”. Są to fragmenty form określonych rzeczowników, czasowników itd., które zazwyczaj pozostają niezmienne przy deklinacji, odmianie itp.. Zawierają one także łańcuchy znaków z nawiasami kwadratowymi, wewnątrz których znajdują się określone litery. Biblioteka *re* interpretuje je jako formy, gdzie litery w nawiasach są zamienne (np. “rann[aiy]” interpretuje jednocześnie jako “ranna”, “ranni” i “ranny”). Utworzona metoda *classify_press_notes* dla każdej badanej notatki tworzy pustą listę, w której będą umieszczane potencjalne kategorie dla niej. Jest ona uzupełniana za pośrednictwem dwóch wcześniej wspomnianych sposobów. Po zakończeniu “poszukiwań”, notatka jest wpisywana do wszystkich kategorii wymienionych w liście. Jeżeli nie odnaleziono żadnej kategorii dla artykułu, zostaje on umieszczony w grupie “inne”. Może to wynikać z obecności potencjalnej kategorii, która nie została uwzględniona przez autora lub niedoboru kluczowych wyrażeń.

Należy pamiętać, że z daną dziedziną może być związanych wiele pojęć i nie będzie się pamiętało o wszystkich od razu. Dlatego jest to jeden z tych algorytmów, które

warto jest cały czas kontrolować i uzupełniać o brakujące słowa klucze, w oparciu np. o artykuły z kategorii “inne” lub osoby bardziej zaznajomione z daną dziedziną.
Klasa zawiera również kilka metod pomocniczych, np. weryfikacja, czy notatka już należy do jednej z kategorii (jeśli tak, można założyć, że pozostałe też już są zaklasyfikowane), sortowanie po dacie publikacji w kolejności malejącej (analogiczna do *PressNotesDownloader.py*), czy też reset klasyfikatora, opróżniającego plik *notes_classified.json*, aby móc przeprowadzić klasyfikację od nowa (przydatna, gdy algorytm zostanie np. uzupełniony o nowe słowa klucze lub kategorie). Rzecz jasna, istnieje również metoda odpowiedzialna za zapis zmian w klasyfikacji do odpowiedniego pliku.

PressNotesVisualizer.py

Obiekt klasy *PressNotesVisualizer* pobiera zawartość pliku *notes_classified.json* oraz określa listę wszystkich ustalonych kategorii. Jej główna metoda *continuous_visualization* odpowiada za tworzenie menu, przedstawionego w instrukcji użytkownika. Działa ono na zasadzie pętli, podobnie do *Menu.py* i podobnie jak tam, żeby wyjść, należy wprowadzić odpowiedni *input* (‘q’) (lub podać błędne dane, które menu główne potem obsłuży). Po wybraniu opcji przez użytkownika, program wywołuje metodę *show_press_notes_from_category*, wybierającą kategorię z listy, ściśle powiązaną z podanym wcześniej indeksem, a następnie wyświetla w odpowiednim formacie wszystkie notatki z tej kategori w kolejności od najstarszej do najnowszej.

Instalacja i podręcznik administratora:

Żeby program działał jak trzeba, potrzebne będą następujące biblioteki i moduły:

- *json* - pozwala operować na plikach typu *.json* (pobierać ich treść, modyfikować je);
- *requests* - pobiera zawartość stron internetowych na podstawie URL;
- *bs4 (beautifulsoup4)* - przekształca treść pobraną przez *requests* na html i pozwala na niej operować;
- *collections* - z niej pochodzi *OrderedDict*, w którym można sortować treść słowników;

- *datetime* - pomocna przy zamienianiu formatu daty ze *string* na *datetime* (bezpieczne rozwiązanie przy sortowaniu wg daty lub porównywaniu dat);
- *re (regex)* - przydatna przy wykrywaniu określonych wyrażeń w łańcuchach znaków.

Aby zainstalować program, należy go najpierw pobrać z repozytorium, w którym się znajduje (tym samym, co niniejsza dokumentacja), a następnie - wypakować otrzymany plik *.zip* w wybranym miejscu. Otrzymany folder warto uruchomić jako cały projekt np. w takim PyCharm'ie, ale otwarcie pliku *Menu.py* w wierszu poleceń też raczej powinno zadziałać. Przede wszystkim, nie należy nic zmieniać w układzie plików i folderów. W przeciwnym wypadku, aplikacja może nie zadziałać.
Także, jak już wspomniano wcześniej, procesy pobierania i klasyfikacji notatek prasowych mogą być czasochłonne. Zatem należy się uzbroić w cierpliwość.