Informatyka, studia dzienne, I st. VI
Komputerowe systemy rozpoznawania **2019/2020**
Prowadzący: dr hab. inż. Adam Niewiadomski, profesor uczelni pon, 12:00

Aleksandra Wnuk 216924
Michał Kidawa 216796

Zadanie 1 - Ekstrakcja cech, miary podobieństwa, klasyfikacja

1. Cel

Celem zadania jest stworzenie aplikacji konsolowej z funkcjonalnością klasyfikacji zestawów tekstów wykorzystując algorytm k-NN, a następnie zbadanie, jak poszczególne parametry wejściowe wpływają na jakość klasyfikacji.

2. Wprowadzenie

Przez klasyfikację statystyczną rozumiemy rodzaj algorytmu, który przydziela obserwacje statystyczne do klas, bazując na atrybutach, czyli cechach tych właśnie obserwacji.

2.1. Ekstrakcja cech

Przed przystąpieniem do klasyfikacji konieczne jest przeprowadzenie ekstrakcji cech charakterystycznych dla danego tekstu.
Wybrane przez nas cechy na podstawie których utworzyliśmy wektor cech:

1. Podobieństwo do słów kluczowych
   $c_1 = n$

   Słowa kluczowe zostały wybrane metodą Term frequency - inverse document frequency (tf-idf).
   W przypadku naszego programu wykorzystujemy znormalizowaną postać Term frequency. Dla danego dokumentu d_j Term frequency określa wzór:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

gdzie $n_{i,j}$ - liczba wystąpień terminu t_i w dokumencie d_j, a mianownikiem jest suma wystąpień wszystkich terminów w dokumencie d_j.

Inverse Document Frequency - pozwala stwierdzić, czy dane słowo występuje we wszystkich dokumentach.

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|} \tag{2}$$

gdzie $|j : t_i \in d_j|$ - to liczba dokumentów, w których słowo t_i występuje przynajmniej raz, $|D|$ - liczba wszystkich dokumentów.
Wartość tf-idf jest iloczynem wartości Term Frequency i Inverse Document Frequency.

$$(tf - idf)_{ij} = tf_{ij} \times idf_i \tag{3}$$

W przypadku naszego programu, generujemy dla każdego kraju osobną listę słów kluczowych.
Dla każdego słowa kluczowego sprawdzamy jego miarę podobieństwa do każdego słowa w artykule, dla którego obliczany jest wektor cech.

Przykład: Artykuł zawiera 3 słowa (po procesie stemizacji i usunięciu słów ze stop-listy): "kot, ogląd, telewizj, mleko". Lista słów kluczowych to "mysz, krowa, mleko, telewizor". Dla każdego słowa kluczowego liczymy jego podobieństwo do wszystkich słów w artykule uogólnioną miarą n-gramów:

mysz-kot -> 0
mysz-ogląd -> 0
mysz-telewizj -> 0,03
krowa-kot -> 0,13
.
.
.
mleko-mleko -> 1
.
.
.
telewizor-telewizj -> 0,84

Następnie wszystkie wyżej obliczone wartości są sumowane i suma ta jest cechą C1 w wektorze cech dla danego artykułu.
Sposób ten jest ulepszeniem w odniesieniu do bezpośredniego liczenia słów kluczowych, ponieważ w sytuacji, gdy słowo w artykule nie byłoby identyczne ze słowem kluczowym, nie zostałoby w ogóle wzięte pod uwagę. Natomiast miara podobieństwa umożliwia dokładniejsze zbadanie związku artykułu ze słowami kluczowymi.
Na opisanym wyżej przykładzie: cecha "liczba słów kluczowych" nie uwzględniałaby słowa "telewizj", ponieważ nie jest ono identyczne ze słowem kluczowym

"telewizor", a cecha "podobieństwo do słów kluczowych" weźmie już pod uwagę to słowo.

