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.

Zaimplementowaliśmy 4 metody ekstrackji cech, będziemy je nazywać kolejno:
— Pierwszy sposób ekstrakcji
— Drugi sposób ekstrakcji
— Trzeci sposób ekstrakcji
— Czwarty sposób ekstrakcji

2.1. Pierwszy sposób ekstrakcji

Wektor cech w pierwszym sposobie ekstrakcji jest tworzony poprzez wybieranie słów kluczowych z tekstów. Kolejno dla każdej z etykiet wykonuje się kroki:

1. Z tekstów zawierających daną etykietę wybierane są wszystkie słowa jakie się w nich znajdują i tworzony jest z nich wektor.
2. Z wektora usuwane są słowa znajdujące się na stop liście.
3. Z wektora usuwane są jakiekolwiek liczby.
4. Przy pomocy pomiaru częstotliwości pojawiania się słów bliżej początku tekstu, wybierana jest określona wcześniej liczba słów.

Następnie wszystkie wektory powstałe dla każdej etykiety łączone są w jeden wektor nie zawierający powtórzeń. Wektor cech w pierwszym sposobie ekstrakcji przyjmuje postać:

$$v = [\text{"}word1\text{"}, \text{"}word2\text{"}, ... \text{"}wordN\text{"}]$$

Następnie dla każdego z tekstów znajdującego się w zbiorze testowym ustalany jest wektor poprzez określenie podobieństwa słów z tekstu do słów w wektorze. Następnie następuje normalizacja do przediału wartości od 0 do 1. Powstają wtedy wektory gotowe do umieszczenia w przesrzeni knn przyjmujące na przykład postać:

$$v = [0, 0.4, 0.6, 1, 0.6]$$

2.2. Drugi sposób ekstrakcji

Wektor w drugim sposobie ekstrakcji składa się z 11 cech(11 liczb). Pierwsza z nich określa liczbę zdań w tekście, druga liczbę słów. Następnie 8 kolejnych liczb odpowiada: liczbie słów składających się z 2 i mniej liter, liczbę słów składających się z 3 liter, liczbę słów składających się z 4 liter, liczbę słów składających się z 5 liter, liczbę słów składających się z 6 liter, liczbę słów składających się z 7 liter, liczbę słów składających się z 8 liter oraz liczbę słów składających się z 9 i więcej liter. Ostatnim elementem jest liczba słów zaczynających się z wieliej litery.
Wektor cech w drugim sposobie ekstrakcji przyjmuje postać:

$$v = [1, 2, 3, 5, 12, 3, 3, 4, 10, 5, 0]$$

2.3. Trzeci sposób ekstrakcji

Trzeci sposób jest analogiczny do pierwszego, różni się tylko zamiana słów na reprezentujące je skróty. Skrót taki składa się z pierwszej litery słowa, ostatniej litery słowa i liczby liter w słowie, przykładowo przyjmiuje wartość:

$$s(american) = an8$$

Podobnie jak w pierwszym sposobie ekstrakcji(proszę zwrócić uwagę na dodanie 4 punktu) wykonuje sie kroki:

1. Z tekstów zawierających daną etykietę wybierane są wszystkie słowa jakie się w nich znajdują i tworzony jest z nich wektor.
2. Z wektora usuwane są słowa znajdujące się na stop liście.
3. Z wektora usuwane są jakiekolwiek liczby.
4. Zamiana słów na skróty.
5. Przy pomocy pomiaru częstotliwości pojawiania się słów bliżej początku tekstu, wybierana jest określona wcześniej liczba słów.

Następnie wszystkie wektory powstałe dla każdej etykiety łączone są w jeden wektor nie zawierający powtórzeń.
Wektor cech w trzecim sposobie ekstrakcji przyjmuje postać:

$$v = [\ddot{a}k8", "wn6", ..." fg5"]$$

Następnie dla każdego z tekstów znajdującego się w zbiorze testowym ustalany jest wektor poprzez określenie podobieństwa słów z tekstu do słów w wektorze. Następnie następuje normalizacja do przediału wartości od 0 do 1.

Powstają wtedy wektory gotowe do umieszczenia w przesrzeni knn przyjmujące na przykład postać:

$$v = [0, 0.4, 0.6, 1, 0.6]$$

W trzecim sposobie ekstrakcji zastosowana została własna miara podobieństwa obiczaza poprzez wzór przedstawiony w formie wycinku kodu z klasy *OurComparator*:

```
if(s1.charAt(0)==s2.charAt(0) && s1.charAt(1)==s2.charAt(2)) {
    return (float)1/(1+Math.abs(Integer.parseInt(s1.substring(2))
        - Integer.parseInt(s2.substring(2))));
}
return 0;
```

Gdzie $s1$ i $s2$ są elementami, które ze sobą porównujemy.

2.4. Czwarty sposób ekstrakcji

Wektor w czwartym sposobie ekstrakcji składa się z 6 elementów. Pierwsze 4 są generowane poprzez podzielenie liczby różnych końcówek słów znajdujących się na końcu linii(kolejno w w 1 liczbie końcówka składa się z 1 litery w 2 z 2 liter w 3 z 3 liter a w 4 z 4 liter) poprzez liczbę linijek. Piąty element stanowi maksymalną różnicę w liczbie sylab pomiędzy linijkami, podzieloną poprzez maksymalną liczbę sylab w jednej linijce. Ostatą wartość tworzy się poprzez podzielenie maksymalnej różnicy długości w literach linijek w tekście, poprzez maksymalną liczbę liter w linijce. Wektor cech w czwartym sposobie ekstrakcji przyjmuje postać:

$$v = [0.1, 0.3, 0.5, 0.1, 0.8, 0.5]$$

2.5. Pozostałe wzory użyte w programie

Obliczenia odległości dokonano w trzech metrykach.

