Informatyka, studia dzienne, I st. semestr VI

Komputerowe systemy rozpoznawania **2019/2020**

Prowadzący: dr hab. inż. Adam Niewiadomski prof. uczelni pon., 12:15

Data oddania: _____ Ocena: _____

Mateusz Walczak 216911
Konrad Kajszczak 216790

Zadanie 1: Ekstrakcja cech, miary podobieństwa, klasyfikacja*

1. Cel

Celem zadania było stworzenie aplikacji służącej do klasyfikacji artykułów prasowych metodą k-NN. Korzystając z różnych metod wyboru słów kluczowych i ekstrakcji wektorów cech oraz istniejących miar podobieństwa, należało porównać przypisane przez naszą aplikacje kategorie artykułów do tych faktycznych. Należało również podjąć próbę opracowania własnej miary podobieństwa i/lub metryki.

2. Wprowadzenie

Algorytm k najbliższych sąsiadów jest bardzo prostym klasyfikatorem probabilistycznym. Niekiedy mówi się, że algorytm k-NN jest leniwy. Wynika to z faktu, że nie tworzy on wewnętrznej reprezentacji danych treningowych (uczących), ale ropoczyna poszukiwanie rozwiązania dopiero podczas analizy konkretnego wzorca ze zbioru testowego.

Algorytm przechowuje zbiór wszystkich wzorców uczących, względem których obliczana jest odległość wzorca testowego, zdefiniowana poprzez odpowiednią metrykę. Następnie algorytm wybiera k wzorców treningowych, nazywanych sąsiadami, do których aktualnie badany wzorzec testowy ma

* SVN: `https://github.com/Walducha1908/KSR1`

najmniejszą odległość. Ostateczny rezultat - kategoria, do której zostanie przypisany analizowany wzorzec - stanowi najczęściej występująca kategoria wśród k najbliższych sąsiadów.

2.1. Metryki

Do obliczenia odległości pomiędzy tekstami posłużyliśmy się następującymi metrykami:

- Metryka Euklidesowa - w celu obliczenia odległości $d_e(x, y)$ między dwoma punktami x, y należy obliczyć pierwiastek kwadratowy z sumy kwadratów różnic wartości współrzędnych o tych samych indeksach, zgodnie ze wzorem:
$$d_e(x, y) = \sqrt{(y_1 - x_1)^2 + \cdots + (y_n - x_n)^2} \tag{1}$$
- Metryka uliczna (Manhattan, miejska) - w celu obliczenia odległości $d_m(x, y)$ między dwoma punktami x, y należy obliczyć sumę wartości bezwzględnych różnic współrzędnych punktów x oraz y, zgodnie ze wzorem:
$$d_m(x, y) = \sum_{k=1}^{n} |x_k - y_k| \tag{2}$$
- Metryka Czebyszewa - w celu obliczenia odległości $d_{ch}(x, y)$ między dwoma punktami x, y należy obliczyć maksymalną wartość bezwzględnych różnic współrzędnych punktów x oraz y, zgodnie ze wzorem:
$$d_{ch}(x, y) = \max_i |x_i - y_i| \tag{3}$$
- Metryka Hamminga - definiujemy jako ilość różnic pomiędzy dwoma wektorami o tej samej długości. Aby obliczyć odległość $d_h(x, y)$ między dwoma punktami x, y należy posłużyć się wzorem [1] :
$$d_h(x, y) = \sum_{i=1}^{n} |h(i)|, \tag{4}$$
gdzie
$$h(i) = \begin{cases} 0 & \text{jeśli } v_{1i} = v_{2i} \\ 1 & \text{w przeciwnym wypadku} \end{cases} \tag{5}$$
- Odległość Canberra - ważona wersja metryki ulicznej, aby obliczyć odległość $d_c(x, y)$ między dwoma punktami x, y należy posłużyć się wzorem:
$$d_c(x, y) = \sum_i \frac{|x_i - y_i|}{|x_i| + |y_i|} \tag{6}$$

2.2. Miary

Klasyfikację metodą KNN przeprowadzono również z wykorzystaniem następujących miar podobieństwa:

- Term Frequency Matrix - czyli "macierz częstości występowania terminów". Określa podobieństwo dokumentów ze względu na wybrany zbiór terminów, np. słów kluczowych [1].

- Metoda n-gramów - metoda ta określa podobieństwo łańcuchów tekstowych s_1, s_2 w oparciu o ilość wspólnych podciągów n-elementowych. W naszym przypadku rozpatrujemy $n = 3$, czyli trigramy. Formuła opsiująca trigramy jest następująca:

$$sim_3(s_1, s_2) = \frac{1}{N-2} \sum_{i=1}^{N-2} h(i), \tag{7}$$

gdzie N stanowi liczbę elementów dłuższego z łańcuchów s_1, s_2:

$$N = max\{N(s_1), N(s_2)\}, \tag{8}$$

zaś $h(i) = 1$ jeśli 3-elementowy podciąg zaczynający się od i-tej pozycji w s_1 występuje przynajmniej raz w s_2, w przeciwnym przypadku $h(i) = 0$ [1].

2.3. Wyznaczanie słów kluczowych

Aby wyznaczyć słowa kluczowe posługujemy się poniższą metodą:

- Term frequency - metoda polegająca na zliczeniu liczby wystąpień danego słowa we wszystkich dokumentach.

Przeprowadzamy obliczenia na zbiorze wszystkich posiadanych danych (w naszym przypadku na wszystkich artykułach) i otrzymujemy zestaw par - słowo i wartość. Taki zestaw par sortujemy malejąco po wartości i wybieramy n pierwszych słów. Wybrane n słów staje się słowami kluczowymi.

Taki schemat powtarzamy l razy, gdzie l jest liczbą kategorii na jakie klasyfikujemy. Ostatecznie otrzymujemy l zestawów słów kluczowych, przy czym każdy zestaw reprezentuje inną kategorię. Otrzymane zbiory słów kluczowych oznaczamy:

$$K_1, K_2, \ldots, K_{l-1}, K_l. \tag{9}$$

Otrzymany zbiór słów kluczowych będziemy używać we wszystkich iteracjach programu. Słowa kluczowe będą niezmienne, a wszystkie przeprowadzone przez nas eksperymenty będą bazowały na tym samym zbiorze słów kluczowych.

2.4. Wyznaczanie ważonych słów kluczowych

W celach poprawienia jakości klasyfikacji wprowadzono "ważone słowa kluczowe". Tak nazwaliśmy zestaw par - słowo kluczowe i waga (wartość zmiennoprzecinkowa), z wykorzytsaniem których przeprowadziliśmy takie same eksperymenty jak z wykorzystaniem "zwykłych" słów kluczowych, opisanych w poprzednim podpunkcie.

Ważone słowa kluczowe to nic innego jak obliczony wcześniej, ten sam zestaw słów, jednak ubogacony o wagę, obliczaną zgodnie z opracowanym przez nas wzorem:

$$W_i = \left(1 - \frac{N_{W_i \in K_l}}{l-1}\right)^2, \tag{10}$$

gdzie W_i - waga i-tego słowa kluczowego, l - liczba kategorii, $N_{W_i \in K_l}$ - liczba kategorii słów kluczowych (innych od swojej własnej), w których i-te słowo kluczowe występuje.

Dla jasności przeanalizujemy przykład. Niech $l = 3$, a obliczone słowa kluczowe mają postać:

$$K_1 = \{"jesien", "ogon", "krowa"\}, \tag{11}$$

$$K_2 = \{"wiosna", "ogon", "pies"\}, \tag{12}$$

$$K_3 = \{"lato", "ogon", "krowa"\}, \tag{13}$$

Obliczmy wartości wag dla wybranych słów kluczowych z powyższego zestawu. Dla słowa "jesien" otrzymamy następującą wartość:

$$W_{jesien} = \left(1 - \frac{0}{2}\right)^2 = 1, \tag{14}$$

słowo "jesien" wystąpiło tylko w jednej, "swojej" kategorii, ma zatem największą możliwą wagę.

Dla słowa "krowa":

$$W_{krowa} = \left(1 - \frac{1}{2}\right)^2 = 0.25, \tag{15}$$

słowo "krowa" wystąpiło w jednej dodatkowej kategorii (łącznie w dwóch).

Dla słowa "ogon":

$$W_{ogon} = \left(1 - \frac{2}{2}\right)^2 = 0, \tag{16}$$

słowo "ogon" wystąpiło we wszystkich kategoriach, dlatego też uznajemy, że nie ma dla nas żadnego znaczenia, jego waga jest równa 0.

Z powyższych rozważań bardzo jasno wynika, że wagi słów kluczowych mogą osiągać wartości z przedziału $\langle 0; 1 \rangle$.

2.5. Cechy poddawane ekstrakcji

Ekstrakcja cech charakterystycznych tekstu - w tym celu tworzymy wektor cech, który opisuje tekst (w naszym przypadku artykuł) na podstawie konkretnych, zdefiniowanych cech. Poniżej znajduje się opis wszystkich cech użytych w doświadczeniu.

Przed ekstrakcją cech, tekst został odpowiednio przygotowany. Z artykułów usunięte zostały nic nie wnoszące słowa (z tzw. "stop" listy), tekst został poddany stemizacji oraz pozbawiony znaków interpunkcyjnych.

Przyjęto następujące oznaczenia:
T_i - zbiór słów do badania,

K - stały zbiór słów kluczowych[1],
$N_{K \in T}$ - liczba wystąpień elementów zbioru K w zbiorze T[2],
$C_i(T, K)$ - wartość funkcji cechy.

2.5.1. Liczba wystąpień wszystkich słów kluczowych w całym artykule

Cecha opisująca liczbę słów kluczowych, które występują w całej sekcji głównej artykułu (body).

$$C_1(T_1, K) = N_{K \in T_1}, \tag{17}$$

gdzie T_1 - zbiór słów sekcji głównej artykułu.

