Przygotowanie zasobów do analizy cechy *Opinion*

Instrukcja na podstawie pracy magisterskiej Marty Lisińskiej pt. „Profilowanie opinii klientów o wybranym produkcie z wykorzystaniem analizy sentymentu"; opieka merytoryczna: M. Nowakowska

W pliku z danymi każdego zespołu, otrzymanym ze scenariusza nr 1, docelowo mają powstać następujące kolumny będące wynikiem prac:

Tkns -– lista tokenów (wyrazów) występujących w opinii
Wrd_Cnt – informacja o liczbie wyrazów w opinii
Lmms – lista lematów (postaci kanonicznych wyrazów) w opinii
Vcblr – słownik dla opinii (lista unikatowych lematów występujących w opinii)
Vcblr_Cnt – lista liczników dla kolejnych elementów słownika
POS_Tags – lista otagowanych wyrazów opinii; jeden element listy to krotka zawracają wyraz i część mowy (wykaz symboli części mowy NLP w Pythonie załączono na końcu instrukcji)
Nouns – lista rzeczowników opinii w formie kanonicznej
Verbs – lista czasowników opinii w formie kanonicznej
Adj – lista przymiotników opinii w formie kanonicznej
Adv – lista przysłówków opinii w formie kanonicznej

*Sntmnt_* – sentyment przypisany do każdej opinii.

Poszczególne kolumny będą powstawać wraz z zaawansowaniem analizy.

Wstępne przetwarzanie tekstu

1. Wykonać wstępne przetwarzanie opinii wg poniższych punktów.
   - Obsłużyć emoji i emotikony. W nowej kolumnie *Opinion_norm* zastąpić ww. symbole emocji jednym wyrazem zachowującym wydźwięk emocjonalny a pozostałe opinie przepisać do tej kolumny.
   - Wykonać operację normalizacji w kolumnie *Opinion_norm*; usunąć znaki specjalne i liczby oraz jednostki miary, zamienić wszystkie litery na małe.
2. Wykorzystując wybrane narzędzia IT (MS Access, Excel, Python) wyznaczyć dla każdej opinii wartości w kolumnach: *Tkns, Wrd_Cnt*, *Lmms*, *Vcblr*, *Vcblr_Cnt*, *POS_Tags*, *Nouns*, *Verbs*, *Adj*, *Adv*.
3. Wykorzystując wybrane narzędzia IT (MS Access, Excel, Python) utworzyć słownik korpusu opinii dla wybranych części mowy. W tym celu połączyć słowniki z poszczególnych opinii i wygenerować dla połączonego zbioru wykaz słów unikatowych.

Dalsze przetwarzanie danych obejmuje **w podanej kolejności**:

a) opracowanie i zastosowanie stop listy,
b) opracowanie i zastosowanie tezaurusa.

Stop lista zawiera słowa, które w analizie domenowej NLP nie wnoszą znaczącej informacji. Stop lista umożliwia usunięcie takich zbędnych wyrazów, które stanowią około 40-60% słów z tekstu (przyimki, np.: *in - w*, *on - na*, zaimki, np.: *I - ja*, *he*, *she – on*, *ona*). Jeśli najpierw usunie się te słowa, tezaurus będzie musiał przetworzyć znacznie mniej danych.

W przetwarzaniu języka naturalnego (NLP) tezaurus służy głównie do normalizacji semantycznej: zamiany synonimów na jedno słowo kluczowe (np. *auto, fura, pojazd → samochód*; *car, automobile, carriage → vehicle*), co ujednolica dane. Jest to też metoda redukcji wymiarowości: zmniejszenie liczby unikalnych słów w słowniku przy zachowaniu sensu wypowiedzi przyspiesza analizę i czyni ją bardziej spójną.

Praca ze stop listą

Opracować w swojej bazie danych tabelę ze stop listą do słownika korpusu analizowanych opinii. Przykładowy wycinek takiego zasobu przedstawiono na rys. 1. Struktura tabeli jest następująca.

