Informatyka, studia dzienne, I st. semestr VI

Komputerowe systemy rozpoznawania **2018/2019**
Prowadzący: dr hab. inż. Adam Niewiadomski poniedziałek, 12:15

Data oddania: _____ Ocena: _____

Stanisław Zakrzewski 210360
Maciej Socha 210321

Zadanie 1: Ekstrakcja cech, miary podobieństwa, klasyfikacja

1. Cel

Celem zadania było poznanie oraz zaimplementowanie różnych metod ekstrakcji cech z tekstów, określania podobieństwa oraz klasyfikacji tekstów.

2. Wprowadzenie

Celem projektu jest stworzenie programu pozwalającego na klasyfikację wybranego zbioru elementów. Klasyfikatorem wybranym do tego celu jest metoda k-najbliższych sąsiadów.

Algorytm k najbliższych sąsiadów, nazywamy też potocznie algorytmem knn, pozwala na klasyfikację zbioru wieloelementowego według określonych etykiet. Na początku działania algorytmu k najbliższych sąsiadów określane są wektory dla każdego z elementów podlegających klasyfikacji. W naszym przypadku określanie wektorów polega na odpowiednim przetworzeniu tekstu zawierającego się w elementach zbioru do klasyfikacji. Następnie wektory są umieszczane na przestrzeni n elementowej, gdzie n stanowi liczebność elementów w wektorze. Odsłaniane są etykiety, domyślnie 10% dla każdej z etykiet. Odsłonięcie etykiet stanowi jeden ze sposobów rozwiązania problemu zimnego startu. Następnie kolejne etykiety są nadawane kolejnym elementom, poprzez znalezienie k najbliższych elementów i wybranie spośród etykiet należących do danych elementów tych, które są najliczniejsze, w przypadku identycznej liczebności etykiet wybierana jest ta, której średnia odległość do aktualnie klasyfikowanego elementu jest mniejsza.

Do wytworzenia wektora cech stosowane są dwa warianty ekstrakcji cech typu Dictionary Matching (DM). W obu przypadkach teksty znajdujące się w artykułach są początkowo poddane procesowi lemmatyzacji. Proces lemmatyzacji jest to czynność mająca na celu znalezienia lemmy dla danego słowa, lemma jest to forma podstawowa wyrazu w obszarze części mowy, którą reprezentuje. Następnie następuje proces przyznawania punktów dla poszczególnych słów. Wybierane zostają słowa mające najwięcej punktów. Pierwszy z na początku usuwa wszystkie słowa znajdujące się na przygotowanej wcześniej stop-liście, usuwa wartości liczbowe oraz zwiększa punktację słów znajdujących się bliżej początku tekstu. Drugi sposób bazuje natomiast na algorytmie TFIDF oraz również usuwane są wszelkie wartości liczbowe.

Algorytm TFIDF jest jedną z metod obliczania wagi słów w oparciu o liczbę ich wystąpień. Jest on stosowany między innymi w wyszukiwarkach internetorych. Jest on obliczany przy pomocy wzoru:

$$(tf - idf)_{i,j} = tf_{i,j} \times idf_i$$

gdzie $tf_{i,j}$ to tak zwany „term frequency" opisany wzorem:

$$tf_{i,j} = log \frac{n_{i,j}}{\sum_k n_{k,j}}$$

gdzie: $n_{i,j}$ jest liczbą wystąpień termu (t_i) w dokumencie d_j, a mianownik jest dumą liczby wystąpień wszystkich termów w dokumencie $d_j \cdot idf_i$ to „inverse document frequency" wyraża się wzorem:

$$idf_i = log \frac{|D|}{|\{d : t_i \epsilon d\}|}$$

gdzie: $|D|$ - liczba dokumentów w korpusie $|\{d : t_i \epsilon d\}|$ - liczba dokumentów zawierających przynajmniej jedno wystąpienie danego termu.

Powstałe w wyniku działania obu ekstraktorów cech wektory są używane do wytworzenia wektorów liczbowych pozwalających na umieszczenie elementów w przestrzeni liczbowej, co jest wymagane w algorytmie k najbliższych sąsiadów.

Obliczenia odległości dokonano w trzech metrykach.