Przeanalizujmy przykład obliczania wartości cechy C_1. Niech zbiór słów kluczowych K ma postać:

$$K = \{"wirus", "choroba", "zaraz", "anihilacja"\}, \tag{18}$$

zaś zbiór słów do badania (zbiór słów sekcji głównej badanego artykułu testowego) T_1 prezentuje się następująco:

$$\begin{aligned} T_1 = \{&"wirus", "niszczy", "wszystko", "droga", \\ &"zaraz", "wirus", "powodowac", "choroba"\}, \end{aligned} \tag{19}$$

Najpierw w wariancie pierwszej metody ekstrakcji - wykorzystując metodę TF i zwykłe słowa kluczowe. Przeanalizujmy występowanie elementów zbioru K w zbiorze T_1:

- "wirus" - występuje 2 razy,
- "choroba" - występuje 1 raz,
- "zaraz" - występuje 1 raz,
- "anihilacja" - nie występuje ani razu.

Po dodaniu wszystkich wystąpień otrzymujemy:

$$C_1(T_1, K) = N_{K \in T_1} = 2 + 1 + 1 + 0 = 4. \tag{20}$$

Teraz zajmijmy się drugą metodą ekstrakcji - wykorzystując ważone słowa kluczowe. Załóżmy, że pary słów kluczowych wraz z obliczonymi wagami dla słów kluczowych zbioru K prezentują się następująco:

$$K_w = \{("wirus", 0.25), ("choroba", 1), ("zaraz", 0), ("anihilacja", 1)\}, \tag{21}$$

W tym przypadku zgodnie z wcześniej zaprezentowanym opisem, musimy obliczyć sumę iloczynów liczby wystąpień poszczególnych elementów zbioru K w zbiorze T_1 i odpowiadających im wag:

$$C_1(T_1, K_w) = N_{K \in T_1} = 2 \cdot 0.25 + 1 \cdot 1 + 1 \cdot 0 + 0 \cdot 1 = 0.5 + 1 + 0 + 0 = 1.5. \tag{22}$$

[1] Na który składają się zbiory $K_1, K_2, \ldots, K_{l-1}, K_l$.

[2] W przypadku ważonych słów kluczowych będzie to suma iloczynów liczby wystąpień poszczególnych elementów zbioru K w zbiorze T i odpowiadających im wag.

2.5.2. Liczba wystąpień wszystkich słów kluczowych w tytule artykułu

Cecha opisująca liczbę słów kluczowych, które występują w tytule artykułu (title).

$$C_2(T_2, K) = N_{K \in T_2}, \tag{23}$$

gdzie T_2 - zbiór słów tytułu artykułu.

2.5.3. Liczba wystąpień wszystkich słów kluczowych w sekcji daty artykułu

Cecha opisująca liczbę słów kluczowych, które występują w sekcji daty artykułu (dateline).

$$C_3(T_3, K) = N_{K \in T_3}, \tag{24}$$

gdzie T_3 - zbiór słów sekcji daty artykułu.

2.5.4. Stosunek liczby wystąpień wszystkich słów kluczowych do ogólnej liczby słów w artykule

Cecha opisująca stosunek liczby słów kluczowych, które występują w całej sekcji głównej artykułu (body), do całkowitej liczby słów występujących w części głównej.

$$C_4(T_4, K) = \frac{N_{K \in T_4}}{|T_4|}, \tag{25}$$

gdzie T_4 - zbiór słów sekcji głównej artykułu, $|T_4|$ - liczba elementów (słów) zbioru sekcji głównej artykułu.

W tym miejscu warto wspomnieć, że w przypadku ważonych słów kluczowych wartość $|T_4|$ będzię iloczynem liczby elementów zbioru sekcji głównej artykułu i maksymalnej wartości osiągalnej przez wagi. Jednak ponieważ maksymalną możliwą wartością wagi słowa kluczowego jest 1 (zgodnie z rozdziałem 2.3) to w obu przypadkach - zwykłych słów kluczowych jak i ważonych słów kluczowych - będzie to dokładnie ta sama wartość liczbowa.

2.5.5. Liczba wystąpień wszystkich słów kluczowych w pierwszych 50 słowach artykułu

Cecha opisująca liczbę słów kluczowych, które występują w pierwszych 50 słowach sekcji głównej artykułu. Jeśli artykuł jest krótszy niż 50 słów to bierzemy pod uwagę wszystkie występujące w nim słowa.

$$C_5(T_5, K) = N_{K \in T_5}, \tag{26}$$

gdzie T_5 - pierwsze 50 słów sekcji głównej artykułu.

2.5.6. Liczba wystąpień wszystkich słów kluczowych w pierwszych 10% artykułu

Cecha opisująca liczbę słów kluczowych, które występują w pierwszych 10% sekcji głównej artykułu.

$$C_6(T_6, K) = N_{K \in T_6}, \tag{27}$$

gdzie T_6 - pierwsze 10% słów sekcji głównej artykułu.

2.5.7. Liczba wystąpień wszystkich słów kluczowych w pierwszych 20% artykułu

Cecha opisująca liczbę słów kluczowych, które występują w pierwszych 20% sekcji głównej artykułu.

$$C_7(T_7, K) = N_{K \in T_7}, \tag{28}$$

gdzie T_7 - pierwsze 20% słów sekcji głównej artykułu.

2.5.8. Liczba wystąpień wszystkich słów kluczowych w pierwszych 50% artykułu

Cecha opisująca liczbę słów kluczowych, które występują w pierwszych 50% sekcji głównej artykułu.

$$C_8(T_8, K) = N_{K \in T_8}, \tag{29}$$

gdzie T_8 - pierwsze 50% słów sekcji głównej artykułu.

2.5.9. Liczba wystąpień wszystkich słów kluczowych w pierwszym paragrafie

Cecha opisująca liczbę słów kluczowych, które występują w pierwszym paragrafie sekcji głównej artykułu.

$$C_9(T_9, K) = N_{K \in T_9}, \tag{30}$$

gdzie T_9 - pierwszy paragraf sekcji głównej artykułu.

2.5.10. Liczba wystąpień wszystkich słów kluczowych w ostatnich 50 słowach artykułu

Cecha opisująca liczbę słów kluczowych, które występują w ostatnich 50 słowach sekcji głównej artykułu. Jeśli artykuł jest krótszy niż 50 słów to bierzemy pod uwagę wszystkie występujące w nim słowa.

$$C_{10}(T_{10}, K) = N_{K \in T_{10}}, \tag{31}$$

gdzie T_{10} - ostatnie 50 słów sekcji głównej artykułu.

2.5.11. Liczba wystąpień wszystkich słów kluczowych w ostatnich 10% artykułu

Cecha opisująca liczbę słów kluczowych, które występują w ostatnich 10% sekcji głównej artykułu.

$$C_{11}(T_{11}, K) = N_{K \in T_{11}}, \tag{32}$$

gdzie T_{11} - ostatnie 10% słów sekcji głównej artykułu.

2.5.12. Liczba wystąpień wszystkich słów kluczowych w ostatnim paragrafie

Cecha opisująca liczbę słów kluczowych, które występują w ostatnim paragrafie sekcji głównej artykułu.

$$C_{12}(T_{12}, K) = N_{K \in T_{12}}, \tag{33}$$

gdzie T_{12} - ostatni paragraf sekcji głównej artykułu.

3. Opis implementacji

Program został napisany w języku Java z wykorzystaniem narzędzia Maven [2], służącego do automatyzacji budowy oprogramowania.

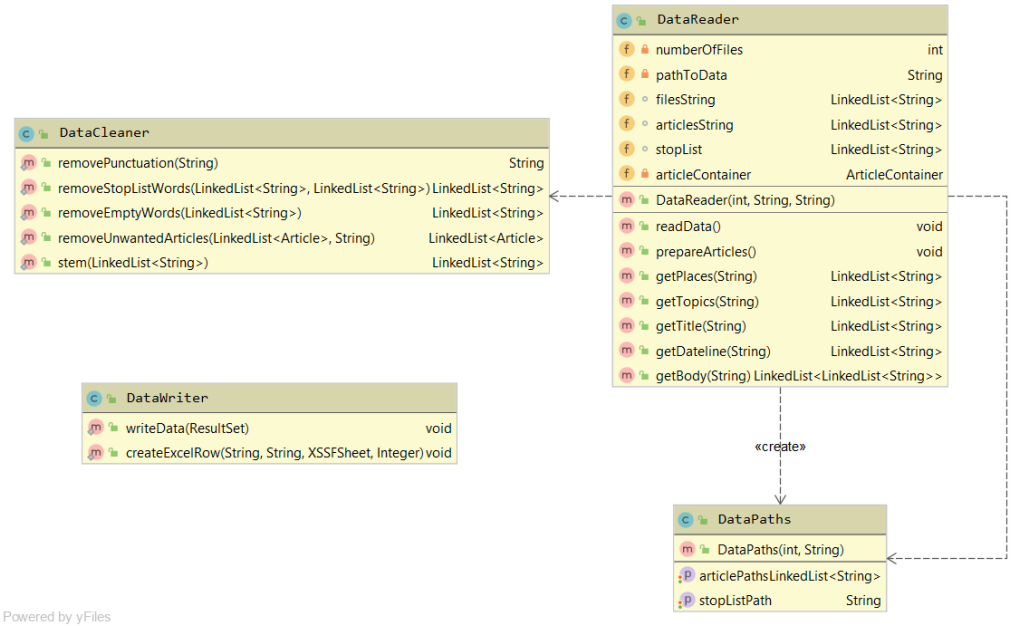
Aplikacja została podzielona na następujące pakiety:

- *Data*,
- *Model* zawierający podpakiety *Testing* i *Training*,
- *Features*,
- *Calculations* zawierający podpakiety *KeyWords*, *Features*, *KNN*, *Metrics* oraz *Measures*,
- *Main*.

W tym rozdziale omówione zostaną wszystkie wyżej wymienione pakiety. Przedstawimy diagramy UML każdego z pakietów a także omówimy zastosowanie poszczególnych klas.

3.1. Pakiet Data

Pakiet *Data* jest odpowiedzialny za wczytywanie danych oraz za przygotowanie struktury obiektów artykułów. Zajmuje się on także eksportowaniem raportu z wyników badań w formacie *xlsx*.