- Kolumna *word* zawiera wszystkie słowa ze słownika korpusu.
- Kolumna *pos* zawiera informację o odpowiadającej temu słowu części mowy.
- Kolumna *stop_w* (typu *Tak*/*Nie*) zaznacza słowa ze słownika, które występują na stop liście standardowej.
- Kolumna *stop_n* (typu *Tak*/*Nie*) zaznacza słowa ze słownika, które występują na stop liście niestandardowej.
- Kolumna *stop_merge* (typu *Tak*/*Nie*) zaznacza słowa z obu list (połączenie informacji).

word	pos	stop_w	stop_n	stop_merge
ab	noun	☑	☑	☑
ability	noun	☐	☐	☐
able	adj	☑	☐	☑
about	adv	☑	☐	☑
above	adv	☑	☐	☑
above	adj	☑	☐	☑
aboveaverage	noun	☐	☑	☑
aboveaverage	adj	☐	☑	☑
abovementioned	adj	☐	☑	☑
abroad	adv	☐	☐	☐
abs	verb	☐	☑	☑
absolutely	adv	☐	☐	☐
absorber	noun	☐	☐	☐
abyss	noun	☐	☐	☐
ac	noun	☑	☑	☑

Rys. 1. Wycinek tabeli z oznaczeniem słów ze stop listy

Poniżej podano wskazówki do wykonania zadania.

4. Przygotować stop listę standardową dla wyrazów swojego słownika – pobrać z zasobu biblioteki NLTK. Sprawdzić zawartość stop listy. Załadować ją do bazy danych. Dalsze operacje realizować w tej bazie.
5. Wygenerować ze słownika wykaz słów spoza stop listy standardowej. Dla tego wykazu opracować autorską stop listę.
6. Połączyć stop listę standardową ze słownikiem swojego korpusu i wprowadzić oznaczenia jak na rys. 1.

Sprawdzić wynik przetwarzania.

Praca z tezaurusem

Opracować dla wyrazów słownika korpusu analizowanych opinii, niebędących stop słowami, odpowiadające im synonimy:

- kolumna *word* zawiera wyrazy ze słownika korpusu z zaznaczeniem słów ze stop listy (w osobnej kolumnie *stop_merge*); kolumny *stop_w* i *stop_n* są pomocnicze i zawierają oznaczenia stop słów odpowiednio: ze standardowej stop listy i stop listy niestandardowej (opracowanej przez studentów),
- kolumna *synonym* zawiera dla wyrazów bliskoznacznych z kolumny *Word* słowa będące ich synonimami.

Zadanie jest realizowane w dwóch etapach (7 i 8).

7. Dla wyrazów zawartych w pliku *Slownik_NVJA.xlsx* niebędących stop słowami zbudować listy ich synonimów pobranych z biblioteki *Wordnet.* Listy tworzą w ww. pliku nową kolumnę *synonyms*. Przeanalizować i wykorzystać program *kod_Python_zadanie7-list_synons.py*. Wynik zapisać w pliku *Slownik_NVJA-synonyms.xlsx* – por. przykładowy wycinek wyniku na rys. 2.

	word	pos	stop_w	stop_n	stop_merge	synonyms
0	i	noun	PRAWDA	FAŁSZ	PRAWDA	[]
1	ford	noun	FAŁSZ	FAŁSZ	FAŁSZ	['Ford_Hermann_Hueffer', 'Gerald_Ford', 'Henry_Ford', 'Ford_
2	year	noun	FAŁSZ	FAŁSZ	FAŁSZ	['class', 'year', 'twelvemonth', 'yr']
3	mean	noun	FAŁSZ	FAŁSZ	FAŁSZ	['think_of', 'miserly', 'average', 'intend', 'mean', 'tight', 'mean_
4	body	noun	FAŁSZ	FAŁSZ	FAŁSZ	['trunk', 'consistency', 'soundbox', 'physical_structure', 'body',
5	friend	noun	FAŁSZ	PRAWDA	PRAWDA	[]
6	copy	noun	FAŁSZ	FAŁSZ	FAŁSZ	['simulate', 'transcript', 'imitate', 'copy', 'written_matter', 'repl
7	cheaply	noun	FAŁSZ	FAŁSZ	FAŁSZ	['inexpensively', 'tattily', 'stingily', 'chintzily', 'cheaply']