2. Podobieństwo pierwszych 10% tekstu do słów kluczowych
   $c_2 = n_{10\%}$
   Słowa kluczowe wybrane w ten sam sposób, co w punkcie pierwszym. Cecha ta może być przydatna, gdy np. dla tekstów z etykietą "japan" większość słów kluczowych będzie znajdowała się w początkowych 10% tekstu.
   Miary podobieństwa zostały wykorzystane w ten sam sposób, co w punkcie pierwszym, jednak z ograniczeniem do pierwszych 10% tekstu.
3. Znormalizowane podobieństwo do słów kluczowych
   $c_3 = \frac{n}{N}$, gdzie N - liczba wszystkich słów w artykule
   Słowa kluczowe wybierane są w ten sam sposób, co w punkcie pierwszym. Cecha ta różni się pod względem znaczenia dla klasyfikacji pod tym względem, że bierze pod uwagę również długość danego artykułu. Może być przydatna w sytuacji, gdy np. artykuły z daną etykietą będą krótsze niż artykuły z innymi etykietami, ale będą zawierały tyle samo słów kluczowych.
   Miary podobieństwa zostały wykorzystane w ten sam sposób, co w punkcie pierwszym, a następnie suma ich wartości została podzielona przez liczbę wszystkich słów w artykule w celu normalizacji wartości cechy.
4. Średnia długość słowa
   $c_4 = \frac{N_{Zn}}{N}$, gdzie N_{Zn} - liczba znaków (bez znaków przestankowych) w artykule
   Cecha ta może pomóc przy klasyfikacji tekstów w sytuacji, gdy np. teksty z etykietą "usa" będą miały zazwyczaj krótsze słowa niż teksty z innymi etykietami.
5. Częstotliwość występowania słów unikatowych (niebędących słowami kluczowymi)
   $c_5 = \frac{N_U}{N}$, gdzie N_U - liczba wszystkich słów unikatowych w artykule
   Cecha ta wyznacza częstotliwość występowania słów unikatowych, czyli takich, które występują tylko raz w całym dokumencie i nie są słowami kluczowymi.
6. Częstotliwość występowania słów zaczynających się małą literą
   $c_6 = \frac{N_M}{N}$, gdzie N_M - liczba wszystkich słów zaczynających się małą literą
   Cecha ta może pomóc w klasyfikacji tekstów, jeżeli dla jednej z etykiet okaże się, że artykuły do niej przypisane zawierają głównie słowa zaczynające się małą literą, czyli np. artykuły z danego kraju rzadko zawierają nazwy własne, które zaczynałyby się wielkimi literami.
7. Długość artykułu
   $c_7 = N$
   Cecha ta jest wyznaczana jako liczbę słów, które zawiera dany artykuł. Cecha ta może być przydatna w sytuacji, jeśli artykuły z daną etykietą będą wyraźnie dłuższe lub krótsze niż artykuły z innymi etykietami.
8. Częstotliwość występowania krótkich słów
   $c_8 = N_{<4}$
   Cecha ta jest liczona poprzez obliczenie częstotliwości występowania w artykule krótkich słów, czyli takich, które zawierają mniej niż 4 znaki. Cecha ta zwiększy prawidłowość klasyfikacji tekstów, jeżeli część z nich będzie zawierała więcej krótkich słów niż inne teksty.

9. Częstotliwość występowania długich słów
   $c_9 = N_{>7}$
   Cechę tą liczymy poprzez wyznaczenie liczby długich słów w tekście, a następnie podzielenie tej liczby przez liczbę wszystkich słów w danym artykule. Słowa długie zdefiniowaliśmy jako słowa, które zawierają więcej niż 7 znaków. Cecha ta pomoże w klasyfikacji tekstów w sytuacji, w której teksty z etykietą np. "west-germany" będą zawierać więcej wyszukanych lub technicznych słów, które zazwyczaj są dłuższe.

2.2. Algorytm k-NN

Algorytm k-NN (ang. k-nearest neighbours) należy do metod klasyfikacji, w których następuje każdorazowe przetwarzanie danych służacych do nauki, stąd typ algorytmu - "leniwy". Algorytm nie tworzy wewnętrznej reprezentacji danych uczących. Działanie algorytmu opiera się znajdowaniu zgodnie z wybraną metryką odległości między wektorami uczącymi, a danym wektorem ze zbioru testowego. Następnie wybierane jest k najmniejszych odległości oraz sprawdzane są ich etykiety. Ta, której jest najwięcej przypisywana jest do wektora testowego. Jeżeli więcej niż jedna etykieta występuje tyle samo razy, przypadek rozpatrywany jest losowo.

2.3. Metryki

W stworzonej aplikacji wykorzystaliśmy następujące metryki:

— Metryka Czebyszewa

$$d_{ch}(x, y) = \max_{i=1,\ldots,N} |x_i - y_i| \tag{4}$$

— Metrykę euklidesową

$$d_e(x, y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \tag{5}$$

— Metrykę uliczną (Manhattan)

$$d_m(x, y) = \sum_{i=1}^{N} |x_i - y_i| \tag{6}$$

2.4. Miary podobieństwa

W aplikacji zastosowaliśmy dwie miary podobieństwa:

— Uogólnioną miarę n-gramów

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

gdzie h(i; j) = 1 jeśli i-elementowy podciąg w słowie s1 zaczynający się od j-tej pozycji w słowie s1 pojawia się przynajmniej raz w słowie s2 (inaczej h(i; j) = 0);
$N(s_1), N(s_2) =$ – ilość liter w słowach s1 i s2;

$N = \max N(s_1), N(s_2)$;
$\frac{N^2+N}{2}$ – ilość możliwych podciągów od 1-elementowych do N-elementowych w słowie o dlugości N

— Uogólnioną miarę n-gramów z ograniczeniami

$$\mu_N(s_1, s_2) = f(N, n_1, n_2) \sum_{i=1}^{N(s_1)} \sum_{j=1}^{N(s_1)-i+1} h(i, j) \tag{8}$$

gdzie $f(N, n_1, n_2) = \frac{2}{(N-n_1+1)(N-n_1+2)-(N-n_2+1)(N-n_2)}$ wyraża ilość możliwych podciągów o długościach od n_1 do n_2, $1 \leqslant n_1 \leqslant n_2 \leqslant$ N, zaś pozostale symbole – jak powyżej.

