Wstęp

Jednym z typowych zastosowań metod sztucznej inteligencji i uczenia maszynowego jest przetwarzanie języka naturalnego (ang. Natural Language Processing, NLP), której typowych przykładem jest analiza dokumentów tekstowych. Można tutaj wyróżnić takie zagadnienia jak:

- grupowanie danych – odnajdywanie w zbiorze dokumentów, tekstów o podobnej tematyce
- wyszukiwanie dokumentów podobnych do wzorca – można to również traktować jako wyszukiwarka dokumentów, lub system antyplagiatowy. Idea polega na podaniu dokumentu wzorcowego i odnalezieniu dokumentów najbardziej podobnych do zadanego wzorca.
- klasyfikacja dokumentów – dysponując zbiorem treningowym, zawierającym dokumenty podzielone na odpowiednie zbiory kategorii, system uczący się powinien nauczyć się rozpoznawać kategoryzacji nowych dokumentów

Wszystkie z przedstawionych zadań rozwiązywalne są na bazie dotychczas poznanych algorytmów. Jednakże kluczem do poprawnego ich rozwiązania jest odpowiednia reprezentacja dokumentów tekstowych w postaci wektora, tak iż pojedynczy dokument tekstowy powinien być zapisany w postaci zbioru atrybutów o określonych wartościach. (plaska struktura danych).

Najpowszechniej stosowanym rozwiązaniem jest tutaj reprezentacja tekstów w postaci częstości występowania wyrazów (ang. term frequency). Idea ta polega na stworzeniu zbioru wyrazów składających się na danych dokument tekstowy i wyznaczeniu częstości występowania każdego z nich. Należy tutaj nadmienić, iż w tej sytuacji dokumenty związane z określoną tematyką będą miały wyraz, które nie występują w dokumentach tekstowych poświęconych innej tematyce lub występują znacznie częściej. Np. częstotliwości wyrazów: *komputer, płyta główna, karta graficzna* w dokumentach poświęconych budowie komputerów będą znacznie większe niż w dokumentach poświęconych tematyce sportu, co więcej większość z nich nie będzie tam występowała, czyli ich częstotliwość będzie równa 0. Podobna relacja zachodzi również w sytuacji odwrotnej gdzie takie wyrazy jak: *bieżnia*, *skocznia, piłka,* będą występowały głównie w dokumentach poświęconych tematyce sportu, a nie komputerów. Należy również pamiętać, iż analizie przez algorytmy inteligencji obliczeniowej poddawane są nie pojedyncze wyrazy, a całe wektory, w których algorytm patrzy (analizuje) częstotliwości określone grupy wyrazów.

Podejście *term frequency (TF)* wymaga jednak odpowiedniego przygotowania analizowanego dokumentu. Na ten proces składają się:

- Tokenizacja - polega na rozbiciu dokumentu tekstowego na poszczególne wyrazy. Na tym etapie należy zwrócić szczególną uwagę na znaki interpunkcyjne, które zgodnie z ogólnie przyjętymi standardami są doklejane na końcach wyrazów. Innym przykładem powyższych problemów jest np. zagadnienie dzielenia wyrazów.
- Ujednolicenie wielkości liter - doprowadzenie do sytuacji, w której wszystkie litery w wyrazie mają ten sam rozmiar
- Usunięcie wyrazów nieistotnych – w języku naturalnym często występują wyrazy nie wnoszące istotnej informacji z punktu widzenia wcześniej zdefiniowanych zadań, są nimi np. zaimki, spójniki itp. W innych językach dochodzą również takie elementy jak np. *the, a* (j. Angielski) itp. Analiza ich częstotliwości nie wnosi więc

żadnej szczególnie istotnej informacji. Dlatego też takie wyrazy należy odfiltrować na etapie przygotowywania danych.