Rysunek 1. Diagram UML dla pakietu *Data*

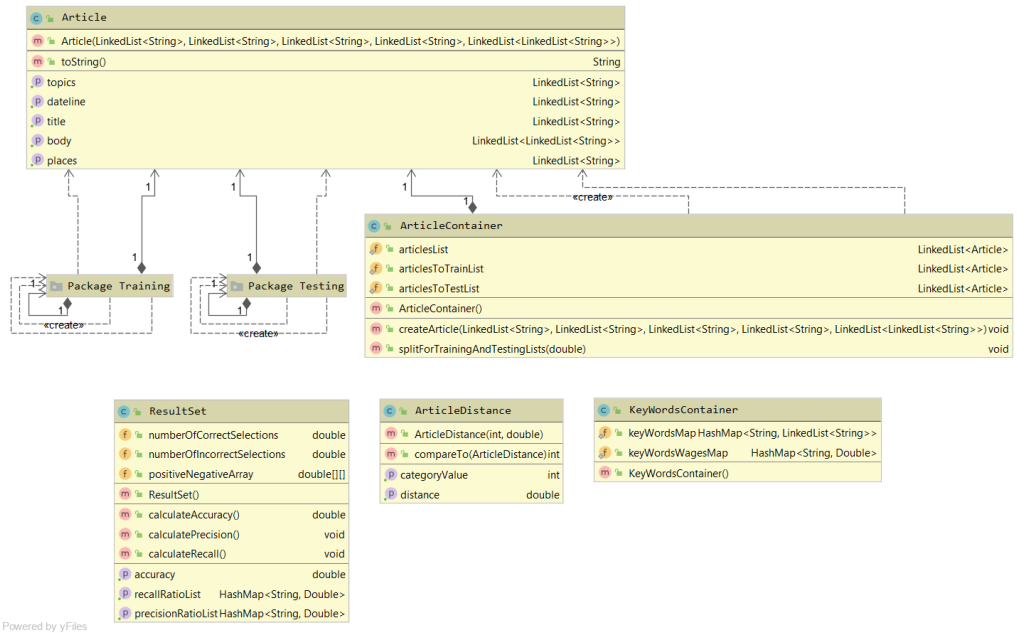
Klasy omawianego pakietu mają następujące zadania:

- *DataReader* - wczytanie danych z plików i zbudowanie struktury obiektów artykułów,
- *DataPaths* - wygenerowanie ścieżek do plików, z których dane będą wczytywane przez obiekt klasy *DataReader*,

- *DataCleaner* - usuwanie słów z artykułów należących do tzw. "Stop" listy, usuwanie znaków interpunkcyjnych, pustych słów, artykułów nienależących do aktualnego zbioru badawczego, a także przeprowadzanie stemizacji tekstu z użyciem biblioteki *snowball – stemmer* [3].
- *DataWriter* - eksportowanie wyników działania programu do raportu w formacie *xslx* z wykorzystaniem biblioteki *poi – ooxml* [4].

3.2. Pakiet Model

Pakiet *Model* zawiera strukturę klas modelowych i kontenerowych, opisuje klasy artykułów, słów kluczowych oraz zestawu rezultatów wykonanych badań. Wykonuje również niektóre proste operacje logiczne takie jak dzielenie artykułów na zbiory testowy i treningowy czy obliczanie parametrów określających jakość klasyfikacji. Zawiera dwa podpakiety - *Testing* oraz *Training*, które opisują strukturę klas artykułów odpowiednio testowych i treningowych.



Rysunek 2. Diagram UML dla pakietu *Model*

Klasy pakietu *Model* są następujące:

- *Article* - klasa modelowa artykułu, przetrzymuje wszystkie listy słów poszczególnych sekcji artykułu,
- *ArticleContainer* - klasa kontenerowa przechowująca wszystkie artykuły, dodatkową funkcjonalnością tej klasy jest dzielenie artykułów na testowe i treningowe (w stosunku określonym w klasie *Settings* w pakiecie *Main*),
- *ArticleDistance* - klasa służąca do porównywania odległości pomiędzy artykułami, implmentuje interfejs Comparable,

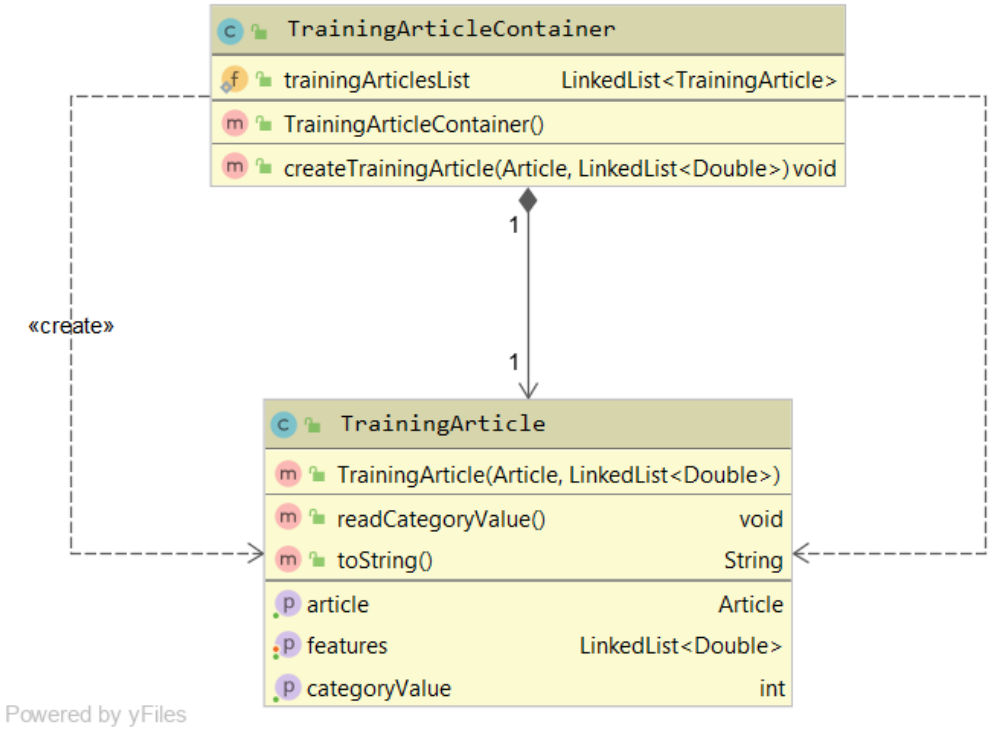
- *KeyWordsContainer* - klasa kontenerowa przechowująca wszystkie słowa kluczowe, w zależności od danej iteracji programu mogą to być zwykłe lub ważone słowa kluczowe,
- *ResultSet* - klasa przetrzymująca wyniki przeprowadzonych badań.

3.2.1. Podpakiet Training

Podpakiet *Training* zawiera klasy opisujące artykuły treningowe. Obiekt klasy *TrainingArticle* zostaje utworzony po tym, jak przeprowadzona zostanie pełna ekstrakcja cech dla odpowiadającego mu obiektu klasy *Article*.

Podpakiet *Training* składa się z dwóch klas:

- *TrainingArticle* - klasa artykułu treningowego zawierająca obiekt klasy *Article* wraz z wektorem wyekstrahowanych cech,
- *TrainingArticleContainer* - klasa kontenerowa przechowująca wszystkie artykuły treningowe (obiekty klasy *TrainingArticle*).



Rysunek 3. Diagram UML dla podpakietu *Training* pakietu *Model*

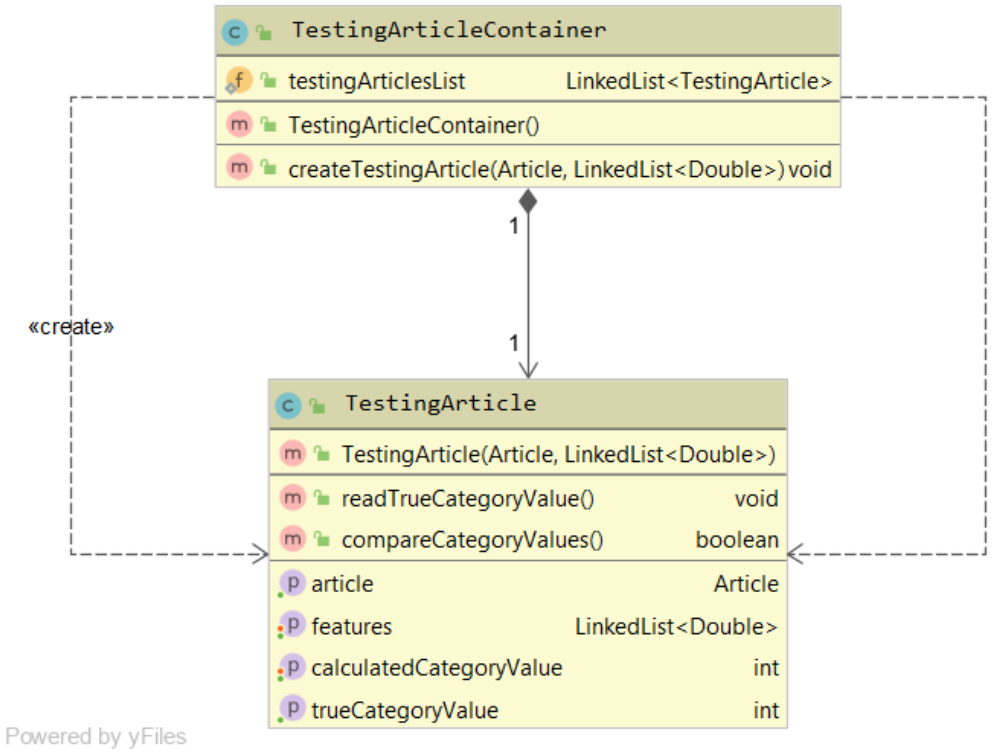
3.2.2. Podpakiet Testing

Podpakiet *Testing* zawiera klasy opisujące artykuły testowe. Podobnie jak w przypadku klasy *TrainingArticle*, obiekt klasy *TestingArticle* zostaje utworzony po tym, jak przeprowadzona zostanie pełna ekstrakcja cech dla odpowiadającego mu obiektu klasy *Article*.

Podpakiet *Testing* składa się z dwóch klas:

- *TestingArticle* - klasa artykułu testowego zawierająca obiekt klasy *Article* wraz z wektorem wyekstrahowanych cech, różni się od klasy *trainingArticle* tym, że oprócz prawdziwej wartości kategorii (tej odczytanej z pliku) posiada rownież wartość obliczoną przez algorytm KNN. Klasa *TestingArticle* implementuje metodę *compareCategoryValues*(), która zwraca *true* w przypadku gdy obliczona przez algorytm KNN wartość kategorii jest identyczna jak wartość prawdziwa, lub $false$ w przeciwnym przypadku,

- *TestingArticleContainer* - klasa kontenerowa przechowująca wszystkie artykuły testowe (obiekty klasy *TestingArticle*).