Opisane miary zostały wykorzystane w miejcach obliczania wektora cech, konkretnie w cechach związanych ze słowami kluczowymi. Zostało to opisane szczegółowo w podrozdziale 2.1.

2.5. Accuracy

Accuracy, czyli dokładność obliczamy wzorem:

$$ACC = \frac{(TP + TN)}{(P + N)} \tag{9}$$

gdzie $TP+TN$ to liczba poprawnie zaklasyfikowanych artykułów, a $P+N$ to liczba wszystkich artykułów

2.6. Precision

Precyzję obliczamy wzorem:

$$precyzja = \frac{TP}{TP + FP} \tag{10}$$

W przypadku naszego programu dla każdej etykiety liczona jest suma T_P, a wynikiem jest stosunek sum T_P do T_P i F_P. Precyzja jest liczona dla każdej etykiety osobno.

2.7. Recall

Recall, czyli swoistość opisujemy wzorem

$$TNR = \frac{TN}{N} \tag{11}$$

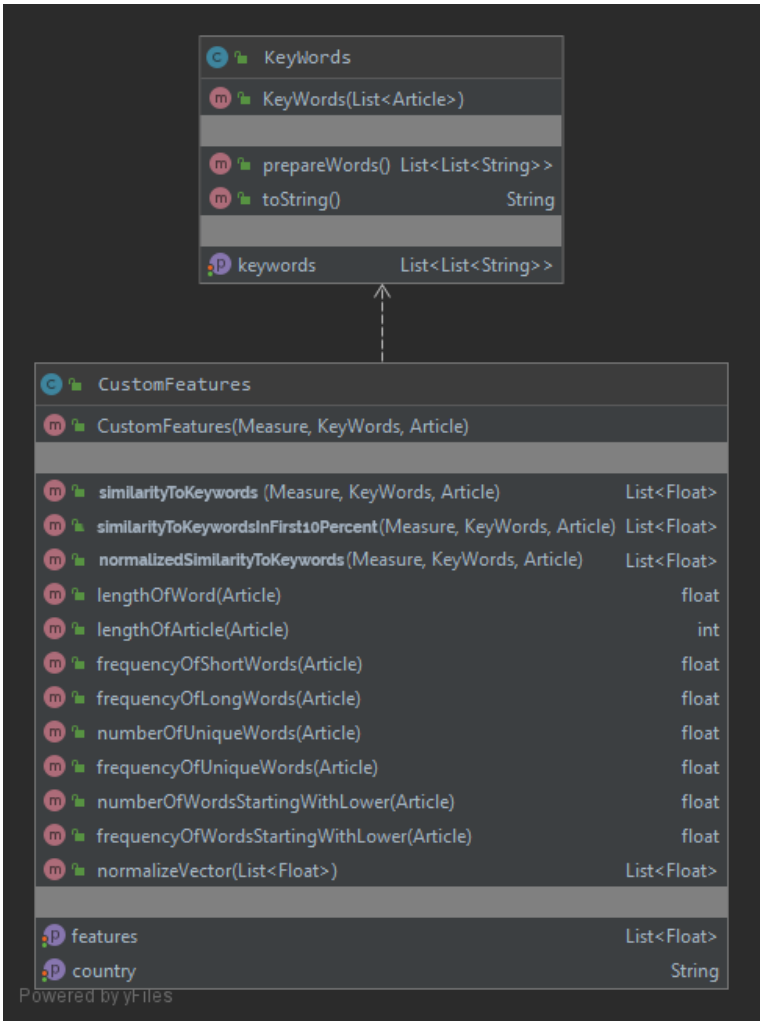
W przypadku naszego programu dla każdej etykiety liczona jest suma T_N, a wynikiem jest stosunek sumy T_N i N. Swoistość jest liczona dla każdej etykiety osobno.

3. Opis implementacji

Aplikacja została zaimplementowana w języku Java, z wykorzystaniem JDK w wersji 1.8 i narzędziem Apache Maven.

3.1. Pakiet extracting

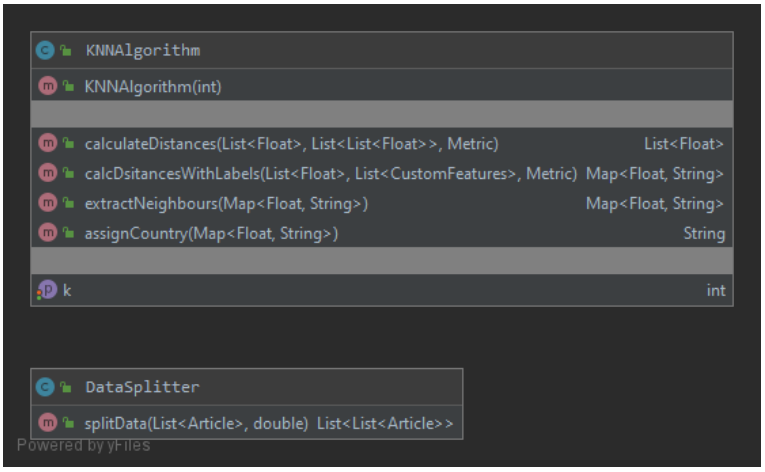
Pakiet ten zawiera klasę CustomFeatures, w ramach której tworzony i zwracany jest wektor cech oraz etykieta. Klasa KeyWords generuje opisanymi wcześniej sposobami słowa kluczowe. Poniżej został przedstawiony diagram UML.