- Operacje poboczne – do tej grupy operacji należy włączyć zagadnienia analizy wyrazów z pominięciem znaków diakrytycznych typowych dla różnych języków. Coraz częściej autorzy tekstów np. sms, e-maili, wpisów na blogach ignorują zasady pisowni związane ze stosowaniem znaków diakrytycznych, posługując się jedynie znakami podstawowymi co prowadzi do braku jednoznaczności w analizie takich wyrazów jak np. *ćwiczyć* i *cwiczyc* Innym problemem są również błędy ortograficzne i literówki. Zagadnienie to komplikuje dodatkowo fakt iż pewne wyrazy w formie „bez symboli diakrytycznych" mogą mieć zupełnie inne znaczenie np. *zęby* i *żeby*. W obydwu tych przypadkach wyraz bez znaków diakrytycznych przyjmuje formę *zeby* pomimo, iż każdy z nich ma inne znaczenie. Integrację wyrazów należy więc robić w sposób bardzo ostrożny. Jednym ze sposobów radzenia sobie w w/w problemem jest rozwiązanie słownikowe, gdzie definiowane są twarde reguły identyfikacji wyrazów. Innym rozwiązaniem jest np. wykorzystanie odległości Levensteina, która umożliwia obliczenie odległości pomiędzy dwoma wyrazami analizując na ile są one do siebie podobne. Mała różnica w odległości Levensteina może oznaczać literówkę, lub różnicę wynikającą z nie stosowania lokalnych znaków diakrytycznych
- Lematyzacja – jest kluczowym elementem analizy procesu przekształcania dokumentów tekstowych. Polega ona na doprowadzeniu każdego termu (wyrazu) do tzw. korpusu. Korpus może być tutaj rozumiany jako forma bezosobowa, bezokolicznik itp. Lepatyzacja związana jest z dużą różnorodnością form każdego z wyrazów. Przykładowo w j. Polskim występuje podział na osoby: forma męska, żeńska i nijaka np. pojechałem, pojechałam, pojechało innym przykładem jest odmiana przez przypadki np. *telefon, telefonu, telefonowi, telefonie, telefonem itp.* Zagadnienie to rozwiązuje się zwykle poprzez wykorzystanie metod słownikowych, lub tez przez integrację metod słownikowych z innymi metodami np. wykorzystującymi wspomnianą wcześniej odległość Levensteina

Po dokonaniu powyższych przekształceń, kolejnym krokiem analizy dokumentów tekstowych jest integracja każdego z dokumentów w zbiór danych. Proces ten realizowany jest poprzez wyznaczenie złączenia listy atrybutów każdego z dokumentów.

Uzyskany tą drogą zbiór danych można poddać analizie z wykorzystaniem dowolnej z grup metod inteligencji obliczeniowej. Typowe zastosowania w tyej grupie metod skłaniają do wykorzystania metod opartych na odległościach jak kNN, algorytmy grupowania itp. Jednakże iż zastosowanie narzuca pewne ograniczenia. Wynikają one z właściwości odpowiednich miar odległości. Dla wyżej opisanej reprezentacji zastosowanie typowych miar odległości jak Euklidesa może powodować zupełnie błędne działanie systemu. Sytuacja taka maże mieć miejsce jeśli zostaną poddane dokumenty o rożnej długości. Wówczas dla dokumentów dłuższych częstosliwości występowania poszczególnych wyrazów będą znacznie większe niż częstotliwości występowania wyrazów w dokumentach krótkich. W takiej sytuacji należy albo zastosować miary odległości które w sposób automatyczny dokonują normalizacji jak odległość kosinusowa, albo też dokonać wstępnej normalizacji każdego z wektorów zapewniając by każdy z wektorów miał jednostkową długość.

Kolejną możliwością poprawy jakości systemów predykcyjnych jest koncepcja TF/IDF, która polega na wyznaczeniu ilorazów częstotliwości występowania pojedynczych wyrazów w dokumencie w stosunku do częstotliwości występowania tego wyrazu w całej grupie dokumentów. Można to zapisać jako wyznaczenie TF:

1. Sposób 1:

$$TF(t_i, d_i) = \begin{cases} 0 & \text{dla } n_{ij} = 0 \\ \frac{n_{ij}}{\sum_{k=1}^{m} n_{kj}} & \text{dla } n_{ij} > 0 \end{cases}$$

2. Sposób 2:

$$TF(t_i, d_i) = \begin{cases} 0 & \text{dla } n_{ij} = 0 \\ \frac{n_{ij}}{max_k n_{kj}} & \text{dla } n_{ij} > \end{cases}$$

gdzie:

n_{ij} – liczba wystąpień danego termu ti w dokumencie

n_{kj} liczba wszystkich termów w dokumencie

oraz wartości współczynnik IDF jako:

1. Sposób 1:

$$IDF(t_i) = \frac{D}{D_{ti}}$$

2. Sposób 2:

$$IDF(t_i) = \log \frac{1 + |D|}{|D_{ti}|}$$

gdzie:

D - zbiór wszystkich dokumentów

D_{ti} - zbiór dokumentów w których wystąpił term ti

Przeprowadzenie powyżej opisanych kroków pozwala na przekształcenie zbioru dokumentów do postaci tabelarycznej, co z kolei umożliwia zastosowanie metod uczenia maszynowego poznanych na wcześniejszych zajęciach do realizacji na wstępie zdefiniowanych celów.

Zadania

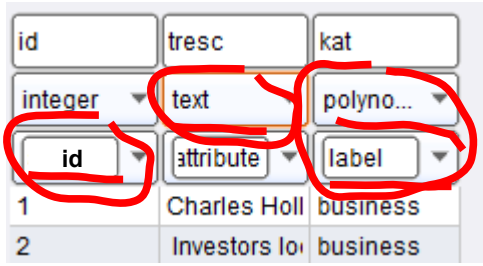
Przygotowanie zbiorów danych

1. Wykorzystując operator *Create document* stwórz prosty dokument tekstowy (najlepiej w języku angielskim)
2. Sprawdź wpływ różnych wybranych przez siebie metod przetwarzania dokumentów na jakość uzyskiwanych wyników. Na tym etapie możliwe jest określenie operacji przetwarzających tekst. Możliwymi operatorami są tutaj:
    - Tokenizacja – można ją zrealizować za pomocą operatora *Tokenize*, który przyjmuje na wejściu dokument tekstowy i przekształca go na odpowiadający zbiór term

- Transformacja wielkości liter – realizowana przez operator *Transform Cases* – umożliwia ujednolicenie wielkości liter w dokumencie
- Lematyzacja – realizowana przez zbiór operatorów zawartych w *Text Processing* → *Stemming.* Algorytm *Stem (WordNet)* wymaga załadowania słownika opartego na projekcie WordNet. Można w tym celu wykorzystać operator *Open WordNet* jednocześnie pobierając bazę danych ze strony: http://wordnet.princeton.edu/wordnet/download/current-version/#nix

  (plik: Download just database files: WNdb-3.0.tar.gz)
- Filtracja – filtracja umożliwia wstępne odfiltrowanie tokenów poprzez wybranie jednego z operatorów znajdujących się w *Text Processing* → *Filtering* które umożliwiają filtrację tokenów po ich długości, lub typie (*Filter Stopwords*)

3. Otwórz dokument Excela *teksty.xls* i sprawdź jego zawartość
4. W programie RapidMinera (RM) wczytaj dokument *teksty.xls* (*Excel Read*), w ostatnim kroku kreatora importu pamiętaj aby odpowiednio zaznaczyć typy kolumn:



