

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 określone 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. Odślaniane są etykiety, domyślnie 10% dla każdej z etykiet. Odślonięcie 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 lematyzacji. Proces lematyzacji 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 internetowych. 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:

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

gdzie: $|D|$ - liczba dokumentów w korpusie $|\{d : t_i \in d\}|$ - 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:

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

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

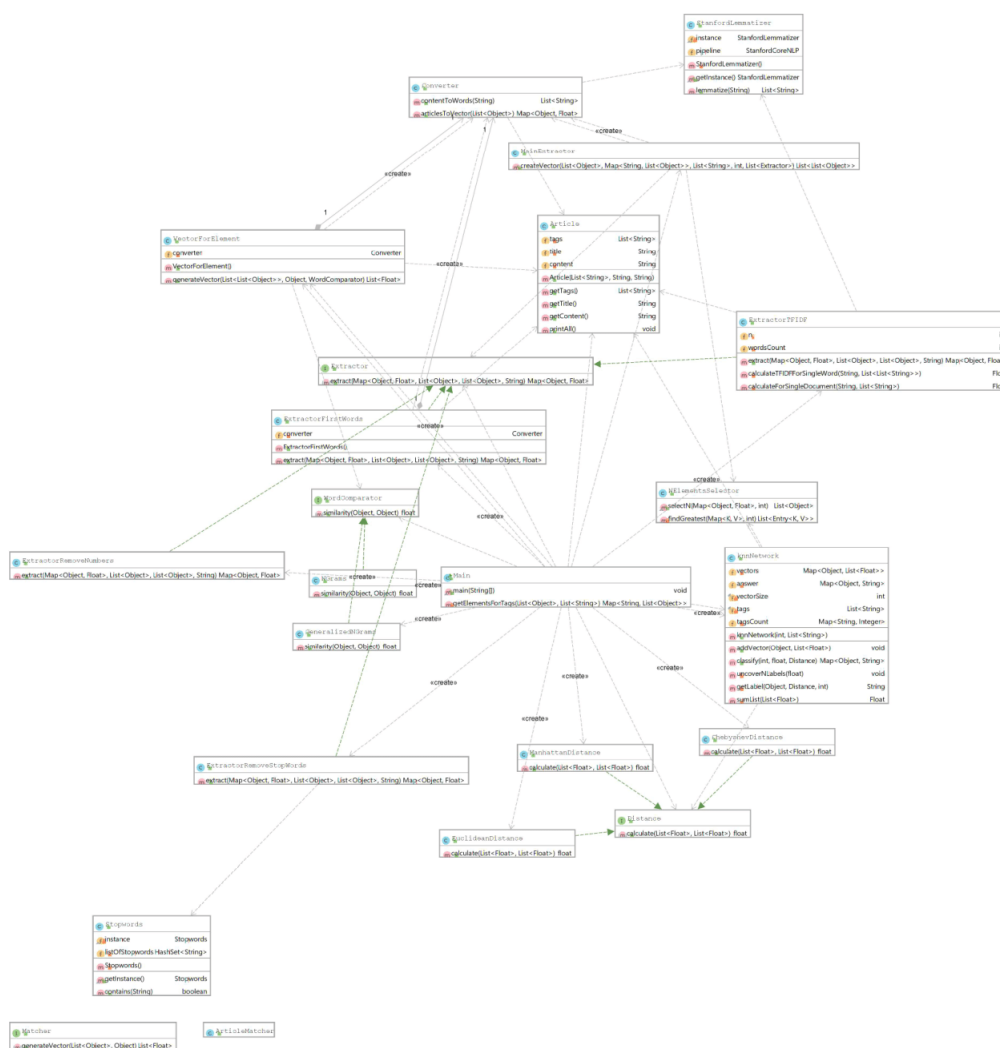
$$d(x, y) = \sum |x_i - y_i|$$

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 potrzeby procesu lematyzacji wykorzystano, udostępnioną przez Stanford Natural Language Processing Group, bibliotekę 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 lematyzacji. 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 artykułów 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 ekstrakcji cech. Implementują ją liczne klasy zawierające się w dwóch sposobach ekstrakcji cech zawartych w programie.

Klasa `knnNetwork` zawiera w sobie implementację algorytmu k najbliższych sąsiadów do ustalania przynależności wektorów odpowiadającym 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` za pomocą 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

Należy tu zamieścić krótki i zwięzły opis zaprojektowanych klas oraz powiązań między nimi. Powinien się tu również znaleźć diagram UML (diagram klas) prezentujący najistotniejsze elementy stworzonej aplikacji. Należy także podać, w jakim języku programowania została stworzona aplikacja.

4. Materiały i metody

W tym miejscu należy opisać, jak przeprowadzone zostały wszystkie badania, których wyniki i dyskusja zamieszczane są w dalszych sekcjach. Opis ten powinien być na tyle dokładny, aby osoba czytająca go potrafiła wszystkie przeprowadzone badania samodzielnie powtórzyć w celu zweryfikowania ich poprawności (a zatem m.in. należy zamieścić tu opis architektury sieci, wartości współczynników użytych w kolejnych eksperymentach, sposób ini-

cjalizacji wag, metodę uczenia itp. oraz informacje o danych, na których prowadzone były badania). Przy opisie należy odwoływać się i stosować do opisanych w sekcji drugiej wzorów i oznaczeń, a także w jasny sposób opisać cel konkretnego testu. Najlepiej byłoby wyraźnie wyszczególnić (ponumerować) poszczególne eksperymenty tak, aby łatwo było się do nich odwoływać dalej.

5. Wyniki

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

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: Parametry bazowe

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 napotkane 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. Znow, tak jak poprzednio, najistotniejsze są wnioski o charakterze uniwersalnym.

Literatura

- [1] David D. Lewis. *Feature Selection and Feature Extraction for Text Categorization*, University of Chicago,
Dostępny w Internecie: <https://aclweb.org/anthology/H92-1041?fbclid=IwAR248ftiyFqXrFpi51IDLorT7Ngso369BPT0a0eSYE3QGG1gYD9TNfy58qc>
- [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>