Pierwszą z nich jest metryka Euklidesa, odległość d obliczana jest przy pomocy wzoru, gdzie n oznacza długość wektora w przestrzeni knn:

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

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:

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

Podobieństwo pomiędzy poszczególnymi słowami wyznaczaliśmy przy pomocy dwóch spobów: Pierwszym z nich była uogólniona miara n-gramów. Obliczana jest przy pomocy wzoru:

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

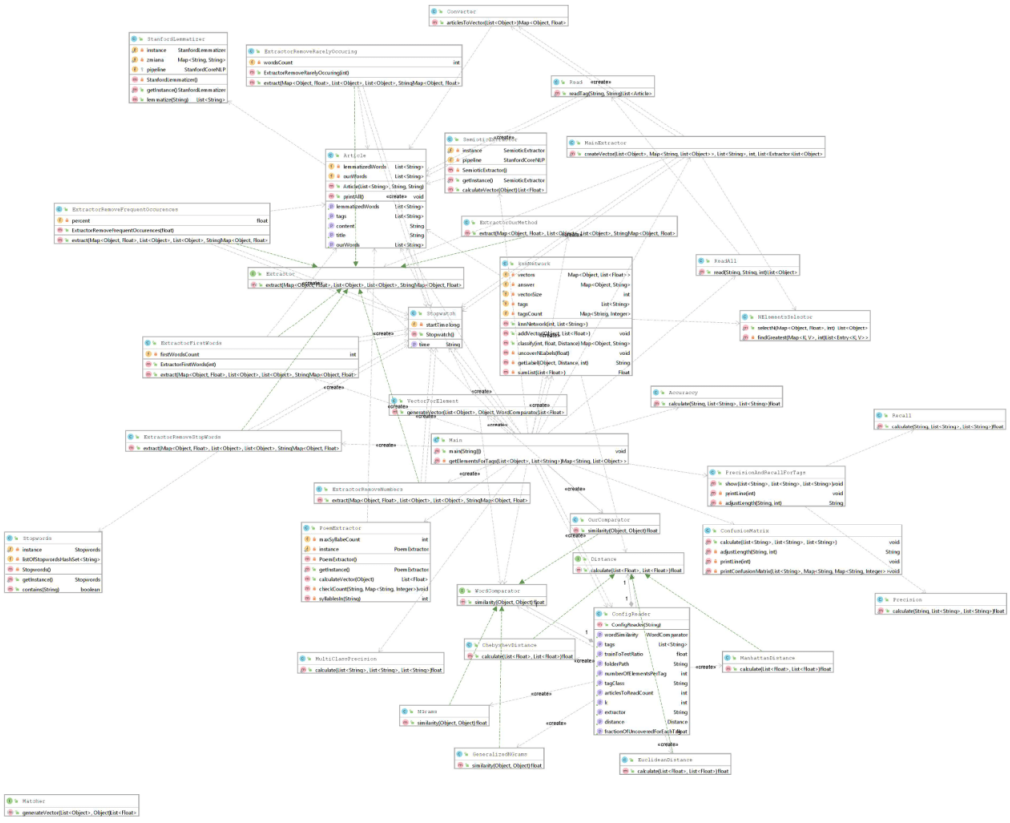
Drugim były natomiast trigramy. Ich użycie miało na celu pokazanie przewagi uogólnionej miary n-gramów nad miarą z jednym jasno określonym n. Obliczaja jest przy pomocy wzoru:

$$sim_n(s_1, s_2) = \frac{1}{N - n + 1} \sum_{i=1}^{N-n+1} h(i)$$

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.

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:
— tagClass - tag dla którego etykiety będzie nadawał program
— folderPath - ścieżka do folderu z plikami z danymi



Rysunek 1. UML Diagram

— articlesToReadCount - liczba plików z artykułami, które program ma wczytać
— k - liczba najbliższych sąsiadów według których algorytm będzie klasyfikował
— fractionOfUncoveredForEachTag - część elementów należących do każdej z etykiet, która ma zostać odkryta w klasyfikacji knn
— tags - etykiety, według których program ma klasyfikować
— numberOfElementsPerTag - liczba elementów jakie ma zawierać w sobie cecha dla każdej z etykiet
— trainToTestRatio - stosunek zbioru treningowego do testowego
— distanceKNN - metryka pomiaru dystansu w przestrzeni dla algorytmu knn
— wordSimilarity - metryka podobieństwa słów
— 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, 13, 21), 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)(13766 elementów), przy stosunku zbioru treningowego do testowego 60%(8260 elementów) do 40%(5506 elementów) i dla 10%(551 elementów) odkrytych etykiet w zbiorze testowym. W drugim przypadku według tagu TOPICS etykietowano przy pomocy dwóch etykiet (coffee i gold), przy równym podziale na zbiór treningowy i testowy. 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%(10 etykiet) odkrytych etykiet w zbiorze testowym. Dodatkowo do przetestowania czwartego sposobu ektrakcji użyto zbioru własnego składającego się w połowie z wierszy Williama Blake'a a w połowie z artykułów Reutera, jego liczebnośc wynosi 100 elementów, jest on podzielony na dwie części treningową i testową z czego w testowej odkryte jest 20% etykiet(10 etykiet).