Rysunek 4. Diagram UML dla podpakietu *Testing* pakietu *Model*

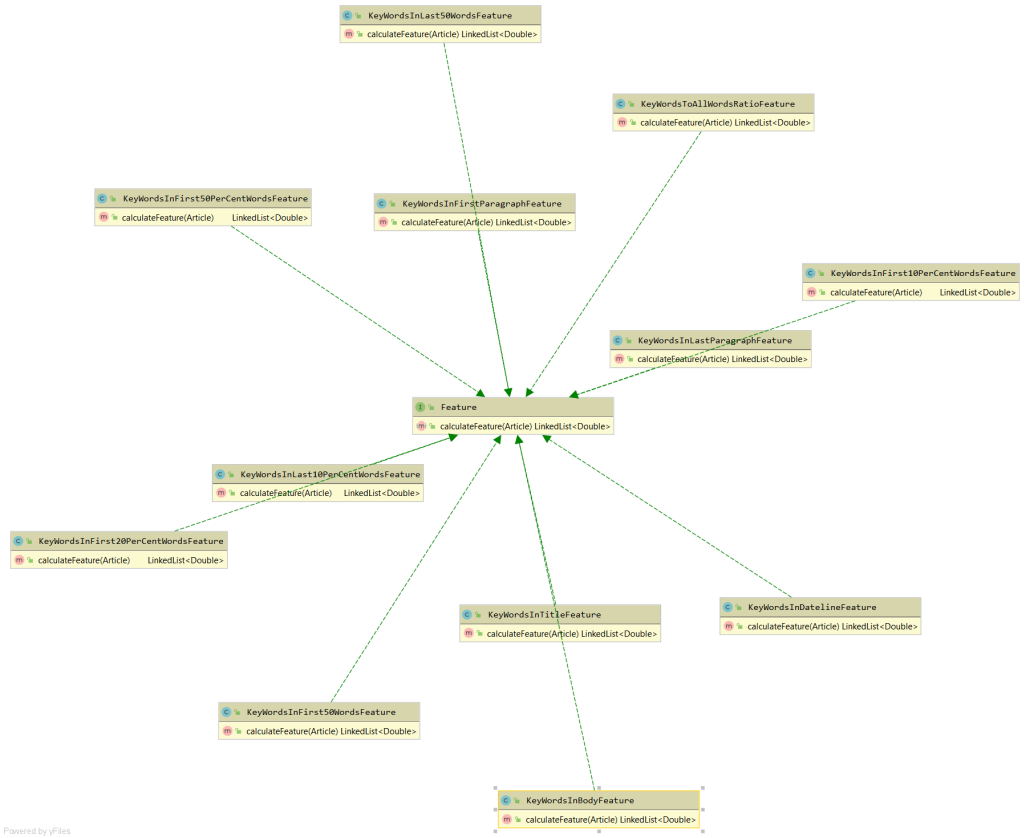
3.3. Pakiet Features

Pakiet *Features* zawiera klasy służące do ekstrahowania cech. Wszystkie klasy pakietu *Features* implementują interfejs *Feature*.

Klasy pakietu *Features* są następujące:

- *Feature* - interfejs implementowany przez wszystkie cechy,
- $KeyWordsInBodyFeature$ - klasa ekstrahująca cechę C_1.
- $KeyWordsInTitleFeature$ - klasa ekstrahująca cechę C_2.

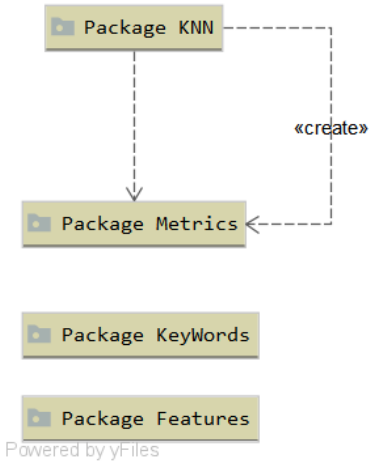
- $KeyWordsInDatelineFeature$ - klasa ekstrahująca cechę C_3.
- $KeyWordsToAllWordsRatioFeature$ - klasa ekstrahująca cechę C_4.
- $KeyWordsInFirst50WordsFeature$ - klasa ekstrahująca cechę C_5.
- $KeyWordsInFirst10PerCentWordsFeature$ - klasa ekstrahująca cechę C_6.
- $KeyWordsInFirst20PerCentWordsFeature$ - klasa ekstrahująca cechę C_7.
- $KeyWordsInFirst50PerCentWordsFeature$ - klasa ekstrahująca cechę C_8.
- $KeyWordsInFirstParagraphFeature$ - klasa ekstrahująca cechę C_9.
- $KeyWordsInLast50WordsFeature$ - klasa ekstrahująca cechę C_{10}.
- $KeyWordsInLast10PerCentWordsFeature$ - klasa ekstrahująca cechę C_{11}.
- $KeyWordsInLastParagraphFeature$ - klasa ekstrahująca cechę C_{12}.



Rysunek 5. Diagram UML dla pakietu *Features*

3.4. Pakiet Calculations

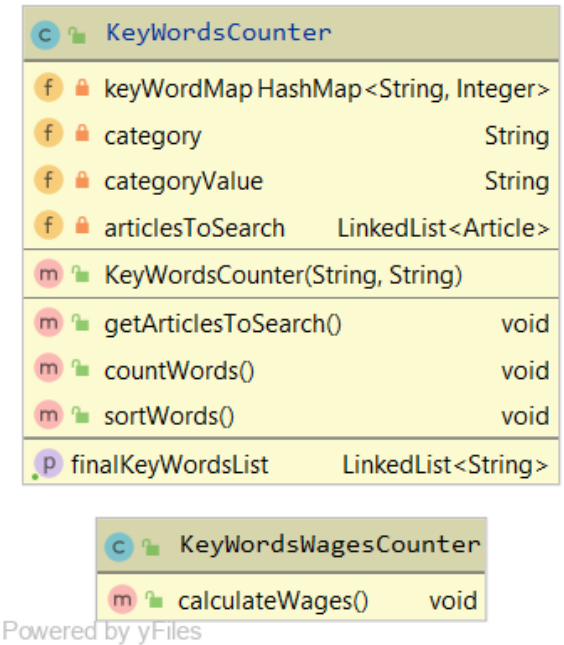
Pakiet *Calculations* jest odpowiedzialny za wszelkiego rodzaju obliczenia. Zawiera on cztery podpakiety - $KeyWords$, $Features$, KNN oraz $Metrics$.



Rysunek 6. Diagram podpakietów pakietu *Calculations*

3.4.1. Podpakiet KeyWords

Podpakiet *KeyWords* odpowiada za wyznaczenie słów kluczowych. Klasy omawianego podpakietu implementują algorytm wyznaczania zwykłych jak i ważonych słów kluczowych, który został opisany we wcześniejszych rozdziałach.



Rysunek 7. Diagram UML dla podpakietu *KeyWords* pakietu *Calculations*

Podpakiet *KeyWords* zawiera dwie klasy:

- *KeyWordsCounter* - klasa implementująca algorytm wyznaczania słów kluczowych,

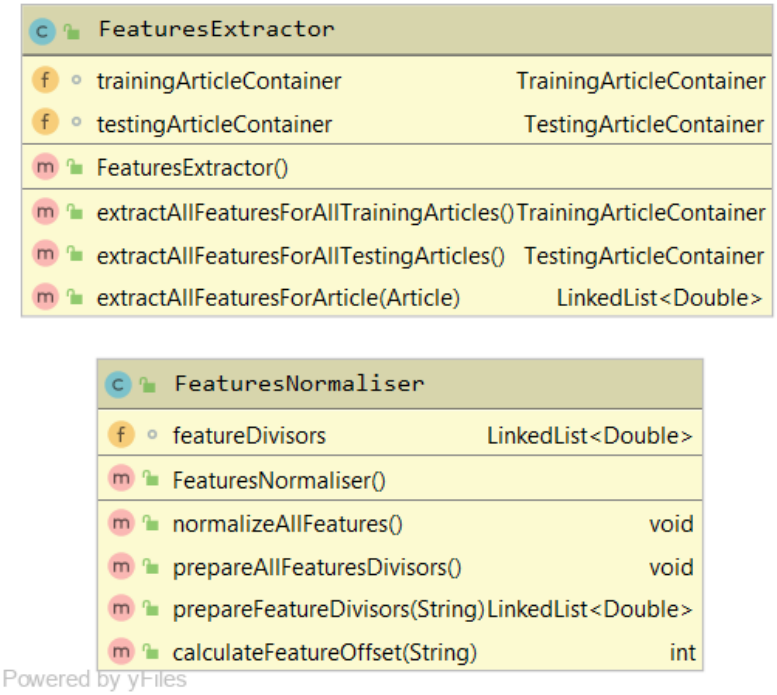
- *KeyWordsWagesCounter* - klasa służąca do obliczania wag słów kluczowych.

3.4.2. Podpakiet Features

Podpakiet *Features* pakietu *Calculations* ma dwa główne zadania - przeprowadzenie ekstrakcji cech dla wszystkich artykułów, a następnie normalizację otrzymanych wektorów cech.

Podpakiet *Features* składa się z dwóch klas:

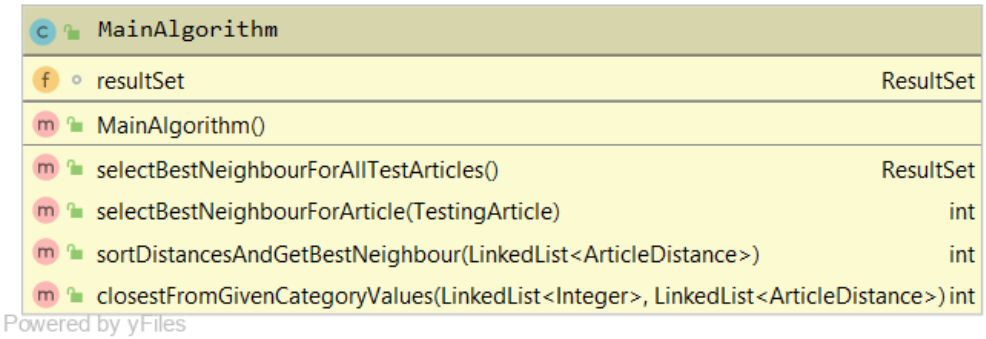
- *FeaturesExtractor* - klasa odpowiedzialna za przeprowadzenie procesu ekstrakcji cech kolejno dla wszystkich wczytanych artykułów,
- *FeaturesNormaliser* - klasa służąca do normalizacji otrzymanych wektorów cech.