Rys. 2. Przykładowy wynik działanie programu *kod_Python_zadanie7a-list_synons.py*

8. Na podstawie zawartości pliku wynikowego z zadania 7 (*Slownik_NVJA-synonyms.xlsx*) utworzyć drugi plik zawierający wszystkie kolumny z ww. pliku z danymi oraz dodatkową kolumnę *synonym* z pojedynczym wyrazem będącym dopasowanym do słowa w kolumnie *word* synonimem. Przeanalizować i wykorzystać program *kod_Python_zadanie8-synonym.py*. Wynik zapisać w pliku *Slownik_NVJA-jeden_synonim.xlsx* – por. przykładowy wycinek wyniku na rys. 3.

   W programie przyjęto, że synonimami będą słowa z kolumny *word* (mapownie 'odwrotne'). Jeżeli jakieś słowa w tej kolumnie są bliskoznacznymi (sprawdzenie w kolumnie *synonyms*) to zostaną zastąpione przez najliczniejsze z nich, a nie ze słownika *wordnet*.

   Zweryfikować jakość dopasowania synonimów do swoich lematów (praca w MS Access lub MS Excel). Dyskusja w grupie studenckiej.
   Opracować dla słów spoza tezaurusa standardowego tezaurus autorski (wewnętrzny). W tym procesie sprawdzać czy można wykorzystać synonimy już istniejące.
   Uzyskany tezaurus domenowy zostanie wykorzystany do uzupełnienia pliku do analizy o kolumnę zawierająca wynik normalizacji leksykalnej przez synonimizację – efekt operacji, w której w miejsce wybranych lematów występujących w każdej opinii są wpisane przygotowane dla nich synonimy.
   Sprawdzić wynik przetwarzania.

Koncepcję tezaurusa przedstawia przykładowy układ danych:

word	pos	synonyms
car	noun	['auto', 'gondola', 'automobile', 'machine', 'railway_car', 'railroad_car', 'elevator_car', 'cable_car', 'car', 'motorcar', 'railcar']
car	noun	['auto', 'gondola', 'automobile', 'machine', 'railway_car', 'railroad_car', 'elevator_car', 'cable_car', 'car', 'motorcar', 'railcar']
auto	noun	['auto', 'automobile', 'machine', 'car', 'motorcar']
auto	noun	['auto', 'automobile', 'machine', 'car', 'motorcar']
auto	noun	['auto', 'automobile', 'machine', 'car', 'motorcar']

Wynik programu:

word	pos	synonyms	synonym
car	noun	['auto', 'gondola', 'automobile', …]	auto
car	noun	['auto', 'gondola', 'automobile', …]	auto
auto	noun	['auto', 'automobile', …]	auto
auto	noun	['auto', 'automobile', …]	auto
auto	noun	['auto', 'automobile', …]	auto

	Unnamed:	word	pos	stop_w	stop_n	top_merg	synonyms	synonym
0	0	i	noun	PRAWDA	FAŁSZ	PRAWDA	[]	
1	1	ford	noun	FAŁSZ	FAŁSZ	FAŁSZ	['Ford_Hermann_Hueffer', 'Gerald_Ford', 'Henry_Fo	ford
2	2	year	noun	FAŁSZ	FAŁSZ	FAŁSZ	['class', 'year', 'twelvemonth', 'yr']	class
3	3	mean	noun	FAŁSZ	FAŁSZ	FAŁSZ	['think_of', 'miserly', 'average', 'intend', 'mean', 'tigh	average
4	4	body	noun	FAŁSZ	FAŁSZ	FAŁSZ	['trunk', 'consistency', 'soundbox', 'physical_structur	body
5	5	friend	noun	FAŁSZ	PRAWDA	PRAWDA	[]	
6	6	copy	noun	FAŁSZ	FAŁSZ	FAŁSZ	['simulate', 'transcript', 'imitate', 'copy', 'written_ma	copy
7	7	cheaply	noun	FAŁSZ	FAŁSZ	FAŁSZ	['inexpensively', 'tattily', 'stingily', 'chintzily', 'cheap	cheaply