Pierwszą z nich jest metryka Euklidesa, odległość d obliczana jest przy pomocy wzoru:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2}$$

Drugą z nich jest metryka Manhattana, nazywana również metryką uliczną, taksówkarską lub miejską. Odległość jest obliczana przy pomocy wzoru:

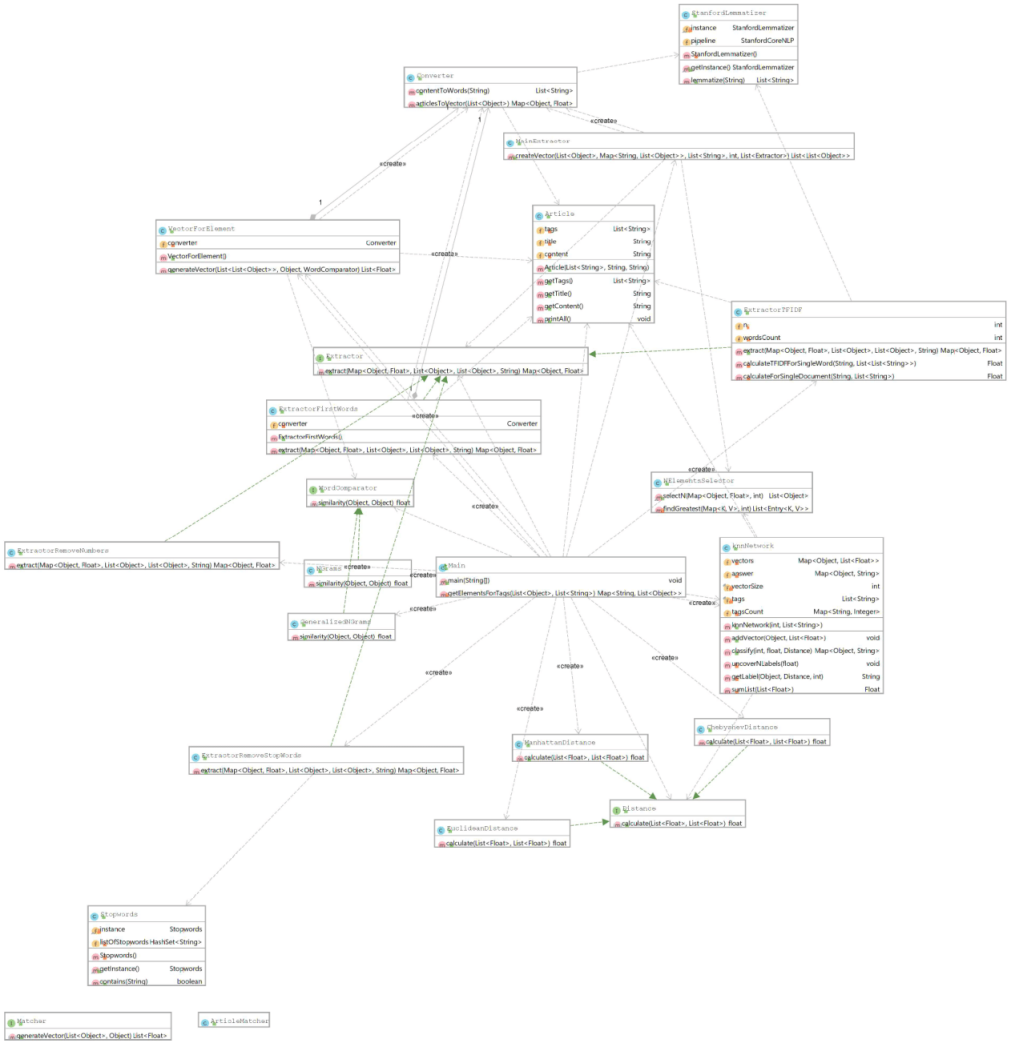
$$d(x, y) = \sum |x_i - y_i|$$

Trzecią i zarazem ostatnią jest metryka Czebyszewa, odległość jest obliczana przy pomocy wzoru:

$$d(x, y) = \frac{\sum |x_i \cdot y_i|}{\sqrt{\sum x_i^2 \cdot \sum y_i^2}}$$

3. Opis implementacji

Algorytmy zostały zaimplementowane w języku Java w wersji 11. Dodatkowo na potrzemy procesu lemmatyzacji wykorzystano, udostępnioną przez Stanford Natural Language Processing Group, biliotekę CoreNLP w wersji 3.9.2. Biblioteka ta jest udostępniona z licencją GNU General Public License v3 co pozwala nam korzystać z niej w naszym programie. Biblioteka ta jest bardzo obszerna, w naszym programie wykorzystujemy jedynie funkcjonalność lemmatyzacji. Jest ona zaimplementowana w klasie StanfordLemmatizer. Implementacja tej klasy została bezpośrednio zaczerpnięta z dokumentacji[5]. Poniżej przedstawiono uproszczony diagram klas. Zaznaczone zostały na nim kluczowe dla działania naszego programu klasy.



Rysunek 1. UML Diagram

Klasa Article odpowiada za przechowywanie informacji niezbędnych do działania programu. Wykorzystujemy ją zarówno do przetwarzania artukułów zawartych w zbiorze danych reuters jak i zestawu artykułów przygotowanego

przez nas samych.
Interfejs Extractor służy i znajdująca się w nim metoda extract jest wykorzystywana przy procesie ekstakcji cech. Implementują ją liczne klasy zawierające sie w dwóch sposobach ekstarakcji cech zawartych w programie.
Klasa knnNetwork zawiera w sobie implementację algorytmu k najbliższych sąsiadów do ustalania przynależności wektorów odpowiadającycm przekazanym do programu elementom. Klasa pozwala na dodawanie wektorów wraz z odpowiadającymi im elementami, a następnie klasyfikowanie ich przy przekazaniu odpowiedniego parametru k oznaczającego liczbę sąsiadów, uncoveredLabelFraction zapomocą którego przekazujemy jaka część tekstów będzie miała odkryte etykiety oraz distance, metrykę obliczania dystansu pomiędzy wektorami.
Pakiet calculatedistance zawiera w sobie interfejs Distance oraz implementujące go klasy ChebyshevDistance(metryka Czebyszewa), EuclideanDistance(metryka Euklidesa) oraz ManhattanDistance(metryka uliczna). Są to wymagane przez treść zadania metryki pomiaru odległości pomiędzy wektorami.
Za przekazywanie danych do programu odpowiada plik config.txt zawierający w sobie wszystkie potrzebne do działania programu parametry. Są to odpowiednio:

1. tagClass - tag dla którego etykiety będzie nadawał program
2. folderPath - ścieżka do folderu z plikami z danymi
3. articlesToReadCount - liczba plików z artykułami, które program ma wczytać
4. k -
5. fractionOfUncoveredForEachTag -
6. tags - etykiety, według których program ma klasyfikować
7. numberOfElementsPerTag - liczba elementów jakie ma zawierać w sobie cecha dla każdej z etykiet
8. trainToTestRatio - stosunek zbioru treningowego do testowego
9. distanceKNN - metryka pomiaru dystansu w przestrzeni dla algorytmu knn
10. wordSimilarity - metryka podobieństwa słów
11. extractors - zestaw ekstraktorów

4. Materiały i metody

Klasyfikacja tekstów Reutersa oraz danych zebranych przez nas została wykonana dla obu sposobów ekstrakcji cech i dla każdej z 3 metryk obliczania dystansu w algorytmie k najbliższych sąsiadów. Dla parametru k wybrano niektóre wartości ze zbioru (3, 5, 8, 11, 19), tak aby jak najlepiej ukazać właściwości każdego z sposobów ekstrakcji oraz najlepiej dopasować je do zbioru danych testowych. Klasyfikacja na zbiorze Reutersa według tagu PLACES została przeprowadzona dla sześciu etykiet (west-germany, usa, uk, canada, france, japan), przy stosunku zbioru treningowego do testowego 60-40 i dla 10% odkrytych etykiet,
DODANIE OPISU DRUGIEGO TAGA
natomiast w naszych tekstach klasyfikacja nastąpiła według tagu REVIEWS

dla dwóch etykiet (movie, restaurant), równego podziału zbioru elementów na część treningową i testową oraz 20% odkrytych etykiet.

5. Wyniki

Wyniki kolejnych przeprowadzanych eksperymentów zostały umieszczone w tabelach poniżej. Początkowa konfiguracja programu znajduje się poniżej, przy kolejnych eksperymentach zostały wspominanie tylko te parametry które były zmienione.

1. tagClass = PLACES
2. articlesToReadCount = 23
3. k = 3
4. fractionOfUncoveredForEachTag = 0.1
5. tags = west-germany, usa, france, uk, canada, japan
6. numberOfElementsPerTag = 5
7. trainToTestRatio = 0.6
8. distanceKNN = euclidean
9. wordSimilarity = NGrams
10. extractors = 1

Poniżej znajdują się wyniki dla przedstawionych powyższej parametrów.

Label	Precision	Recall
west-germany	0.39534885	0.3923077
usa	0.83159405	0.9416126
france	0.5064935	0.35779816
uk	0.62025315	0.24873096
canada	0.49079755	0.23738873
japan	0.5652174	0.23636363

Tablica 1: Precision i Recall dla parametrów bazowych

Multi-Class Pecision: 0.7962223

	west-germany	usa	france	uk	canada	japan
west-germany	51	74	0	2	3	0
usa	57	4064	31	57	71	36
france	1	65	39	0	2	2
uk	10	273	5	98	7	1
canada	5	248	2	1	80	1
japan	5	163	0	0	0	52

Tablica 2: Przypisanie tagów dla parametrów bazowych

5.1. Eksperymento 1 - Różne metryki odległości

Metryka euklidesowa

Label	Precision	Recall
west-germany	0.39534885	0.3923077
usa	0.83159405	0.9416126
france	0.5064935	0.35779816
uk	0.62025315	0.24873096
canada	0.49079755	0.23738873
japan	0.5652174	0.23636363

Tablica 3: Precision i Recall dla metryki euklidesowej

Multi-Class Pecision: 0.7962223
Metryka uliczna

Label	Precision	Recall
west-germany	0.5566038	0.4402985
usa	0.8411453	0.914457
france	0.32407406	0.2651515
uk	0.45026177	0.22994652
canada	0.45614034	0.31610942
japan	0.53797466	0.425

Tablica 4: Precision i Recall dla metryki ulicznej

Multi-Class Pecision: 0.78732294
Metryka Czebyszewa

Label	Precision	Recall
west-germany	0.64788735	0.35384616
usa	0.8515893	0.91464823
france	0.37195122	0.5126051
uk	0.4652778	0.17539267
canada	0.52517986	0.43843845
japan	0.49222797	0.4589372

Tablica 5: Precision i Recall dla metryki Czebyszewa

Multi-Class Pecision: 0.7954958

5.2. Eksperymento 2 - różne metryki porównywania słów

NGramy, n = 3

Label	Precision	Recall
west-germany	0.39534885	0.3923077
usa	0.83159405	0.9416126
france	0.5064935	0.35779816
uk	0.62025315	0.24873096
canada	0.49079755	0.23738873
japan	0.5652174	0.23636363

Tablica 6: Precision i Recall dla NGramów, n = 3

Multi-Class Pecision: 0.7962223
NGramy, n = 2

Label	Precision	Recall
west-germany	0.5416667	0.104
usa	0.80586374	0.94232106
france	0.22972973	0.16037735
uk	0.26666668	0.12276215
canada	0.4347826	0.16574585
japan	0.47619048	0.09756097

Tablica 7: Precision i Recall dla NGramów, n = 2

Multi-Class Pecision: 0.7675263
NGramy, n = 4

Label	Precision	Recall
west-germany	0.5652174	0.36879432
usa	0.847703	0.9302486
france	0.36619717	0.23853211
uk	0.41411042	0.34526855
canada	0.6785714	0.23312883
japan	0.6884058	0.4871795

Tablica 8: Precision i Recall dla NGramów, n = 4

Multi-Class Pecision: 0.80366874
Uogólnione NGramy

Label	Precision	Recall
west-germany	0.7589286	0.65891474
usa	0.8819797	0.9648311
france	0.46666667	0.7241379
uk	0.7490196	0.49354005
canada	1.0	0.099415205
japan	0.76649743	0.7190476

Tablica 9: Precision i Recall dla UogólnionychNGramów

Multi-Class Pecision: 0.85633856

6. Dyskusja

Sekcja ta powinna zawierać dokładną interpretację uzyskanych wyników eksperymentów wraz ze szczegółowymi wnioskami z nich płynącymi. Najcenniejsze są, rzecz jasna, wnioski o charakterze uniwersalnym, które mogą być istotne przy innych, podobnych zadaniach. Należy również omówić i wyjaśnić wszystkie napotakane problemy (jeśli takie były). Każdy wniosek powinien mieć poparcie we wcześniej przeprowadzonych eksperymentach (odwołania

do konkretnych wyników). Jest to jedna z najważniejszych sekcji tego sprawozdania, gdyż prezentuje poziom zrozumienia badanego problemu.

7. Wnioski

W tej, przedostatniej, sekcji należy zamieścić podsumowanie najważniejszych wniosków z sekcji poprzedniej. Najlepiej jest je po prostu wypunktować. Znów, tak jak poprzednio, najistotniejsze są wnioski o charakterze uniwersalnym.

Literatura

[1] David D. Lewis. *Feature Selection and Feature Extract ion for Text Categorization*, University of Chicago,
Dostępny w Internecie: `https://aclweb.org/anthology/H92-1041?fbclid=IwAR248ftiyFqXrFpi51IDLorT7Ngso369BPTOaOeSYE3QGG1gYD9TNfy58qc`

[2] David Dolan Lewis. *Representation and learning in information retrieval*, University of Massachusetts,
Dostępny w Internecie: `http://ciir.cs.umass.edu/pubfiles/UM-CS-1991-093.pdf`

[3] David D. Lewis. *Data Extraction as Text Categorization : An Experiment With the MUC-3 Corpus*, University of Chicago,
Dostępny w Internecie: `https://www.aclweb.org/anthology/M91-1035`

[4] Marina Sokolova, Guy Lapalme. *A systematic analysis of performance measures for classification tasks*, Information Processing and Management no 45,
Dostępny w internecie: `http://rali.iro.umontreal.ca/rali/sites/default/files/publis/SokolovaLapalme-JIPM09.pdf?fbclid=IwAR2M7_a4QxL_F4yCOB_Akp4ghkoUKrBnHT9xzCfuTcoVrLBe3lN3kIlPt00`

[5] `https://stanfordnlp.github.io/CoreNLP`