Rysunek 8. Diagram UML dla podpakietu *Features* pakietu *Calculations*

3.4.3. Podpakiet KNN

Podpakiet *KNN* implementuje omówiony we wprowadzeniu algorytm k najbliższych sąsiadów. Najlepsza kategoria wyznaczana jest dla wszystkich artykułów testowych.

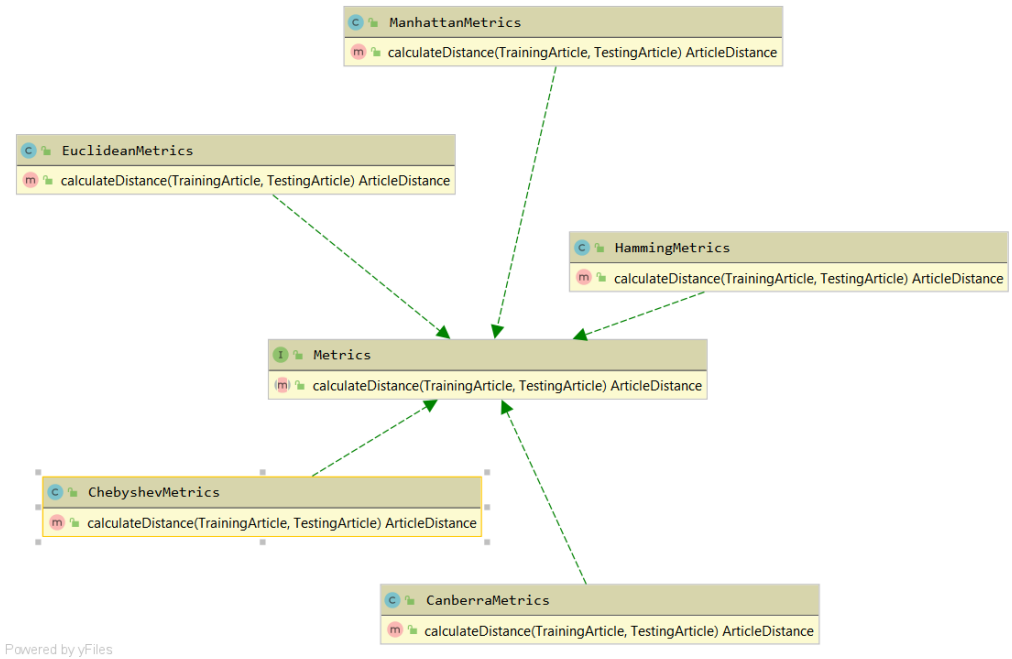


Rysunek 9. Diagram UML dla podpakietu *KNN* pakietu *Calculations*

Podpakiet *KNN* składa się z jednej klasy - *MainAlgorithm*, odpowiedzialnej za przeprowadzenie klasyfikacji dla wszystkich artykułów testowych.

3.4.4. Podpakiet Metrics

Podpakiet *Metrics* zawiera implementacje wszystkich metryk omówionych we wprowadzeniu. Każda z pięciu metryk implementuje interfejs *Metrics*.



Rysunek 10. Diagram UML dla podpakietu *Metrics* pakietu *Calculations*

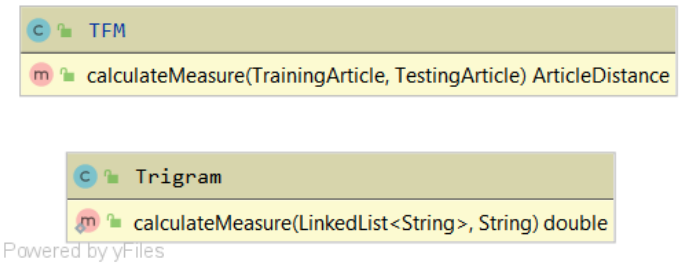
Podpakiet *Metrics* zawiera następujące klasy:

- *Metrics* - interfejs implementowany przez wszystkie metryki,
- *EuclideanMetrics* - klasa implementująca metrykę Euklidesową,
- *ManhattanMetrics* - klasa implementująca metrykę uliczną,

- *ChebyshevMetrics* - klasa implementująca metrykę Czebyszewa,
- *HammingMetrics* - klasa implementująca metrykę Hamminga,
- *CanberraMetrics* - klasa implementująca metrykę Canberra.

3.4.5. Podpakiet Measures

Podpakiet *Measures* zawiera implementacje obu miar omówionych we wprowadzeniu.



Rysunek 11. Diagram UML dla podpakietu *Measures* pakietu *Calculations*

Podpakiet *Measures* zawiera dwie klasy:

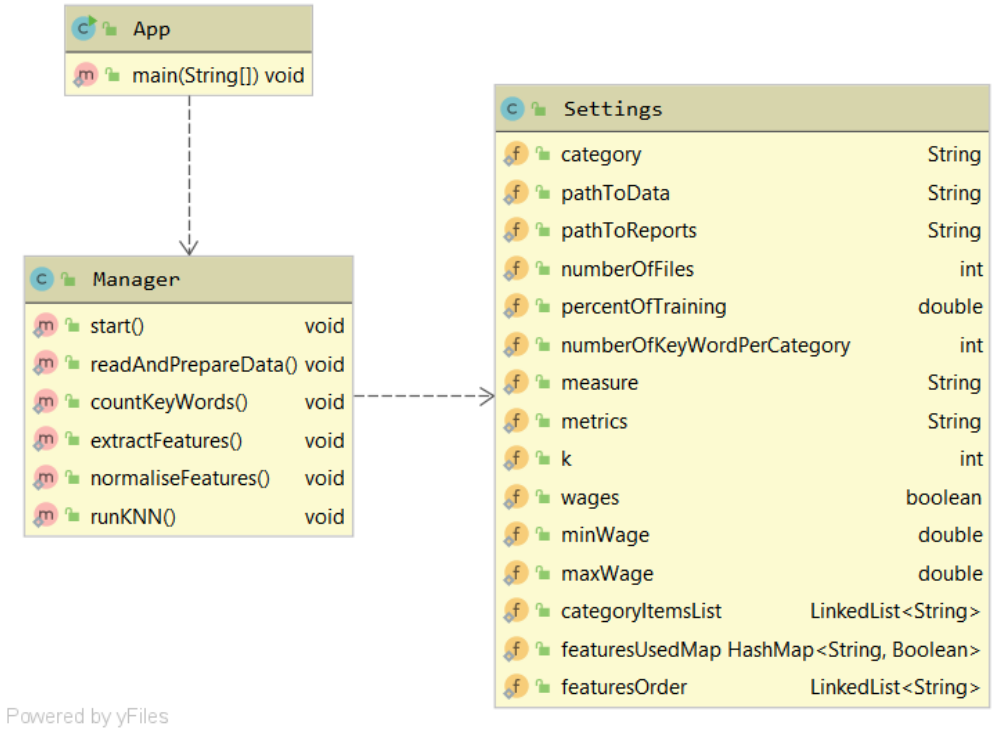
- *TFM* - klasa implementująca miarę Term Frequency Matrix,
- *Trigram* - klasa implementująca miarę Trigram.

3.5. Pakiet Main

Pakiet *Main* jest pakietem głównym, odpowiedzialnym za uruchomienie aplikacji, koordynowanie jej działania oraz określenie konfiguracji w jakiej aplikacja ma działać.

Pakiet *Main* zawiera trzy klasy:

- *App* - klasa uruchamiająca program, wywołuje metodę *start*() w klasie *Manager*,
- *Manager* - klasa sterująca kolejnością wywołania kolejnych modułów aplikacji,
- *Settings* - klasa ustawień, pozwalająca na ustawienie takich parametrów konfiguracyjnych jak metryka, wartość k czy podział artykułów na treningowe i testowe.



Rysunek 12. Diagram UML dla pakietu *Main*

4. Materiały i metody

W tym rozdziale omówione zostaną poszczególne eksperymenty jakie wykonano z użyciem naszego programu.

Klasyfikacje artykułów przeprowadzano ze względu na dwa różne rodzaje etykiet. Pierwszym z nich była lokalizacja (place). Kategorie (etykiety) jakie wyróżniliśmy były następujące: west-germany, usa, france, uk, canada, japan. Klasyfikacja przeprowadzana była jedynie z wykorzystaniem artykułów, których pole "places" przyjmowało jedną z powyższych wartości.

Drugim rodzajem etykiet był temat (topic). Kategorie (etykiety) jakie wyróżniliśmy były następujące: earn, trade, money-supply, acq. Podobnie jak w pierwszym przypadku, klasyfikacja przeprowadzana była jedynie z wykorzystaniem artykułów, których pole "topics" przyjmowało jedną z powyższych wartości.

4.1. Wpływ liczby k sąsiadów oraz wyboru metryki na klasyfikację

Klasyfikacja tekstów została wykonana z wykorzystaniem zbioru (zwykłych) słów kluczowych. Eksperymenty wykonano z użyciem wszystkich pię-

ciu metryk. Dla każdego przypadku testowego dokonano klasyfikacji tekstu dla następujących wartości współczynnika k:

$$k \in \{1, 3, 4, 6, 8, 10, 12, 14, 17, 20\}. \tag{34}$$

W każdym przypadku testowym zbiór treningowy stanowił 70% artykułów, zaś zbiór testowy 30% artykułów.

4.2. Wpływ liczby k sąsiadów oraz wyboru miary na klasyfikację

Podobnie jak w pierwszym przypadku, klasyfikacja została wykonana z wykorzystaniem zbioru (zwykłych) słów kluczowych. Badania przeprowadzono dla obu miar opisanych we wprowadzeniu. Dokonano klasyfikacji tekstu dla wartości współczynnika k wymienionych w poprzednim podrozdziale (34). We wszystkich przypadkach testowych zbiór treningowy stanowił 70% artykułów, zaś zbiór testowy 30% artykułów.