Rys. 3. Przykładowy wynik działanie programu *kod_Python_zadanie7b-synonym.py*

Połączenie danych

Podsumowanie przygotowanych do połączenia zasobów danych

- Pierwszy plik: *dane_pos_lema.xlsx* – plik z opiniami, obejmuje kolumny *Nouns*, *Verbs*, *Adj* i *Adv*, które mają postać listy i zawierają słowa kluczowe (odpowiednio: wybrane części mowy) z opinii internautów.
- Drugi plik *Slownik_NVJA-jeden_synonim.xlsx* zawiera słoniki korpusu; stop słowa słownika są zaznaczone wartością PRAWDA w kolumnie *stop_merge.* Słownik korpusu opinii zawiera synonimy wyrazów słownika zapisanymi w kolumnie *synonym* dla tych wyrazów, które nie są na stop liście. W kolumnie *pos* oznaczono, jaką częścią mowy jest wyraz ze słownika.

9. Przeanalizować i wykorzystać program w Pythonie *S2-09_Dane_oczyszczone.py*, który tworzy nowe kolumny: *Nouns_new*, *Verbs_new*, *Adj_new* i *Adv_new* zawierające listy wyrazów z odpowiadających im starych kolumn (*Nouns*, *Verbs* itd.) utworzone następująco: nie ma na listach stop słów, a wyrazy nie będące stop słowami są zastąpione ich synonimami pobranymi z kolumny *synonym* dla kolejnych części mowy. Wynik przetwarzania zawierający wszystkie kolumny z pliku *dane_pos_lema.xlsx* oraz cztery nowopowstałe kolumny są zapisane w pliku *Dane_samochody-oczyszczone.xlsx*.

Sprawdzić wynik przetwarzania.

Dalsze przetwarzanie pliku *Dane_samochody-oczyszczone.xlsx* uwzględnia analizę sentymentu opinii internautów, wyznaczenie chmur słów dla każdej grupy opinii oraz statystyk charakteryzujących te grupy.

Dodatek – symbole Pythona do operacji POS taggingu

W bibliotece NLTK (Natural Language Toolkit) tagowanie części mowy (POS tagging) najczęściej korzysta z zestawu tagów wywodzących się z Penn Treebank Tagset. Poniżej zestawiono najważniejsze symbole POS używane w *nltk.pos_tag*().

Rzeczowniki (Nouns)

- NN – rzeczownik w liczbie pojedynczej (np. *dog*)
- NNS – rzeczownik w liczbie mnogiej (*dogs*)
- NNP – nazwa własna, liczba pojedyncza (*John*)
- NNPS – nazwa własna, liczba mnoga (*Americans*)

Czasowniki (Verbs)

- VB – bezokolicznik (*run*)
- VBD – czas przeszły (*ran*)
- VBG – forma -ing (*running*)
- VBN – imiesłów bierny (*eaten*)
- VBP – czas teraźniejszy (bez 3 os. l.poj.) (*run*)
- VBZ – czas teraźniejszy (3 os. l.poj.) (*runs*)

Przymiotniki (Adjectives)

- JJ – przymiotnik (*big*)
- JJR – stopień wyższy (*bigger*)
- JJS – stopień najwyższy (*biggest*)

Przysłówki (Adverbs)

- RB – przysłówek (*quickly*)
- RBR – stopień wyższy (*faster*)
- RBS – stopień najwyższy (*fastest*)

Przyimki i spójniki

- IN – przyimek lub spójnik (in, on, because)
- CC – spójnik współrzędny (and, but)

Liczebniki

- CD – liczebnik (*one, 2024*)

Zaimki

- PRP – zaimek osobowy (*he, she*)
- PRP**$** – zaimek dzierżawczy (*his, her*)

Determinanty i inne

- DT – rodzajnik (*the, a*)
- PDT – predeterminer (*all, both*)

Wh- słowa (pytania)

- WDT – zaimek względny (*which*)
- WP – zaimek pytający (*who*)
- WP**$** – dzierżawczy (*whose*)
- WRB – przysłówek pytający (*where, when*)

Inne ważne

- TO – „to” (np. *to go*)
- MD – czasownik modalny (*can, must*)
- EX – egzystencjalne „there”
- FW – wyraz obcy
- LS – marker listy
- UH – wykrzyknik (*oh, wow*)