Rysunek 1: Diagram UML pakietu extracting

3.2. Pakiet knn

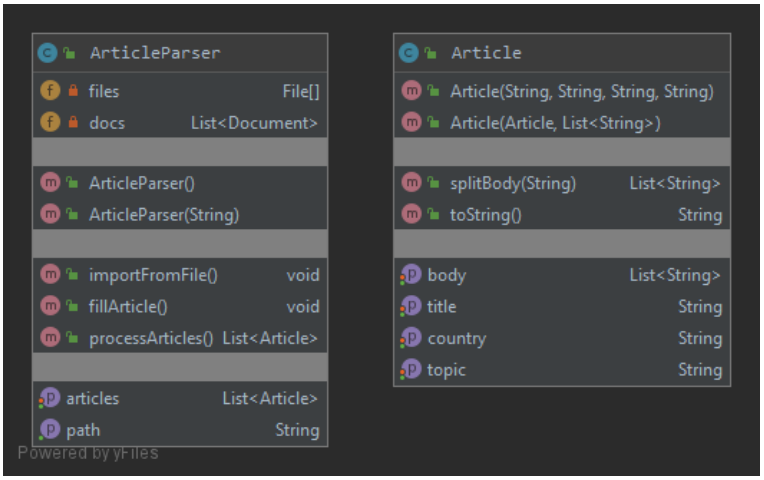
Pakiet zawiera klasę DataSplitter odpowiedzialną za podzielenie danych na zbiór uczący i testowy. Ponadto pakiet ten zawiera klasę KNNAlgorithm, gdzie zaimplementowany został algorytm k-NN. Poniżej przedstawiamy diagram UML pakietu.



Rysunek 2: Diagram UML pakietu knn

3.3. Pakiet loading

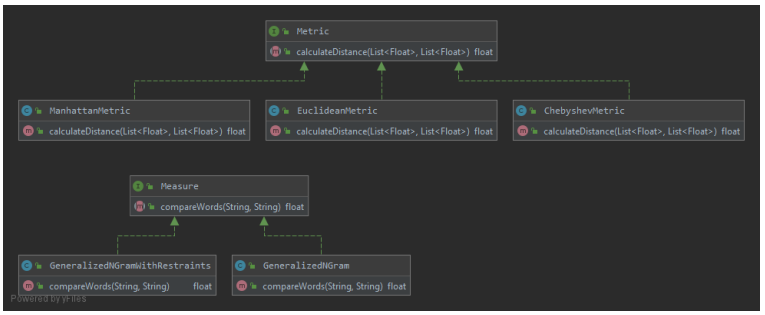
W pakiecie loading można znaleźć klasy odpowiadające za przedstawienie artykułu w postaci obiektu oraz pozwalające przetworzyć pliki wejściowe w postaci dokumentów znacznikowych na tekst. Poniżej przedstawiamy diagram UML pakietu.



Rysunek 3: Diagram UML pakietu loading

3.4. Pakiet metrics

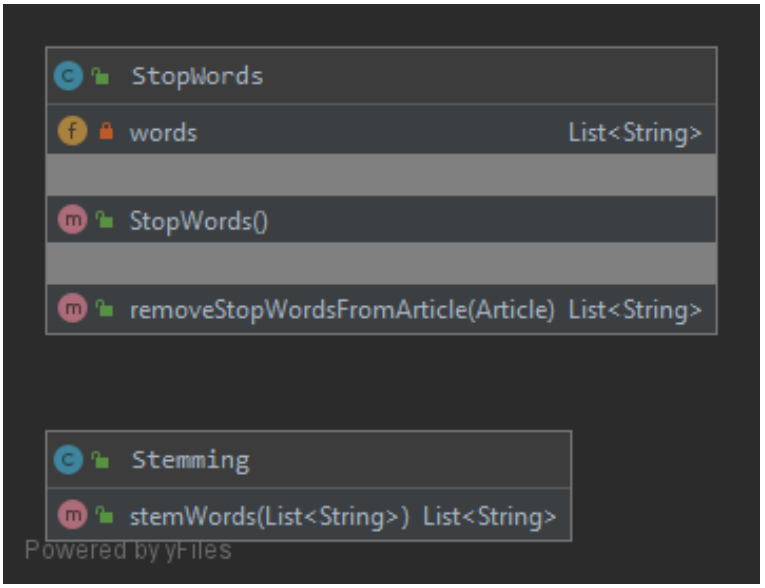
Pakiet ten zawiera zaimplementowane klasy reprezentujące metryki i miary podobienstwa. Poniżej przedstawiamy diagram UML pakietu.