4.3. Wpływ podziału tekstów na zbiory treningowe i testowe na klasyfikację

Klasyfikacja tekstów została wykonana z wykorzystaniem zbioru (zwykłych) słów kluczowych. Eksperymenty przeprowadzono posługując się metryką Euklidesową. Wartość parametru k była stała i wynosiła $k = 6$. Przeprowadzono klasyfikacje dla pięciu różnych podziałów artykułów na zbiory testowe i treningowe:

- Zbiór treningowy: 40% artykułów, zbiór testowy 60%,
- Zbiór treningowy: 50% artykułów, zbiór testowy 50%,
- Zbiór treningowy: 60% artykułów, zbiór testowy 40%,
- Zbiór treningowy: 70% artykułów, zbiór testowy 60%,
- Zbiór treningowy: 80% artykułów, zbiór testowy 20%.

4.4. Wpływ konkretnych cech na klasyfikację

Klasyfikacja tekstów została wykonana z wykorzystaniem zbioru (zwykłych) słów kluczowych. Eksperymenty przeprowadzono posługując się metryką Euklidesową. Wartość parametru k była stała i wynosiła $k = 6$. W każdej iteracji programu zbiór treningowy stanowił 70% artykułów, zaś zbiór testowy 30% artykułów. Przeprowadzono klasyfikacje dla czterech różnych zestawów cech, wybranych spośród wszystkich cech omówionych w rozdziale 2.4. Wybrane zestawy cech były następujące (aby nie duplikować treści, w tym miejscu posługuję się indeksami funkcji cech z rodziału 2.4):

- Zestaw 1: C_1, C_2, C_3, C_4, C_{10}, C_{11}, C_{12},
- Zestaw 2: C_1, C_2, C_3, C_4,
- Zestaw 3: C_5, C_6, C_7, C_8, C_9,
- Zestaw 4: C_2, C_3, C_6, C_{11}.

4.5. Wpływ użycia ważonych słów kluczowych na klasyfikację

Klasyfikacja tekstów została wykonana z wykorzystaniem zbioru zwykłych oraz z użyciem ważonych słów kluczowych. Eksperymenty wykonano z

użyciem wszystkich pięciu metryk. Wartość parametru k była stała i wynosiła $k = 6$. W każdym przypadku testowym zbiór treningowy stanowił 70% artykułów, zaś zbiór testowy 30% artykułów.

5. Wyniki

W tym rozdziale zamieszczono tabele oraz wykresy prezentujące wyniki przeprowadzanych przez nas eksperymentów.

5.1. Wpływ liczby k sąsiadów oraz wyboru metryki na klasyfikację

k	**places** [%]	**topics** [%]
1	81,99	91,35
3	84,99	95,06
4	85,41	94,69
6	85,79	96,23
8	85,14	95,97
10	85,07	94,90
12	85,34	95,49
14	85,19	95,38
17	85,07	95,49
20	85,04	95,70

Tabela 1. Skuteczność klasyfikacji dla metryki Euklidesowej

k	**places** [%]	**topics** [%]
1	81,64	86,25
3	84,92	88,06
4	85,36	87,74
6	85,86	88,91
8	84,35	88,75
10	84,27	88,96
12	84,35	89,28
14	84,35	87,58
17	84,20	87,53
20	84,12	87,05

Tabela 2. Skuteczność klasyfikacji dla metryki Chebysheva

k	places [%]	topics [%]
1	82,15	94,64
3	86,26	96,66
4	86,26	96,76
6	87,00	97,03
8	87,10	96,97
10	86,93	96,92
12	86,78	97,03
14	86,75	96,82
17	86,28	96,82
20	86,08	96,60

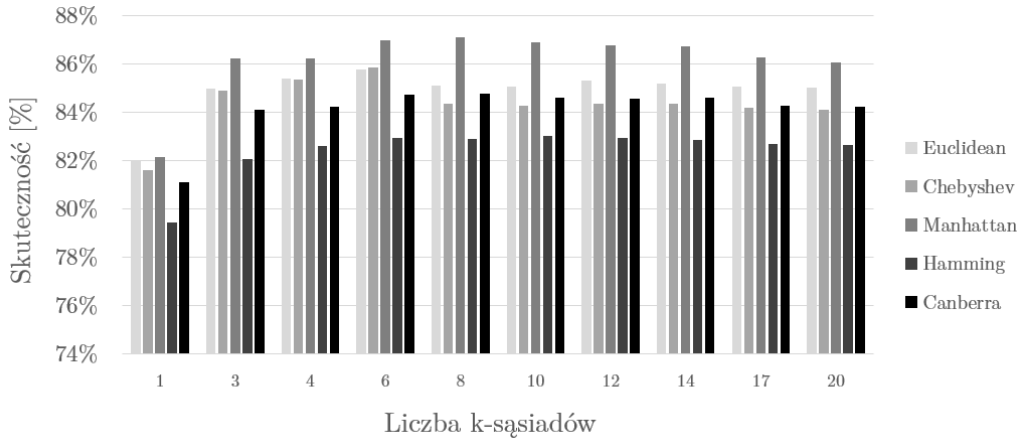
Tabela 3. Skuteczność klasyfikacji dla metryki ulicznej

k	places [%]	topics [%]
1	79,46	92,20
3	82,09	94,11
4	82,63	93,79
6	82,96	93,90
8	82,91	94,32
10	83,03	94,11
12	82,96	94,00
14	82,86	94,00
17	82,71	94,00
20	82,66	94,16

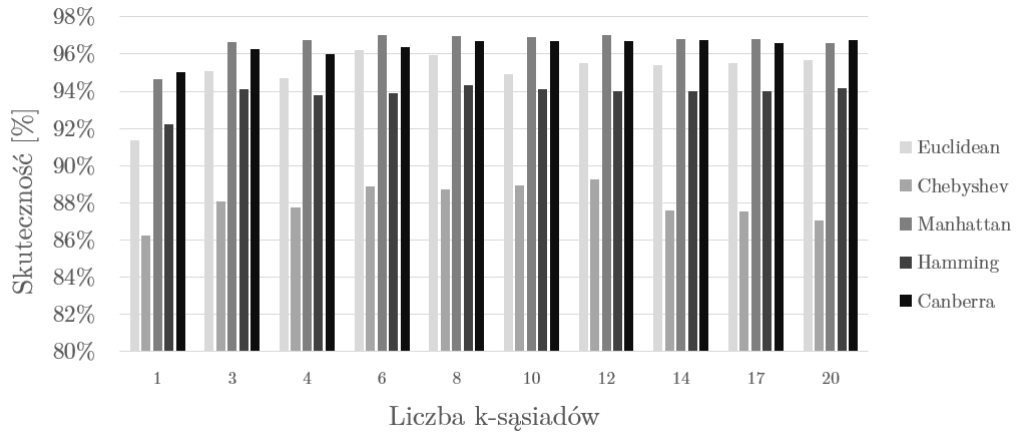
Tabela 4. Skuteczność klasyfikacji dla metryki Hamminga

k	places [%]	topics [%]
1	81,10	95,01
3	84,12	96,28
4	84,25	96,02
6	84,74	96,39
8	84,79	96,71
10	84,62	96,71
12	84,59	96,71
14	84,64	96,76
17	84,30	96,60
20	84,25	96,76

Tabela 5. Skuteczność klasyfikacji dla metryki Canberra



Rysunek 13. Wizualizacja danych z Tabel 1-5 dla kategorii "places"



Rysunek 14. Wizualizacja danych z Tabel 1-5 dla kategorii "topics"

5.2. Wpływ liczby k sąsiadów oraz wyboru miary na klasyfikację

k	places [%]	topics [%]
1	82,29	93,68
3	85,59	94,85
4	85,76	95,06
6	85,83	95,54
8	86,23	95,38
10	86,23	95,33
12	86,06	95,49
14	85,93	95,49
17	85,71	95,44
20	85,44	95,28

Tabela 6. Skuteczność klasyfikacji dla miary TFM

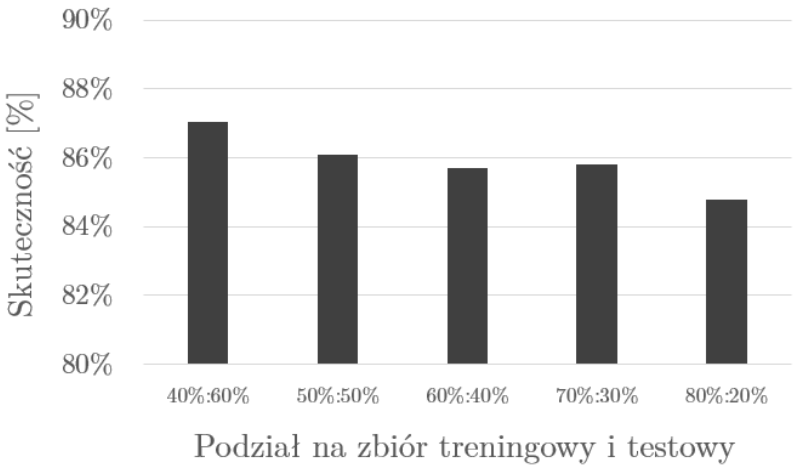
k	places [%]	topics [%]
1	82,98	90,61
3	85,93	94,21
4	85,96	94,48
6	85,96	94,85
8	84,77	94,53
10	85,02	92,83
12	85,14	91,99
14	85,16	92,99
17	85,09	94,27
20	85,02	94,16

Tabela 7. Skuteczność klasyfikacji dla miary trigramów

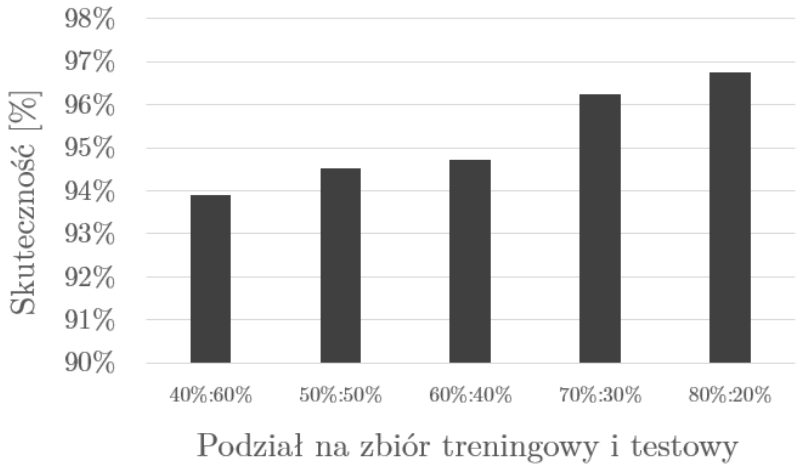
5.3. Wpływ podziału tekstów na zbiory treningowe i testowe na klasyfikację