Do zbadania wyników klasyfikacji posłużyliśmy metodą tablicy pomyłek (zwaną również macierzą błędów, ang. confusin matrix). Pozwala ona ocenić jakoś klasyfikacji. Poniżej znajduje się tabela wyjaśniająca sposób jej tworzenia.

Klasa rzeczywista	Klasyfikacja jako Pozytywna	Klasyfikacja jako Negatywna
Pozytywna	Prawdziwie Pozytywna (ang. True Positive, TP)	Fałszywie Negatywna (ang. False Negative, FN)
Negatywna	Fałszywie Pozytywna (ang. False Positive, FP)	Prawdziwie Negatywna (ang. True Negative, TN)

Tablica 1: Macierz błędów

Skrótów angielskich nazw będziemy używać dalej przy określaniu miar. Znając te wartości jesteśmy w stanie obliczyć kolejne miary:

1. Accuracy
2. Precision
3. Recall (Sensitivity)

opisane poniżej.

4.1. Accuracy

Dokładność (ang. Accuracy) jest ogólną efektywnością klasyfikacji. Wyraża się wzorem:

$$Accuracy = \frac{tp + tn}{tp + fn + fp + tn}$$

4.2. Precision

Precyzja (ang. Precision) oznacza zgodność etykiet z pozytywnie zaklasyfikowanymi cechami.

$$Precision = \frac{tp}{tp + fp}$$

4.3. Recall

Czułość (ang. Recall lub Sensitivity) jest to stosunek poprawnie zaklasyfikowanych pozytywnych przypadków do liczby pozytywnych przyadków w zbiorze

$$Recall = \frac{tp}{tp + fn}$$

5. Wyniki

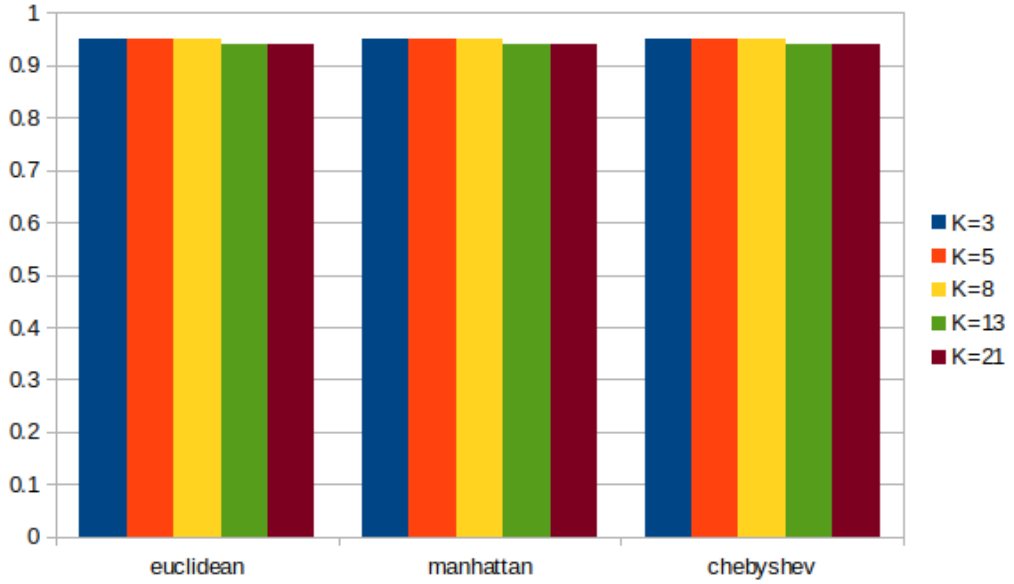
Wyniki kolejnych przeprowadzanych eksperymentów zostały umieszczone poniżej.

5.1. Wyniki klasyfikacji dla tagu PLACES

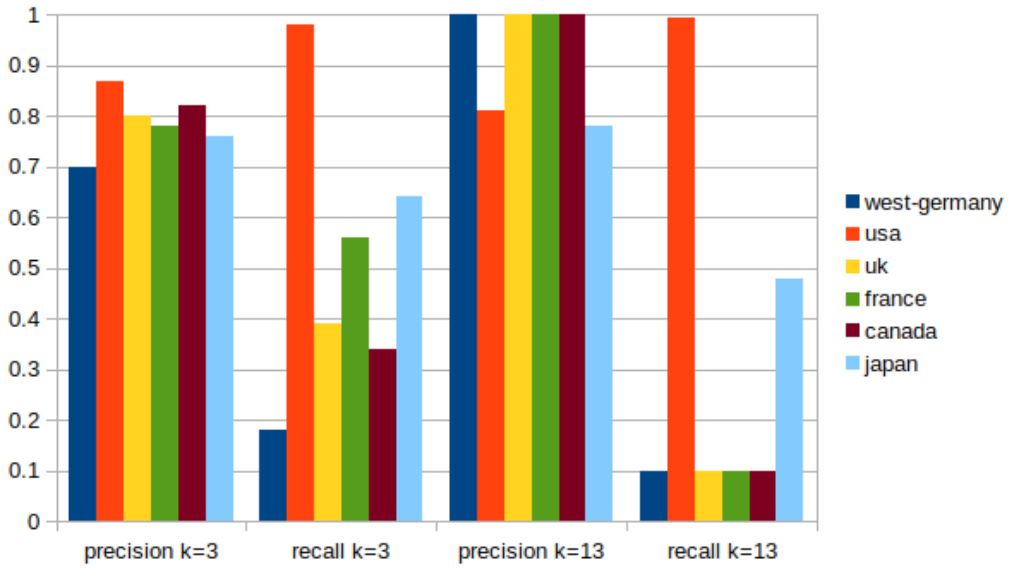
Wyniki eksperymentów dla tagu PLACES zostały przedstawione w tabeli **??** oraz na rysunkach od **??** do **??**.