Rysunek 4: Diagram UML pakietu metrics

3.5. Pakiet processing

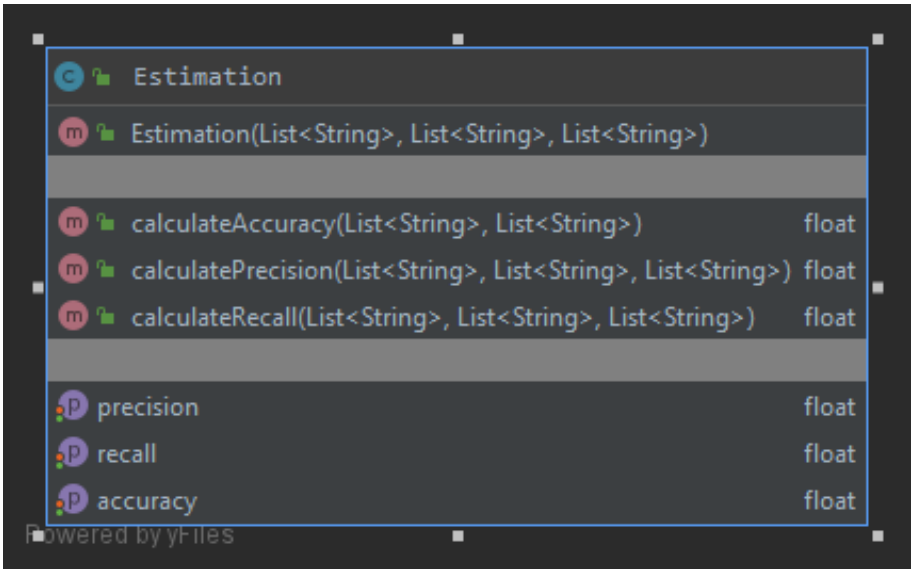
W tym pakiecie zdefiniowane są metody do stemizacji i stoplisty, które są wykorzystywane w pakiecie loading.



Rysunek 5: Diagram UML pakietu processing

3.6. Klasa Estimation

Zawarte w klasie Estimation metody wykorzystywane są do analizy jakości otrzymanych wyników. Poniżej diagram UML klasy.



Rysunek 6: Diagram UML klasy Estimation

4. Materiały i metody

Dane na których przeprowadzone zostały badania zostały zaczerpnięte ze strony archive.ics, ze zbioru artykułów Reuters z 1987 roku [1]. Wzięliśmy pod uwagę wyłącznie artykuły, które posiadały jedną z etykiet w znaczniku "places" - usa, west-germany, canada, france, uk, japan. Odrzuciliśmy także artykuły, których treść brzmiała "blah blah blah". Następnie artykuły zostały poddane procesowi stemizacji, a ich treść została zredukowana o słowa zawarte w stopliście. Powstałe w wyniku tego procesu artykuły stanowią dane wejściowe. Jego liczność to. Około 80% artykułów stanowią artykuły z etykietą usa. W celu przeprowadzenia analizy wpływu cech na jakość klasyfikacji przeprowadziliśmy szereg testów, w trakcie których zmienialiśmy wartości parametrów wejściowych programu.

Najpierw wyznaczyliśmy zależność Accuracy od k, przy stałych wartościach innych parametrów. Następnie przy wybranej stałej wartości k wyznaczyliśmy zależność Accuracy od pięciu wartości proporcji podziału zbioru, przy stałych wartościach innych parametrów. Podczas wykonywania testów wybraliśmy następnuje stosunki podziałów uczący - testowy (w %): 40 - 60, 50 - 50 , 60 - 40 , 75 - 25, 90 - 10. Następnie wyznaczyliśmy zależność Accuracy od wyboru metryki (przy pozostałych parametrach stałych).

Na koniec zbadaliśmy jakość klasyfikacji zależnie od wybranych cech oraz miar używanych podczas ekstrakcji cech C1, C2 i C3. Na podstawie otrzymanych wyników byliśmy w stanie stwierdzić, które cechy potencjalnie mają najmniejszy, a które największy na wyniki klasyfikacji.

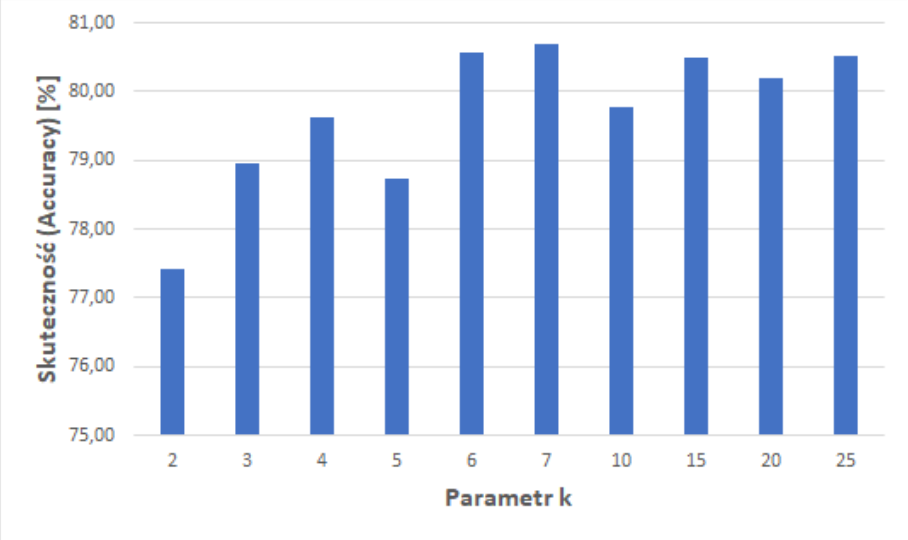
5. Wyniki

5.1. Zależność Accuracy od k

Aby wyznaczyć zależność między wartością Accuracy, a liczbą k-sąsiadów przeprowadziliśmy test dla 10 różnych wartości k, dla podziału zbioru uczący - testowy w stosunku 75 do 25, przy wszystkich cechach i dla metryki euklidesowej.