Podział	places [%]	topics [%]
40:60	87,04	93,89
50:50	86,09	94,52
60:40	85,69	94,71
70:30	85,79	96,23
80:20	84,78	96,74

Tabela 8. Skuteczność klasyfikacji dla różnych podziałów artykułów (podano w kolejności treningowe:testowe)



Rysunek 15. Wizualizacja danych z Tabeli 6 dla kategorii "places"

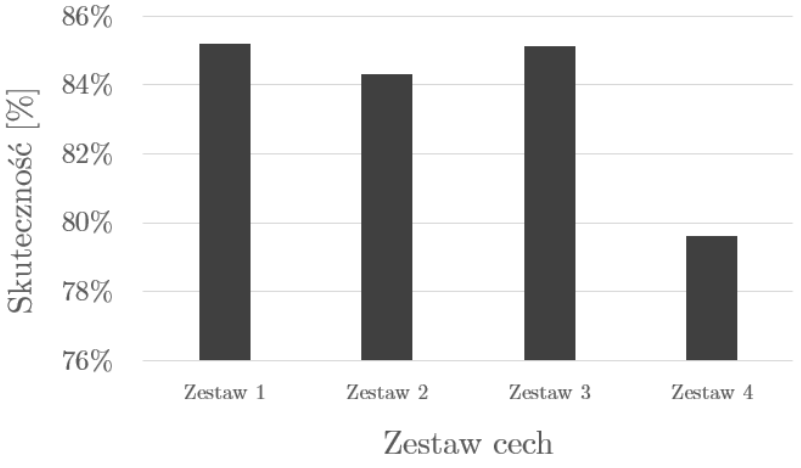


Rysunek 16. Wizualizacja danych z Tabeli 6 dla kategorii "topics"

5.4. Wpływ konkretnych cech na klasyfikację

Zestaw	**places [%]**	**topics [%]**
1	85,19	95,91
2	84,32	95,38
3	85,14	96,18
4	79,61	79,41

Tabela 9. Skuteczność klasyfikacji dla różnych zestawów cech



Rysunek 17. Wizualizacja danych z Tabeli 7 dla kategorii "places"

Rysunek 18. Wizualizacja danych z Tabeli 7 dla kategorii "topics"

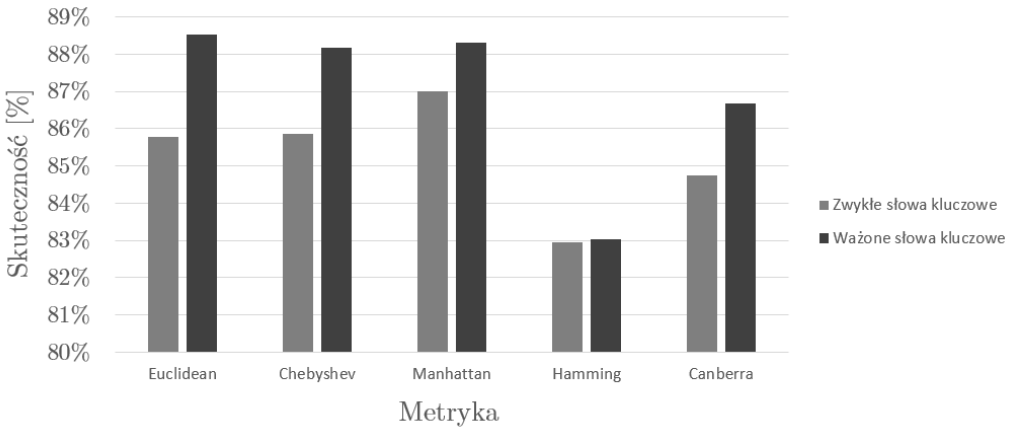
5.5. Wpływ użycia ważonych słów kluczowych na klasyfikację

Metryka	**zwykłe słowa kluczowe [%]**	**ważone słowa klczuowe [%]**
Euclidean	85,79	88,54
Chebyshev	85,86	88,17
Manhattan	87,00	88,32
Hamming	82,96	83,03
Canberra	84,74	86,68

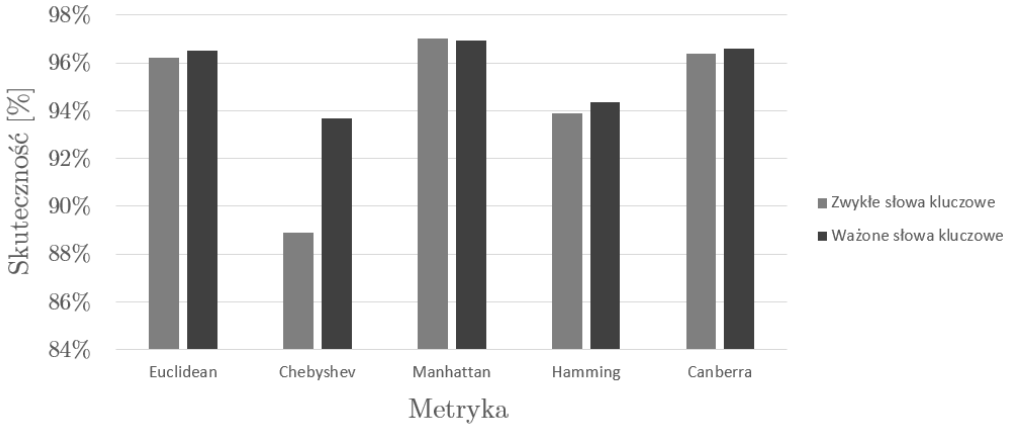
Tabela 10. Skuteczność klasyfikacji dla różnych metod ekstrakcji - zywkłe i ważone słowa kluczowe - kategoria "places"

Metryka	**zwykłe słowa kluczowe [%]**	**ważone słowa kluczowe [%]**
Euclidean	96,23	96,50
Chebyshev	88,91	93,68
Manhattan	97,03	96,92
Hamming	93,90	94,37
Canberra	96,39	96,60

Tabela 11. Skuteczność klasyfikacji dla różnych metod ekstrakcji - zywkłe i ważone słowa kluczowe - kategoria "topics"



Rysunek 19. Wizualizacja danych z Tabeli 8



Rysunek 20. Wizualizacja danych z Tabeli 9

5.6. Najlepsze wyniki

W tabeli poniżej prezentujemy najlepsze wyniki klasyfikacji osiągnięte dla obu rodzajów kategorii.

Kategoria	**k**	**Metryka**	**Słowa kluczowe**	**Skuteczność [%]**
Places	6	Euklidesa	Ważone	88,54
Topics	6	Uliczna	Zwykłe	97,03

Tabela 12. Najlepsze wyniki klasyfikacji dla kategorii "places" i "topics"

6. Dyskusja

W tym rozdziale analizie i dyskusji zostaną poddane wszystkie przedstawione w poprzednim rozdziale wyniki.

6.1. Wpływ liczby k sąsiadów oraz wyboru metryki na klasyfikację

Rozważania rozpoczniemy od analizy wyboru metryki na klasyfikację, następnie przejdziemy do dyskusji liczby k sąsiadów.

6.1.1. Wybór metryki

W przypadku kategorii "places" najlepsze wyniki klasyfikacji osiągane są z wykorzystaniem metryki ulicznej. Niewiele gorsze rezultaty otrzymaliśmy wykorzystując metrykę Euklidesa. Na trzecim miejscu plasują się metryki Czebyszewa i Canberra, przy czym ta pierwsza osiąga lepsze wyniki przy mniejszych wartościach k ($k \leq 6$), ta druga zaś skuteczniejsza jest przy większych k ($k \geq 8$). Zdecydowanie najmniej skuteczną metryką okazała się metryka Hamminga, przy której użyciu skuteczność była mniejsza od konkurencji o 2, a nawet 5%.

Eksperymenty przeprowadzone dla kategorii "topics" ponownie pokazały, jak skuteczna jest klasyfikacja z wykorzystaniem metryki ulicznej. Minimalnie gorsze wyniki skuteczności osiągane były z wykorzystaniem metryk Euklidesowej i Canberra, jednak wszystkie trzy wymienione metryki utrzymywały bardzo wysoki poziom klasyfikacji, na poziomie $95\% - 97\%$. Słabsze rezultaty o mniej więcej 2% osiągane były z użyciem metryki Hamminga, zaś zdecydowanie najmniej skuteczną metryką w przypadku kategorii "topics" okazała się metryka Czebyszewa - klsyfikacja z jej wykorzystaniem ani razu nie osiągnęła skuteczności na poziomie 90% lub wyższej.

6.1.2. Wartość liczby k sąsiadów

W przypadku kategorii "places" najlepsze wyniki klasyfikacji są osiągane dla $k \in \{6, 8, 10, 12, 14\}$. Nieduży spadek jakości klasyfikacji obserwujemy dla większych oraz mniejszych wartości k ($k \in \{17, 20\} \vee k \in \{3, 4\}$), za to bardzo znaczne obniżenie skuteczności eksperymentu obserwujemy dla $k = 1$ - niższa skuteczność w zależności od metryki o 3, do nawet 5%.

Dla kategorii "topics" wybór wartości liczby k nie ma tak dużego wpływu jak w przypadku kategorii "places". Bardzo dobre wyniki klasyfikacji osiągane są dla wszystkich wartości k ze zbioru $k \in \{3, 4, 6, 8, 10, 12, 14, 17, 20\}$, róznice pomiędzy wynikami eksperymentu dla poszczególnych wartości k z tego zbioru są marginalne. Jedynie gdy $k = 1$ wyniki skuteczności są zdecydowanie gorsze, w zależności od metryki różnią się o 1.5 do nawet 4%.