	west-germany	usa	uk	france	canada	japan
west-germany	59	99	2	5	0	4
usa	6	5334	47	21	26	39
uk	1	275	156	8	9	7
france	0	26	3	90	0	7
canada	0	252	7	1	165	1
japan	0	71	2	1	1	159

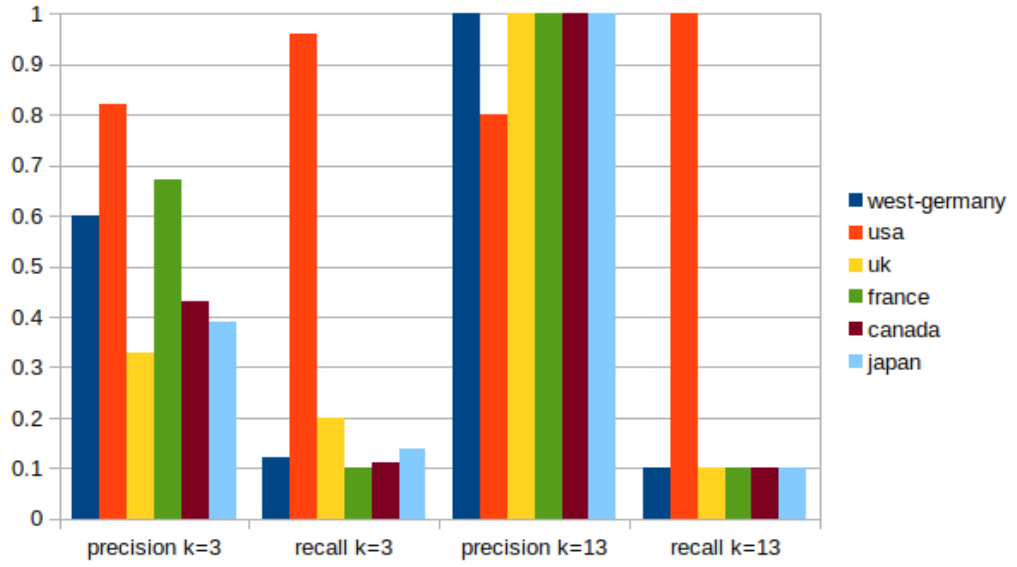
Tablica 2: Confusion Matrix dla tagu PLACES i etykiet: west-germany, usa, uk, france, canada, japan.



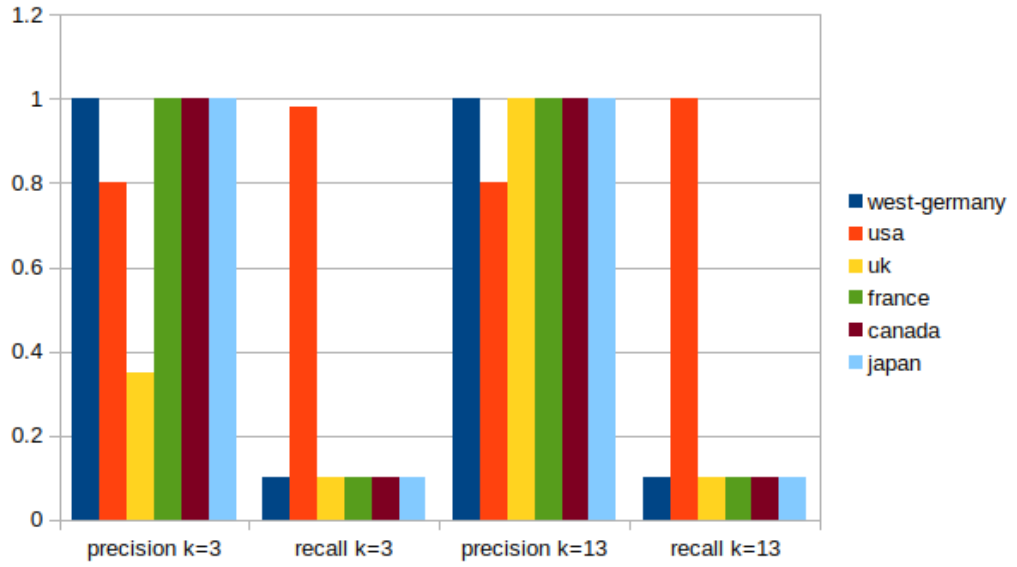
Rysunek 2. Dane przedstawiające wartość parametru Multi-Class Accuracy dla wybranych wartości k i trzech zawartych w programie metryk.



Rysunek 3. Wyniki ekstrakcji pierwszym ekstraktorem przedstawione poprzez wartości parametrów precision i recall dla dwóch wartości k(3 oraz 13) dla sześciu etykiet(west-germany, usa, uk, france, canada, japan).



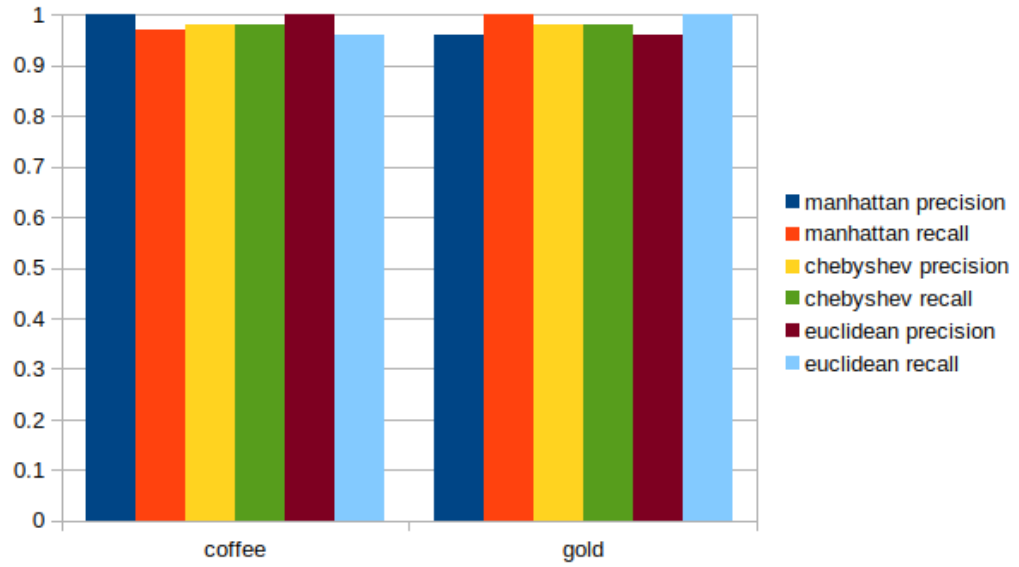
Rysunek 4. Wyniki ekstrakcji drugim ekstraktorem przedstawione poprzez wartości parametrów precision i recall dla dwóch wartości k(3 oraz 13) dla sześciu etykiet(west-germany, usa, uk, france, canada, japan).



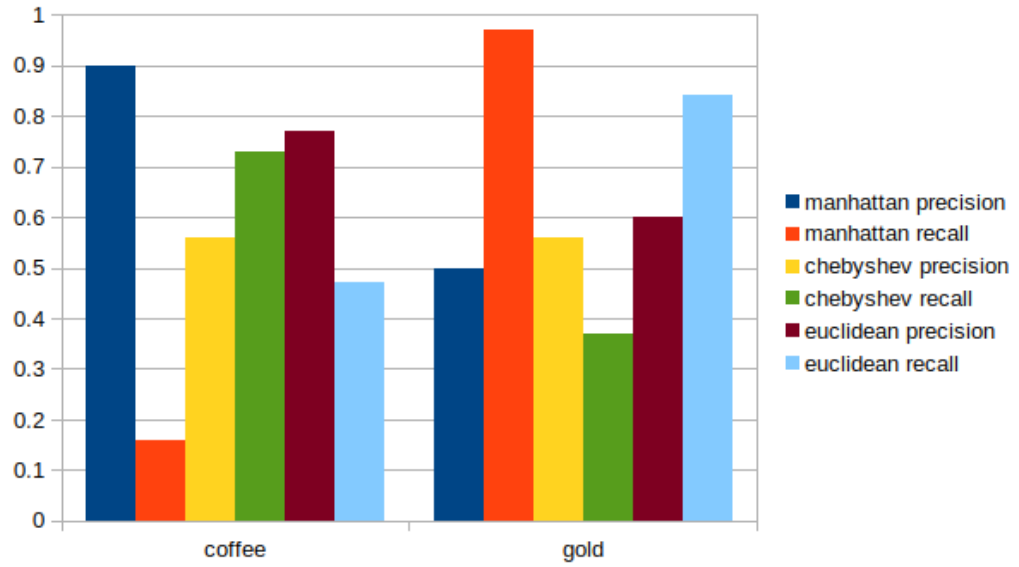
Rysunek 5. Wyniki ekstrakcji trzecim ekstraktorem przedstawione poprzez wartości parametrów precision i recall dla dwóch wartości k(3 oraz 13) dla sześciu etykiet(west-germany, usa, uk, france, canada, japan).

5.2. Wyniki klasyfikacji dla tagu TOPICS

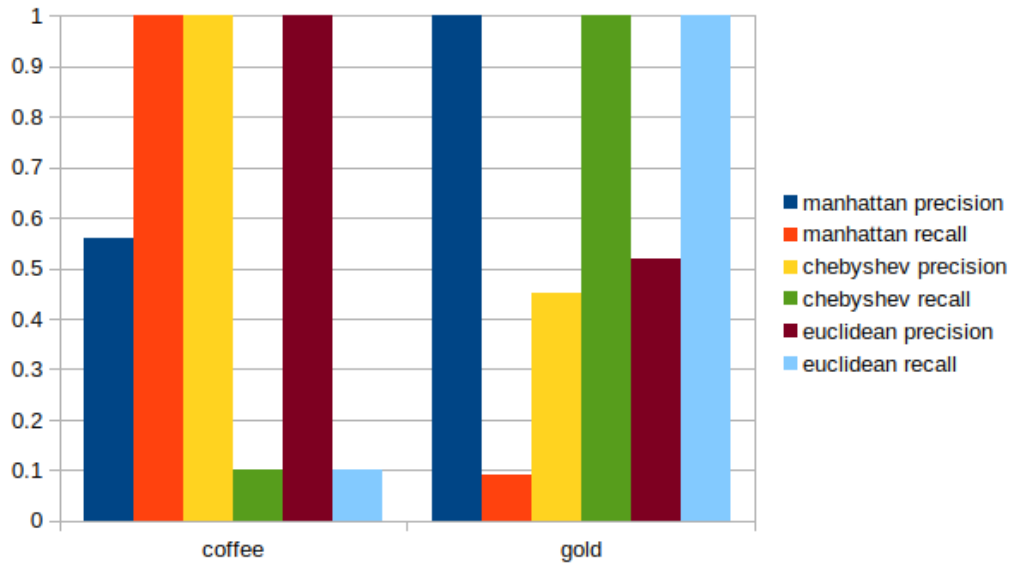
Wyniki eksperymentów dla tagu TOPICS zostały przedstawione na rysunkach od ?? do ??.



Rysunek 6. Wyniki ekstrakcji pierwszym ekstraktorem przedstawione przy poprzez wartości precision i recall dla trzech różnych metryk(manhattana, czebyszewa i euklidesa) dla dwóch etykiet(coffee, gold).



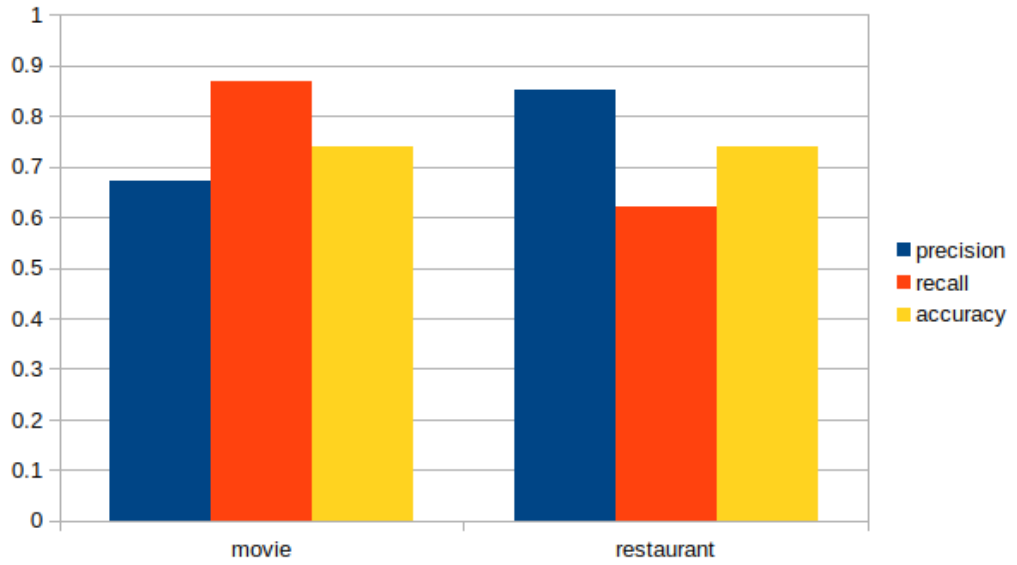
Rysunek 7. Wyniki ekstrakcji drugim ekstraktorem przedstawione przy poprzez wartości precision i recall dla trzech różnych metryk(manhattana, czebyszewa i euklidesa) dla dwóch etykiet(coffee, gold).



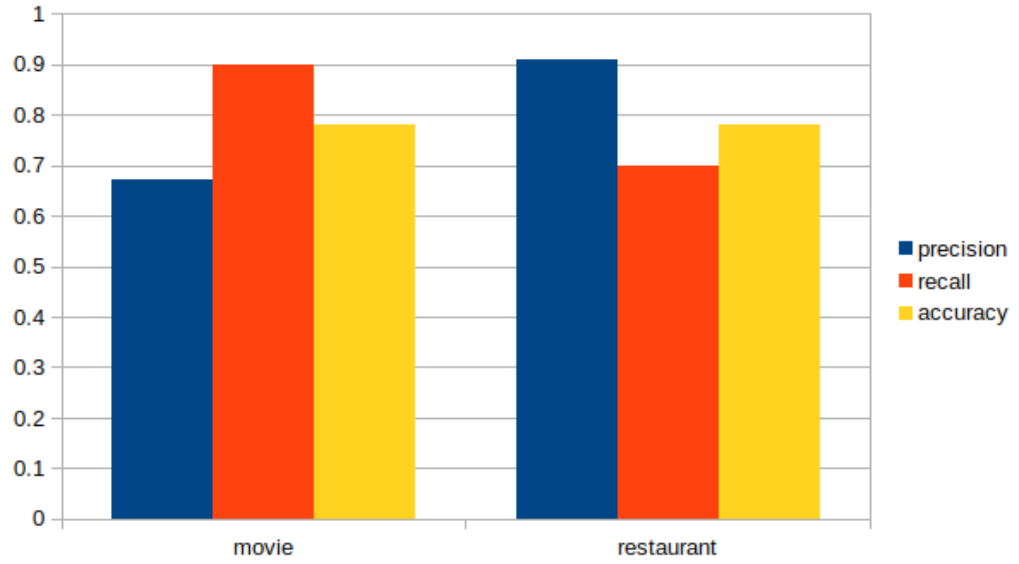
Rysunek 8. Wyniki ekstrakcji trzecim ekstraktorem przedstawione przy poprzez wartości precision i recall dla trzech różnych metryk(manhattana, czebyszewa i euklidesa) dla dwóch etykiet(coffee, gold).

5.3. Wyniki klasyfikacji dla tagu REVIEWS

Wyniki eksperymentów dla tagu REVIEWS zostały przedstawione na rysunkach ?? i ??.



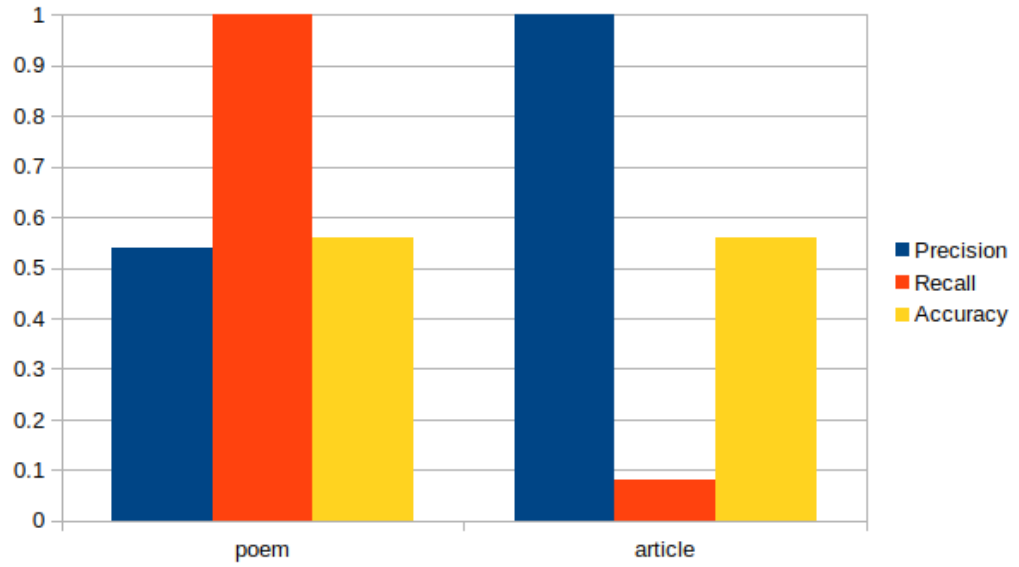
Rysunek 9. Wyniki ekstrakcji pierwszym ekstraktorem przedstawione poprzez wartości parametrów precision, recall i accuracy dla miary podobieństwa trigramów dla dwóch etykiet(movie, restaurant).



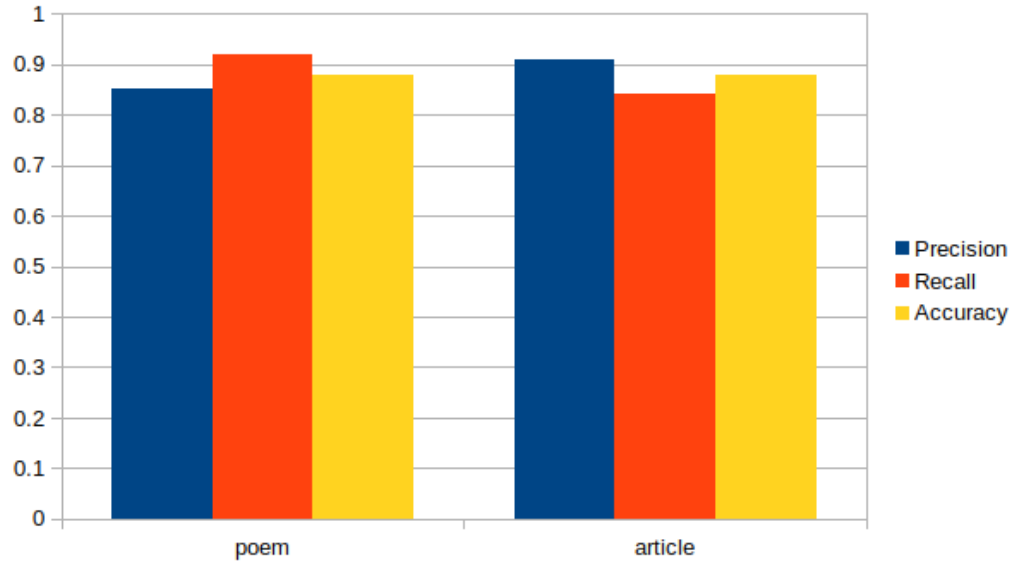
Rysunek 10. Wyniki ekstrakcji pierwszym ekstraktorem przedstawione poprzez wartości parametrów precision, recall i accuracy dla miary podobieństwa uogólnionej miary n-gramów dla dwóch etykiet(movie, restaurant).

5.4. Wyniki klasyfikacji dla tagu LITERATURE

Wyniki eksperymentów dla tagu LITERATURE zostały przedstawione na rysunkach ?? i ??.



Rysunek 11. Wyniki ekstrakcji pierwszym ekstraktorem przedstawione poprzez wartości precision, recall i accuracy dla dwóch etykiet(poem, article).



Rysunek 12. Wyniki ekstrakcji czwartym ekstraktorem przedstawione poprzez wartości precision, recall i accuracy dla dwóch etykiet(poem, article).

6. Dyskusja

6.1. Uwagi ogólne

Napisanie programu klasyfikującego uświadomiło nam, że w rozpoznawaniu danego tekstu nie liczą się jedynie znaczenia poszczególnych słów. Pozytywny wpływ na jakość klasyfikacji miało wykorzystanie w pierwszym ekstraktorze cech wybierania słów kluczowych poprzez preferowanie tych znajdujących się bliżej początku tekstu. Właściwość ta jest związana z podstawowymi założeniami artykułu prasowego, i w tekstach opracowanych przez nas recenzji, które mają na celu w pierwszych słowach określić kontekst danego tekstu. Zdajemy sobie sprawę, że dla innego typu tekstu pisanego, takiego jak na przykład opowiadanie cecha ta niekoniecznie będzie obecna, nastąpi wtedy konieczność stworzenia nowego modelu ekstrakcji cech, do którego przyjęcia nasz program jest zdolny.

Bardzo ważne, przy tworzeniu programów do klasyfikacji tekstów jest rozróżnienie cech językowych od cech metajęzykowych. Poprzez cechy językowe rozumiemy cechy odnoszące się do semantyki. Semantyką nazywamy naukę o budowie wyrazów. Poprzez cechy metajęzykowe rozumiemy wszystkie cechy abstrahujące od znaczenia słów, a skupiające się na ich ułożeniu, długości, rymach, czy liczbie sylab. W pierwszym i trzecim sposobie ekstrakcji użyliśmy zarówno cech językowych jak i metajęzykowych. Natomiast w drugim sposobie użyliśmy wyłącznie cech metajęzykowych.

Znacząco większa liczebność artykułów z etykietą usa negatywnie wpływa na klasyfikację klas z etykietami: west-germany, uk, france, canada i japan. Jest to szczególnie widoczne w tabeli **??** przedstawiającej confusion matrix. Dodatkowo w przypadku znacznej przewagi liczebnej jednej z etykiet przydatność parametru accuracy spada, co widać na rysunku **??**.

Rezultaty dla przygotowywanych przez nas tekstów(Rysunki: **??**, **??**) były znacząco niższe niż dla tekstów zaczerpniętych z bazy Reutera(Rysunki: **??**, **??**, **??**). Jest to zapewnie spowodowane spójnym stylem i określoną formą tekstów Reutera, podczas gdy przygotowane przez nas teksty znacząco się różniły pod względem stylistycznym jak i formalnym.

6.2. Wyższość pierwszego sposobu ekstrakcji nad drugim i trzecim.

Zauważyliśmy znaczącą różnicę pomiędzy wartościami współczynnikow precision, recall i accuracy pomiędzy pierwszym(Rysunek **??**) a drugim(Rysunek **??**) i trzecim(Rysunek **??**) sposobem ekstrakcji na korzyść pierwszego sposobu ekstrakcji.

Wyniki lepsze niż drugi sposób ekstrakcji, sposób pierwszy osiągnął dzięki użyciu w nim cech zarówno językowych jak i metajęzykowych.

Niestety nasza próba uproszczenia sposobu zapisu słów kluczowych w trzecim sposobie ekstrakcji okazała się nietrafiona. Sposób w jaki postanowiliśmy zapisać słowa kluczowe całkowicie nie jest wystarczający, aby przechować wystarzającą informację.

6.3. Zrozumienie idei czwartego sposobu ekstrakcji.

Jak już zostało napisane powyżej w ekstrakcji cech z tekstów ważne jest aby odróżniac cechy języka od cech metajęzyka. Rozróżnienie to jest kluczowe zwłaszcza przy próbie zrozumienia dlaczego czwarty ektraktor działa tak dobrze.

W czwartym ekstraktorze wykorzystane zostały cechy struktury lirycznej jaką jest wiersz(eng. poem). Są to rymy o różnej, acz rozsądnej długości, stała, bądź bardzo zbliżona do siebie liczba sylab w kolejnych linijkach oraz mała różnica w długości linijek. Dla odmiany artykuły nie zawierają w sobie żadnej z wyżej wymienionych cech. Posłużyliśmy się zatem cechami metajęzykowymi.

Dla porównania cechy języka, przy pomocy których operuje najskuteczniejszy w eksperymentach na tagach PLACES, TOPICS oraz REVIEWS pierwszy ekstraktor, nie dają specjalnie wysokich wyników w przypadku tekstów z tagu LITERATURE(Rysunek **??**) w porównaniu do wyników czwartego ekstraktora(Rysunek **??**). Jest to spowodowane ich różnorodnością i brakiem wspólnego znaczenia semantycznego.

7. Wnioski

Stworzenie systemu ekstrakcji cech dla danego rodzaju bądź zbioru tekstów jest zadaniem o wiele trudniejszym niż przypuszczaliśmy i wymaga nie tylko wiedzy o słowach jakie mogą się znaleźć w danym tekście, ale i dogłębnej wiedzy o wybranym typie tekstu, jak w naszym przypadku artykule jak i w wybranym przez nas, podobnej jeśli chodzi o cechy recenzji.

Algorytm k najbliższych sąsiadów ma problem, w momencie, kiedy jedna z etykiet ma znacząco więcej przyporządkowanych do niej elementów. Ważne jest wtedy aby odpowiednio dobrać wartość k aby nie była zbyt wysoka. Należy też wtedy zadbać o to, aby do zbioru początkowo odkrytych etykiet dla elementów trafiły etykiety każdego rodzaju.

Najlepiej radzą sobie algorytmy korzystające zarówno z cech językowych jak i metajęzykowych.

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] Dokumentacja Stanford CoreNLP https://stanfordnlp.github.io/CoreNLP
[6] Adam Niewiadomski. *Materiały, przykłady i ćwiczenia do przedmiotu Komputerowe Systemy Rozpoznawania*, 21 września 2009.