k	Accuracy [%]
2	77,42
3	78,95
4	79,63
5	78,75
6	80,58
7	81,69
10	79,78
15	80,49
20	80,20
25	80,52

Tabela 1: Skuteczność dla różnych wartości k przy stałej wartości innych parametrów



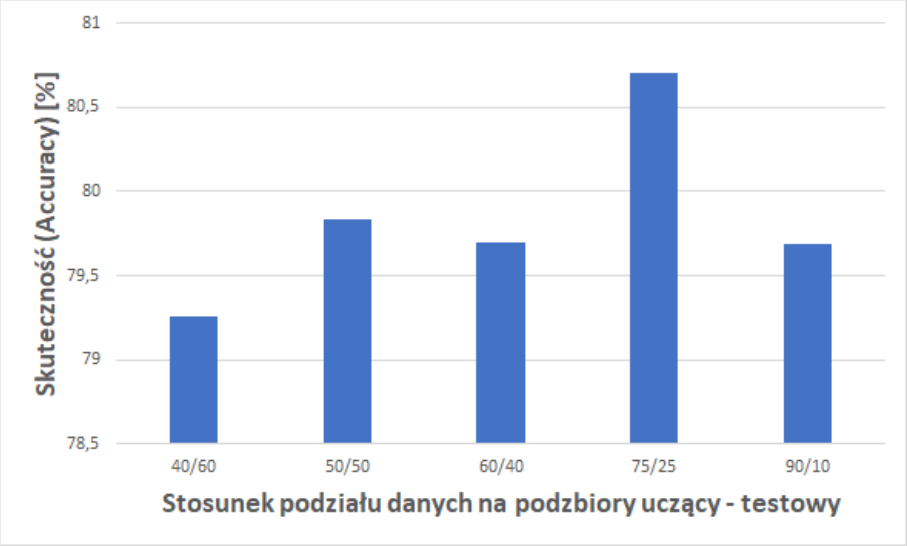
Rysunek 7: Wykres zależnośći skuteczności od parametru k

5.2. Zależność Accuracy od podziału zbioru uczący - testowy

Aby zbadać zależność wartości Accuracy od podziału zbioru na podzbiory uczący i testowy przeprwowadziliśmy 5 testów.

Stosunek podziału	Accuracy [%]
40/60	79,26
50/50	79,83
60/40	79,70
75/25	80,70
90/10	79,69

Tabela 2: Skuteczność dla różnych podziałów, przy stałej wartości innych parametrów



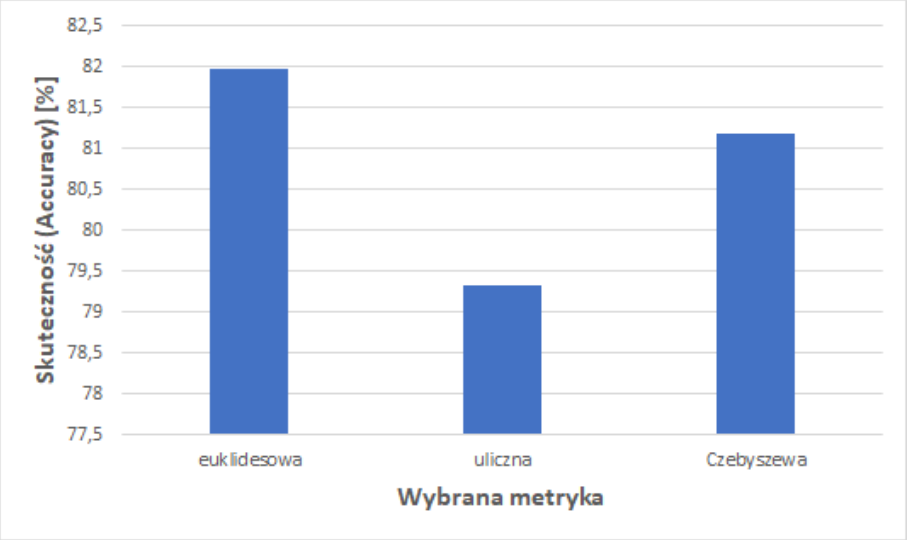
Rysunek 8: Wykres zależności skuteczności od podziału na zbiory uczący - testowy

5.3. Zależność Accuracy od wybranej metryki oraz miar

5.3.1. Metryki

Metryka	Accuracy [%]
euklidesowa	81,98
uliczna	79,32
Czebyszewa	81,17

Tabela 3: Skuteczność dla wykorzystanych metryk, przy stałej wartości innych parametrów

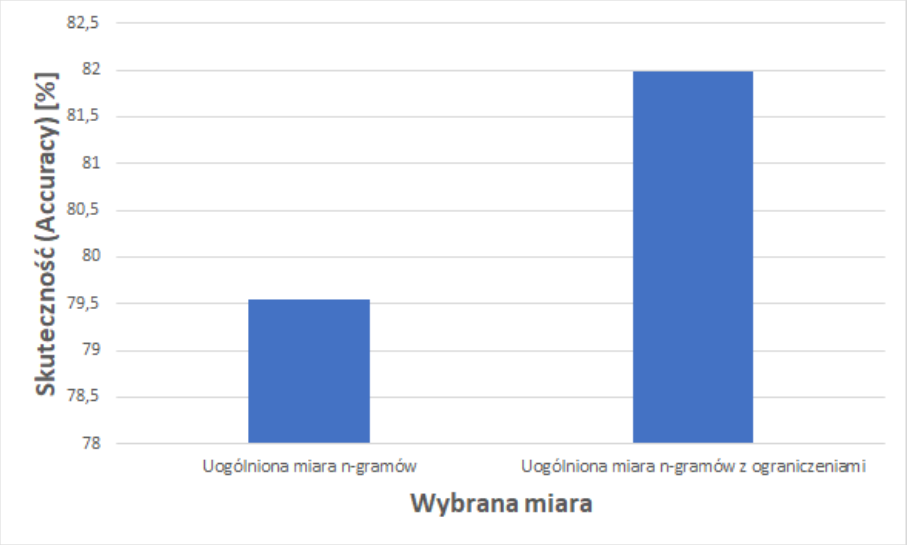


Rysunek 9: Skuteczność wybranych metryk, przy stałej wartości innych parametrów

5.3.2. Miary

Miara	Accuracy [%]
Uogólniona miara n-gramów	79,54
Uogólniona miara n-gramów z ograniczeniami	81,98

Tabela 4: Skuteczność klasyfikacji dla różnych miar podobieństwa użytych w ekstrakcji cech, przy stałej wartości innych parametrów



Rysunek 10: Skuteczność klasyfikacji dla różnych miar podobieństwa użytych w ekstrakcji cech, przy stałej wartości innych parametrów

5.4. Wpływ konkretnych cech na jakość klasyfikacji

Po przeprowadzeniu wielu testów z różnymi konfiguracjami cech wybraliśmy 4 podzbiory:

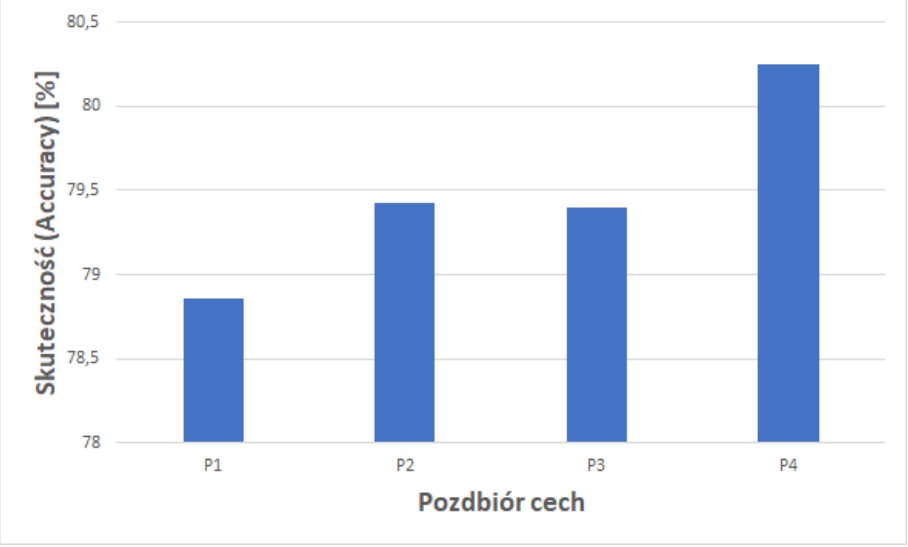
— P_1 - zawierał cechy: C_1, C_2, C_5, C_6, C_7, C_8, C_9
— P_2 - zawierał cechy: C_1, C_2, C_5, C_6, C_7, C_8
— P_3 - zawierał cechy: C_3, C_5, C_8
— P_4 - zawierał cechy: C_3, C_5, C_7

Cechy - legenda:

C_1 - Podobieństwo do słów kluczowych
C_2 - Podobieństwo pierwszych 10% tekstu do słów kluczowych
C_3 - Znormalizowane podobieństwo do słów kluczowych
C_4 - Średnia długość słowa
C_5 - Częstotliwość występowania słów unikatowych (niebędących słowami kluczowymi)
C_6 - Częstotliwość występowania słów zaczynających się małą literą
C_7 - Długość artykułu
C_8 - Częstotliwość występowania krótkich słów
C_9 - Częstotliwość występowania długich słów

Podzbiór cech	Accuracy [%]
P1	78,86
P2	79,43
P3	79,4
P4	80,25

Tabela 5: Skuteczność dla różnych podzbiorów cech, przy stałej wartości innych parametrów