Zdecydowanie słabsze wyniki klasyfikacji dla wartości $k = 1$ w żadnym wypadku nie są zaskakujące. Sytuacja, w której bierzemy pod uwagę tylko jeden artykuł treningowy, który okazał się najbliższym dla badanego elementu testowego, może powodować częste błędy klasyfikacji. Istnieje bowiem duża szansa, że ten jeden odnaleziony przez nas element zbioru treningowego jest mylącym wyjątkiem, którego nie będziemy w stanie skorygować innymi,

okolicznymi artykułami treningowymi, ponieważ w omawianym przypadku w ogóle nie bierzemy takowych pod uwagę.

6.2. Wpływ podziału tekstów na zbiory treningowe i testowe na klasyfikację

Dyskusję wpływu podziału artykułów na zbiory treningowe i testowe rozpoczniemy od analizy przypadków dla kategorii "places".

Klasyfikacja okazała się najbardziej skuteczna dla podziału 40% : 60% (pierwsza wartość stanowi procentowy udział zbioru treningowego w ogólnym zbiorze artykułów, druga zaś udział zbioru testowego). Wyniki gorsze o około 1% osiągnięte zostały dla trzech następnych przedziałów poddanych badaniom - 50% : 50%, 60% : 40% oraz 70% : 30%. Zdecydowanie najgorszy rezultat osiągnięto dla podziału 80% : 20%, gdy skuteczność klasyfikacji wyniosła zaledwie 84.78%.

W przeciwieństawie do wyników analizowanych w poprzednim podrozdziale, które o ile okazały się interesujące i pouczające to w żadnym stopniu nie były nieprzywidywalne - wyniki eksperymentów dla różnych stosunków podziału tekstów dla kategorii "places" mogą okazać się niemałym zaskoczeniem. Na "zdrowy rozsądek" wydawać by się mogło, że im bardziej liczny jest zbiór treningowy, tym lepsze powinny być wyniki klasyfikacji. W większości przypadków to prawda, ale w tym omawianym, okazało się że najbardziej reprezentatywnym zbiorem treningowym był ten, składający się z 40% wszystkich artykułów.

Dla kategorii "topics" wyniki przeprowadzonych badań są już dużo bardziej zgodne z intuicją. Najlepsze wyniki osiągnięto dla podziału 80% : 20%, najgorsze zaś dla podziału 40% : 60%. Rożnica pomiędzy najlepszym a najgorszym wynikiem skuteczności wynosi niecałe 3%.

6.3. Wpływ konkretnych cech na klasyfikację

Dyskusję w tym podrozdziale przeprowadzimy zbiorczo, jako ogólny wpływ wyboru konkretnych cech na jakość klasyfikacji, ze względu na fakt, iż w przypadku obu kategorii, zależności pomiędzy wynikami osiągniętymi przy klasyfikacji z wykorzystaniem poszczególnych zestawów cech są bardzo zbliżone.

W przypadku obu kategorii, rezultaty klasyfikacji z wykorzystaniem zestawów pierwszego i trzeciego są niewiele gorsze od klasyfikacji z wykorzystaniem wszystkich dwunastu cech (opisanych we wprowadzeniu). Użycie zestawu drugiego powoduje niewielki spadek skuteczności w stosunku do zestawów pierwszego i trzeciego. Zdecydowanie najgorsze wyniki osiągane są dla zestawu czwartego - dla kategorii "places" gorsze o około 5% a dla kategorii "topics" nawet o 15% (!).

Z porównania wyników z wykorzystaniem zestawu pierwszego (cechy C_1, C_2, C_3, C_4, C_{10}, C_{11}, C_{12}) oraz zestawu drugiego (cechy C_1, C_2, C_3, C_4) można wywnioskować, że o ile cechy C_1, C_2, C_3, C_4 wydają się wystarczające, to klasyfikacja z wykorzystaniem dodatkowych trzech cech związanych z zawartością końcowych części artykułu (C_{10}, C_{11}, C_{12}) daje lepsze wyniki skuteczności o niecały 1%. Cechy C_{10}, C_{11}, C_{12} nie są zatem niezbędne, aczykolwiek wpływają na poprawę jakości klasyfikacji.

Przedyskutujmy fakty dotyczące wyników badań z użyciem zastawów pierwszego (cechy C_1, C_2, C_3, C_4, C_{10}, C_{11}, C_{12}) i trzeciego (cechy C_5, C_6, C_7, C_8, C_9). Cechy wykorzystane w zestawie trzecim są związane tylko z początkowymi fragmentami artykułu, pomijają zawartość sekcji tytułu i daty, nie angażują w obliczenia również końcowych części tekstu (oczywiście w przypadku krótkich tekstów może się zdarzyć, że analizowana jest zdecydowana większość zawartości artykułu, lub nawet cały artykuł). Skuteczność klasyfikacji dla obu omawianych zestawów jest bardzo zbliżona. Wnioski nasuwają się następujące: po pierwsze, wygląda na to, że cechy C_2 i C_3 związane z zawartością sekcji tytułu i daty są zbędne i nie wpływają znacznie na skuteczność eksperymentu. Po drugie, podobnie skuteczne jest analizowanie całości sekcji body (cechy C_1, i C_4) wraz z większym skupieniem się na jego zakończeniu (cechy C_{10}, C_{11}, C_{12}) jak wykorzystanie do klasyfikacji pięciu cech z zestawu trzeciego opisujących początkowe fragmenty artykułu.

Ostatnim zestawem, którego jeszcze nie poddaliśmy analizie jest zestaw czwarty (cechy C_2, C_3, C_6, C_{11}). Cechy wykorzystane w tym zestawie opisują zwartość sekcji tytułu i daty a także pierwszych i ostatnich 10% słów artykułu. Wyniki klasyfikacji przeprowadzonej z wykorzystaniem zestawu czwartego są zdecydowanie najgorsze. Potwierdza się postulat wysunięty w poprzednim akapicie, dotyczący zbyteczności cech C_2 i C_3. Nowym, przewidywalnym wnioskiem jest fakt, iż poddanie analizie jedynie pierwszych i ostatnich 10% tekstu jest zdecydowanie niewystarczająca do przeprowadzenia skutecznej klasyfikacji.

6.4. Wpływ użycia ważonych słów kluczowych na klasyfikację

W tym podrozdziale pochylimy się nad sensem wprowadzenia naszych autorskich "ważonych słów kluczowych". Wszystkie analizowane wcześniej eksperymenty zostały przeprowadzone z wykorzystaniem zwykłych słów kluczowych. Wszystkie te eksperymenty zostały powtórzone z wykorzystaniem ważonych słów kluczowych, jednak ze względu na przejrzystość badań, nie zdecydowaliśmy się na umieszczenie wszystkich otrzymanych rezultatów, gdyż wymagałoby to podwojenia liczby tabel i wyników zamieszczonych w rozdziale piątym. Przedstawione dane są zdecydowanie wystarczające do wyciągnięcia odpowiednich wniosków.

Dla obu kategorii skuteczność klasyfikacji z wykorzystaniem ważonych słów kluczowych jest wyższa. Największe różnice zanatowano w przypadku kategorii "places" dla metryki Euklidesa i Czebyszewa (między 2 a 3%), a

także w przypadku kategorii "topics" również dla metryki Czebyszewa (prawie 5%). Znaczną poprawę można zauważyć w przypadku metryk ulicznej i Canberra dla kategorii "places" (między 1 a 2%). W pozostałych przypadkach poprawa nie jest aż tak spektakularna, warto jednak zauważyć że jedynym przypadkiem dla którego poprawa nie została zanotowana był eksperyment z wykorzystaniem metryki ulicznej dla kategorii "topics". Mimo wszystko, wynik na poziomie prawie 97% jest więcej niż zadowalający.

Z całą stanowczością należy stwierdzić, że wprowadzenie ważonych słów kluczowych znacznie poprawiło skuteczność klasyfikacji. W przypadku trudnego zadania, jakim jest klasyfikowanie artykułów dla kategorii "places", której zbiór jest skrajnie zdominowany przez elementy z etykietą "USA", udało się osiągnąć skuteczność na poziomie 88.54% co jest wynikiem bardzo dobrym. Dla kategorii "topics" osiągane wyniki są znacznie lepsze, ponieważ oscylują w granicach 96% a dla najlepszych metryk nawet 97%.

7. Wnioski

Poniżej zamieszczono najważniejsze wnioski płynące z przeprowadzonych badań.

- Najbardziej skutecznymi metrykami używanymi do klasyfikacji tekstów są metryki uliczna, Euklidesa i Canberra.
- Mniej skutecznymi metrykami wykorzystywanymi do klasyfikacji tekstów są metryki Czebyszewa i Hamminga.
- Najbardziej optymalnymi wartościami liczby k sąsiadów w algorytmie KNN są wartości k takie, że $k \geq 6 \wedge k \leq 14$.
- Klasyfikacja algorytmem k najbliższych sąsiadów bardzo traci na skuteczności gdy $k = 1$.
- Nie ma jednego "złotego", zawsze odpowiedniego podziału zbiorów na treningowy i testowy.
- Cechy związane z zawartością sekcji tytułu i daty są zbyteczne i nie wpływają znacząco na poprawę jakości klasyfikacji.
- Najważniejszymi cechami są te związane z ogólną liczbą słów kluczowych w tekście, jej stosunkiem do wszystkich słów, a także cechy związane z początkowymi fragmentami tekstu. Mniej ważne (ale nie bezużyteczne) są cechy związane z fragmentami końcowymi.
- "Ważone słowa kluczowe" znacząco porawiają jakość klasyfikacji, opracowany przez nas zestaw par, opisany we wprowadzeniu, okazał się być bardzo skutecznym rozwiązaniem.
- Dla kategorii, w których zbiór elementów treningowych jest bardziej zróżnicowany i zrównoważony, osiągane są dużo lepsze wyniki klasyfikacji niż w kategoriach, których zbiór tekstów jest zdominowany przez elementy z jedną etykietą.

Literatura

[1] A. Niewiadomski *Materiały, przykłady i ćwiczenia do przedmiotu Komputerowe Systemy Rozpoznawania*. 19 czerwca 2012.

[2] Narzędzie Maven
https://maven.apache.org/.

[3] Bibliotek snowball-stemmer
https://mvnrepository.com/artifact/com.github.rholder/snowball-stemmer.

[4] Biblioteka poi-ooxml
https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml.