Dokumentacja końcowa

Zadanie rozpoznawania tekstu w AMOD

Autorzy:   
Karol Gałązka,  
Tomasz Kogut

# Architektura

Ble ble ble ble

# Szczegółowy opis budowy klasyfikatora

BagOfWordsTextClassifier

Udostępnia interfejs:

* int[] ProcedureRecognition(string text)
* int[] NextStagePrediciton(int procedurId, int phaseId, string text)
* int[] NextPersonPrediction(int procedurId, int phaseId, string text)

Poprzez te metody można dokonać kolejno klasyfikacji: tekstu do procedury, następnego etapu procedur na podstawie tekstu, oraz osoby do której należy przekazać sprawę.

Klasyfikator opiera się na typach danych opisanych w punkcie 4. Na ich podstawie przed pierwszym użyciem tworzony jest szereg macierzy i danych pomocniczych, które są wykorzystywane do szybkich obliczeń.

* private DBRepresentation allWords = null;
* private AllCases CasesSet = null;
* private AllProcedures ProceduresSet = null;
* private AllDecisionsPhase PhaseDecisionData = null;
* private AllDecisionsPeople PeopleDecisionData = null;

Dla zmniejszenia złożoności obliczeniowej ograniczamy rozmiar wektora słów, który będzie reprezentował sprawę. Tworzona jest mapa, która słowu przypisuje miejsce w wektorze słów - FetchMeaningfulWords. Brane są pod uwagę tylko te słowa, które występują mniej niż w MaximumFrequency procent spraw. Na podstawie tego współczynnika i ilości słów obliczany jest próg maksymalnej dopuszczalnej liczby wystąpień słowa powyżej, którego słowo jest odrzucane.

* Dictionary<String, int> MapWordToColumn
* const double MaximumFrequency = 0.9;
* int wordThreshold = int.MaxValue;
* void FetchMeaningfulWords()

Ze zbioru procedur wczytywane są wagi słów, które zostały uznane za ważne i umiejscawiane się w macierzy ProcedureMatrix. Wiersze tej macierzy odpowiadają procedurom, kolumny słowom. Kolumny zgadzają się z tymi zapisanymi w MapWordToColumn. Dodatkowo tworzona jest mapa, która odwzorowuje wiersze macierzy na identyfikatory procedur w bazie danych – MapRowToProcedureId, tak by po uzyskaniu najlepszego dopasowania, można było zidentyfikować procedurę.

* double[][] ProcedureMatrix
* int[] MapRowToProcedureId

Ponieważ obliczenia cosinusów odywają się na typie podstawowym double powinny być one szybkie. Do ich wykonywania używany jest stworzona klasa VectorOperations, która operuje na tablicach liczb zmiennoprzecinkowych podwójnej precyzji.

* class VectorOperations
* static double VectorOperations.VectorsConsine(double[] v1, double[] v2)

# Szczegółowy opis dostępu do bazy danych

Ble ble ble

# Sposób reprezentacji wiedzy

Ble ble ble.