Rysunek 11: Skuteczność wybranych pozdbiorów cech

Podzbiór cech	Accuracy [%]
bez C_1	78,33
bez C_2	78,42
bez C_3	78,09
bez C_4	79,68
bez C_5	79,55
bez C_6	78,45
bez C_7	78,22
bez C_8	80,45
bez C_9	80,12

Tabela 6: Skuteczność dla podzbiorów bez wybranych cech, przy stałej wartości innych parametrów

5.5. Precyzja i swoistość

Wartości precyzji i swoistości zostały obliczone wyłącznie dla parametrów, dla których wartość skuteczności była najwyższa, czyli dla k równego 7, podziału na zbiór uczący i testowy 75% - 25%, metryki euklidesowej, uogólnionej miary n-gramów z ograniczeniami i podzbioru cech P4.

Miara jakości	"usa"	"west-germany"	"france"	"uk"	"canada"	"japan"
Precyzja [%]	81,63	33,33	Nan	45,45	28,57	46,15
Swoistość [%]	98,83	1,32	0,00	8,89	1,98	6,74

Tabela 7: Wartości precyzji i swoistości

Wartość precyzji "NaN" oraz swoistości równej 0% otrzymana dla etykiety "france" oznacza, że żaden artykuł nie został zaklasyfikowany jako posiadający tę etykietę.

6. Dyskusja

Analizując uzyskane wyniki, doszliśmy do wniosku, że dobrany zbiór artykułów jest zbyt mało zróżnicowany pod względem zawartości tekstów zależnie od etykiet, aby móć zaklasyfikować je z większą dokładnością. Jednak mimo niewielkich różnic udało się zauważyć pewne zależności.

6.1. Zależność Accuracy od wartości k

Analizując Tabelę 1 oraz wykres na Rysunku 7 zaobserwowaliśmy, że najgorsze wyniki program osiągał przy wartości k równej 2. Zauważyliśmy również spadek skuteczności przy k równym 5. Program odznaczał się największą skutecznością przy k wynoszącym 7, dlatego też zdecydowaliśmy się na wykorzystanie tej wartości w kolejnym etapie testów.

6.2. Podział na zbiór uczący i testowy

Biorąc pod uwagę wyniki otrzymane w Tabeli 2 oraz zaobserwowane na Rysunku 8 doszliśmy do wniosku, że podziałem dla którego skuteczność jest najwyższa jest podział danych na podzbiory uczący - testowy w stosunku 75% - 25%.

6.3. Wybór metryki

Ze wszystkich testowanych przez nas metryk najgorszą skutecznością odznaczała się metryka uliczna, co można zaobserwować na Rysunku 9 i analizując Tabelę 3. Wyniki dla metryki euklidesowej oraz Czebyszewa nie różniły się znacznie od siebie, z minimalną przewagą dla metryki euklidesowej.

6.4. Wybór miary

Analizując Tabelę 4, zaobserwowaliśmy większą skuteczność w przypadku uogólnionej miary n-gramów z ograniczeniami.

6.5. Wpływ konkretnych cech na jakość klasyfikacji

Podczas przeprowadzania testów zauważyliśmy, zgodnie z Tabelą 7 i wykresem na Rysunku 11, że ponad połowa wybranych przez nas cech ma negatywny wpływ na jakość klasyfikacji. Kluczowe dla uzyskania wyższej wartości Accuracy okazały się cechy związane ze znormalizowanym podobieństwem do słów kluczowych (C3) oraz częstotliwością występowania słów unikatowych (C5). Przeprowadziliśmy również testy, które po kolei nie brały pod uwagę jednej z cech (Tabela 6). Na podstawie wyników możemy wskazać cechy związane ze średnią długością słowa oraz częstotliwością występowania krótkich oraz długich słów, jako te, które mają potencjalnie najmniejszy wpływ na wynik klasyfikacji.

7. Wnioski

— Wartość k ma znaczny wpływ na jakość otrzymanej klasyfikacji.
— Metryka jako parametr ma mały wpływ na jakość otrzymanej klasyfikacji.
— Uogólniona miara n-gramów z ograniczeniami jest dokładniejsza niż uogólniona miara n-gramów.
— Cechy odpowiedzialne za średnią długość słowa oraz częstotliwość występowania wyrazów krótkich, lub długich nie sprawdzają się przy klasyfikacji tesktów.
— Podział tekstów na zbiory testowe oraz uczący ma wpływ na jakość otrzymanej klasyfikacji.

Literatura

[1] https://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection
[2] Methods for the Linguistic Summarization of Data: Applications of Fuzzy Sets and Their Extensions, Adam Niewiadomski, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2008

[3] Alpaydin, Ethem (2010). Introduction to Machine Learning. MIT Press. p. 9. ISBN 978-0-262-01243-0
[4] https://opennlp.apache.org/docs/1.7.2/apidocs/opennlp-tools/opennlp/tools/stemmer/Stemmer.html
[5] http://home.agh.edu.pl/horzyk/lectures/miw/MIW-KNN.pdf
[6] Olson, David & Delen, Dursun. (2008). Advanced Data Mining Techniques. 10.1007/978-3-540-76917-0