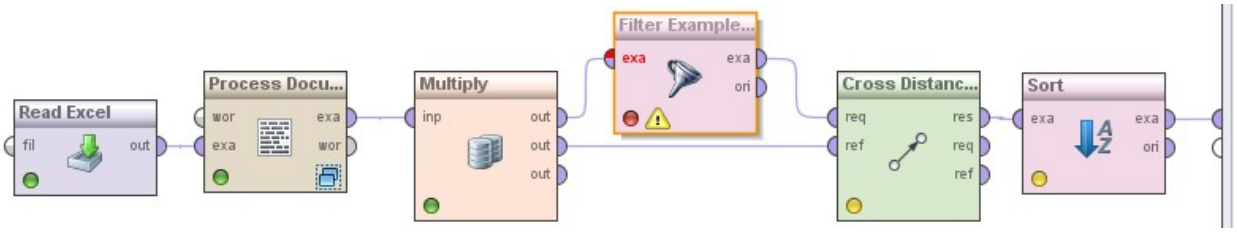
   Przejrzyj zawartość dokumentu w RM
5. Dodaj operator *Process documents from Data* i podłącz wczytany zbiór danych na wejście tego operatora. Dokonaj odpowiedniej konfiguracji:
   - *vector creation* – typ reprezentacji dokumentów tekstowych. TF-IDF oraz Term-Frequency – patrz powyższa dokumentacja, *Term occurances* – forma reprezentacji gdzie w postaci 0/1 oznacza się czy dany wyraz wystąpił czy też nie, innymi słowy jest to uproszczona forma TF
   - *prune method* – sposób oczyszczania danych – realizuje *Usunięcie wyrazów nieistotnych* poprzez jedną z kilku metod:
     - *procentual* - procentowe określenie udziału ilości występowania danego słowa w zbiorze dokumentów,
     - *absolute* – poprzez określenie bezwzględnych wartości częstości występowania poszczególnych wyrazów,
     - by ranking – poprzez usunięcie procentu najrzadszych i najczęstszych słów
6. Na podstawie wiedzy z zad 2. w operatorze *Process documents from Data* skonfiguruj odpowiedni proces tokenizacji oraz filtrowania i lematyzacji tokenów. Sprawdź uzyskane wyniki

Wyszukiwanie dokumentów podobnych

1. Przygotowany w przednim zadaniu proces wykorzystaj do wyszukiwania dokumentów podobnych. W tym celu odfiltruj ze zbioru danych pojedynczy

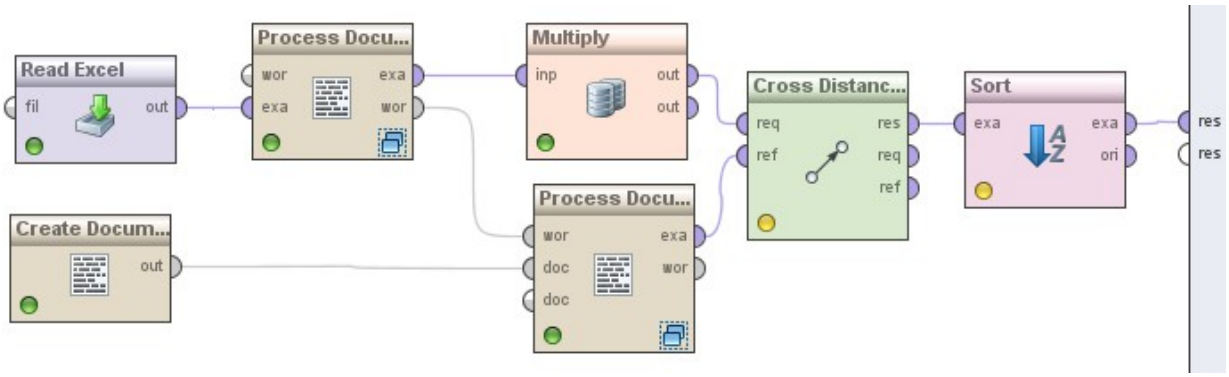
przypadek za pomocą operatora *Filter example range.* Następnie podłącz odfiltrowany przypadek na wejście operatora *Cross Distance*, a jego wyjście *res* podłącz na wejście operatora *Sort*. Operator *Sort* skonfiguruj tak aby dane sortowane były wg kolumny *distance* w porządku narastającym Wyjście z operatora sort zwróć na wyjście procesu. Taka konfiguracja procesu powinna zwrócić obiekty od najbardziej do najmniej podobnego, gdzie w zbiorze wyjściowym w kolumnie *document* znajduje się indeks najbardziej podobnego dokumentu. Sprawdź poprawność uzyskanych wyników.

Proces przedstawia poniższy rysunek:



2. Tworzenie wyszukiwarki: Ustaw pametr *Vector creation* operatora *Process documents form data* na *Term Occurreances*. Wykorzystując operator *Create document* wpisz szukaną frazę. Dodaj operator *Proces Documents* i skonfiguruj go identycznie z operatorem *Process documents form data.* Następnie uzyskany zbiór danych podłącz na wejście *req* operatora *Cross Distance.* Następnie posortuj wyniki jak w zad 1. Sprawdź poprawność procesu wyszukiwania.

   Proces przedstawia poniższy rysunek:



   UWAGA: W niniejszym zadaniu jako miarę odległości dobrze jest wykorzystać odległość Jackarda lub kosinusową.

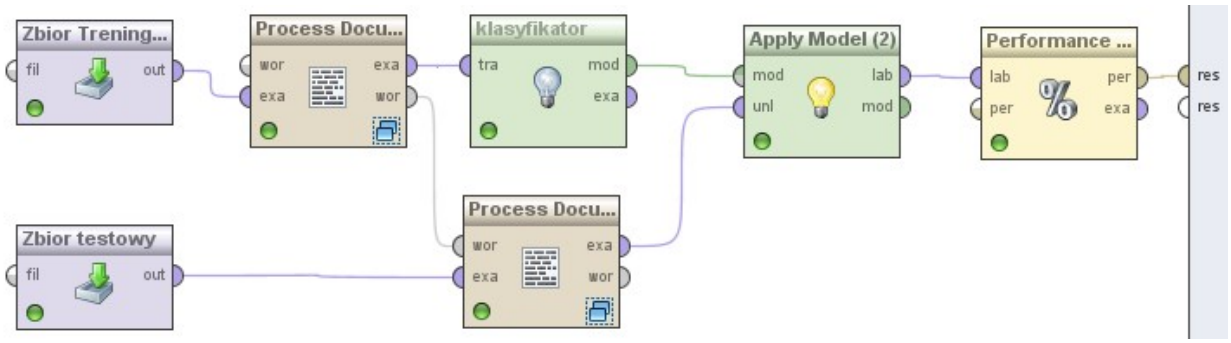
Grupowanie dokumentów

1. Korzystając z metod klasteryzacji dokonaj automatycznego grupowania dokumentów tekstowych. Grupowania dokonaj korzystając z metody k-średnich.
2. Porównaj uzyskane wyniki klasteryzacji z orginalnymi etykietami grup dokumentów.
3. Zbadaj wpływ parametrów konfiguracyjnych procesu tokenizacji i wektoryzacji dokumentow tekstowych na jakość klasteryzacji i jej czas. Pamiętaj że liczba cech znacząco wpływa na czas obliczeń. W tym celu dobrze jest włączyć operację: *Prune method* operatora. Podczas weryfikacji jakości klasteryzacji możesz skorzystać z operatorów *Map Clustering on Labels* oraz operatora *Performance* wykorzystywanego do oceny metod klasyfikacji.

Klasyfikacja dokumentów

1. Na podstawie wcześniejszych eksperymentów przygotuj zbiór danych uczących, a następnie spróbuj zbudować najlepszy możliwy klasyfikator dokonujący automatycznej kategoryzacji treści dokumentów. W tym celu wykorzystaj wiedzę z poprzednich zajęć dotycząca optymalizacji modeli predykcyjnych, sposobów oceny ich dokładności. Pamietaj że na jakość klasyfikacji ma również wpływ zbiór danych oraz sposób przygotowania dokumentów tekstowych.
2. Dysponując klasyfikatorem przygotowanym w poprzednim zadaniu wczytaj zbiór teksty_test.xls i sprawdź poprawność przygotowanego przez siebie klasyfikatora.

   Schemat procesu testowania:



3. W sprawozdaniu zamieść informacje o sposobie realizacji poszukiwania najlepszego klasyfikatora. Jakie klasyfikatory zostały przebadane, dla jakiego zestawu parametrów, jakie etapy preprocesingu zostały wykorzystane (jak zrealizowano funkcję Process Documents) Zamieść też informacje o uzyskanej dokładności na zbiorze testowym i treningowym

Uwaga dla metod bazujących na odległościach (ranking – wyszukiwanie, grupowanie klasyfikacja) zwróć uwagę na typ funkcji odległości, sprawdź czy lepsze wyniki daje odległość Euklidesa, czy może CosineSimilarity.