Alexandra Vacaru
Grupa 243

Romanian sub-dialect identification

Documentatie proiect

Acest proiect presupune identificarea dialectului corespunzator pentru mai multe propozitii.

Am folosit modelul Tfidf Vectorizer din Scikit-Learn, pentru a procesa datele, transformandu-le intr-o matrice, cu ajutorul unui vocabular ce retine frecventa unor secvente de 1 sau 2 cuvinte din datele de antrenare (ngrams). De asemenea, pentru a clasifica datele de validare si datele de test am folosit Clasificatorul Multinomial Naive Bayes. Acest clasificator are avantajul teoremei lui Bayes, ceea ce duce la o probabilitate mare de a face corect predictiile.

Pentru implementarea acestui algoritm am folosit urmatoarele functii si variabile :

a. ***reading_samples*** - functie pentru citirea datelor din fisier
   - aceasta citeste linie cu linie datele din fisier si creeaza o lista care va retine cheile propozitiilor si o lista care va retine propozitiile
   :param file_path - calea catre fisierul care contine datele
   :return un np array care contine cheile si un np array care contine propozitiile
b. ***reading_labels*** = functie care citeste labelurile
   :param file_path - calea catre fisierul care contine datele
   :return un np array care contine labelurile
c. ***reading_labels*** = functie care citeste labelurile
   :param file_path - calea catre fisierul care contine datele
   :**return** un np array care contine labelurile
d. ***writing_predictions*** - functie pentru a scrie predictiile intr-un fisier txt
   :param file_path - calea catre fisierul in care se vor scrie predictiile
   :param keys - np array care contine cheile propozitiilor clasificate
   :param predictions - np array care contine labelurile propozitiilor clasificate
e. ***get_training_set*** - funtie care imparte datele in propozitii romanesti si moldovenesti folosindu-se de labeluri, si care construieste apoi setul de train, in care fiecare propozitie are labelul corespunzator; aceasta are ca scop pastrarea legaturii propozitie – label dupa ce aplicam metoda shuffle
   :param data - np array ce contine datele de train
   :param labels - np array ce contine labelurile pentru datele de train
   :return training_set - o lista de tupluri, fiecare tuplu continand o propoztie si labelul corespunzator
f. ***get_word_ngrams*** - funtie care imparte textul in ngrams
   :param tokens - lista de cuvinte ce vor fi convertite in ngrams
   :param n - numarul de cuvinte in care vreau sa impart textul
   :var ngrams_list - lista care foloseste functia ngrams din nltk
   :return ngrams_list_clear - lista care contine ngrams sub forma de siruri de caratcere

Alexandra Vacaru
Grupa 243

g. ***get_ngram_samples*** - funtie care imparte fiecare propozitie in cuvinte si returneaza un dictionar folosind ngrams si numarul de aparitii ale acestora
:param sent - propozitia ce va fi convertita in ngrams
:var sentence_tokens - retine o lista de cuvinte
- cuvintele sunt despartite de spatii, space, tab, iar acestea sunt extrase cu ajutorul functiei WhitespaceTokenizer().tokenize(sent)
:var features - dictionarul care va retine ngrams si frecventa acestora
:var unigrams - o lista de stringuri (care contin un singur cuvant)
:var bigrams - o lista de stringuri (care contin cate doua cuvinte consecutive din propozitia resprectiva
:return - un dictionar care transforma fiecare token (cuvant) intr-un feature index in matrice, fiecare cuvant unic, secventa de cuvinte unice primeste un feature index
Scop: creearea unui dictionar de ngrams (mai exact secvente de 1 sau 2 cuvinte); acesta va contine secventa si numarul de aparitii ale acesteia in setul de date

Cu ajutorul functiilor definite mai sus vom retine datele de train (samples si labels), dar si datele de test (validation si test).
:var trainingKeys - np array ce va contine cheile datelor de train (la inceputut fiecarei propozitii exista o cheie care ne ajuta sa facem corespondenta intre propozitii si labelurile acestora)
:var trainingSamples - np array ce va contine datele de train (acestea sunt datele pe care modelul le va invata si cu ajutorul carora va face predictii)
:var trainingLables - np array ce va contine labelurile datelor de train

Cu ajutorul variabilelor trainingSamples si trainingLabels vom contrui setul de antrenare, folosind functia get_training_set.
:var validationKeys - np array ce va contine cheile datelor de validare
:var validationSamples - np array ce va contine datele de validare (aceste sunt datele pe care modelul va face predictii si pentru care vom putea calcula acuratetea)
:var validationLabels - np array ce va contine labelurile datelor de validare (aceste labeluri ne vor ajuta la calcularea acuratetei care ne va spune cat de bune au fost clasificarile facute de modelul nostru)
:var testingKeys - np array ce retine cheile datelor de test (aceste chei ne vor folosi atunci cand vom scrie in fisier predictiile facute de modelul antrenat pe datele de train)
:var testingSamples - np array ce retine datele de test (datele ce vor fi testate pentru competitie)
:var trainingSet - retine datele returnate de functia get_training_set (folosite pentru a antrena modelul)

Alexandra Vacaru
Grupa 243

In continuare, am foloit metoda shuffle pentru a reorganiza ordinea propozitiilor din setul de antrenare.
Scop: ne asiguram ca modelul nu face overfit, si ca reducem varianta.

:var trainingSetSentences - lista de propozitii din datele de train, dupa ce acestea au fost reorganizate
:var trainingSetLabels - lista de labeluri pentru propozitiile din datele de train

Tfidf Vectorizer

Tf-idf (scorul = tf * idf) ne spune cat de importante sunt cuvintele din setul de train in dictionar. De asemenea,valoarea acestuia creste direct proportional cu numarul de aparitii ale cuvintelor in datele utilizate.

Term Frequency (tf) reprezinta numarul de aparitii al fiecarui ngram in datele de train / numarul total de ngrams dintr-o propozitie.

Inverse Document Frequence (idf) determina cat de relevante sunt anumite cuvinte in propozitii, ne ajuta sa minimizam importanta cuvintelor care apar foarte frecvent, precum cuvintele de legatura. Astfel, daca un ngram apre des in multe propozitii, idf-ul scade. Idf foloseste urmatoarea formula : 1 + log(numar propozitii/numar propozitii care contin ngram)

Datele vor fi procesate folosid TfidfVectorizer din Scikit-Learn si functia get_ngram_samples.
Acesta transforma datele intr-o matrice de features numerice.
:param analyzer = get_ngram_samples
get_ngram_samples - functie definita ulterior, care ii spune acestuia sa se uite la cuvinte. De asemenea, acesta functie creeaza dictionarul cu
ngram_range = (1,2), care ii spune modelului sa asigneze scoruri secventelor care au maxim 2 cuvinte si minim 1 cuvant.

TfidfVectorizer returneaza o matrice care mapeaza indexul ngram la scorul tfidf.

TfidfVectorizer
:param smooth_idf = True - adauga 1 la frecventa fiecarui termen din vocabular, astfel prevenim impartirea la 0
:param lowercase = False - transforma toate caracterele in litere mici, inainte sa le imparta in cuvinte
tfidf_vectorizer.fit_transorm - metoda care invata vocabulatul si transforma datele de train intr-o matrice de features numerice
:return
:var trainingSetVectors - matrix sparse - datele transformate intr-o matrice in care fiecare cuvant are asignat un scor calculat dupa cum am mentionat mai sus
- acestea sunt datele ce vor fi invatate de modelul nostru

Alexandra Vacaru
Grupa 243

tfidf_vectorizer.transform - transforma datele intr-o matrice de features numerice,folosindu-se de vocabularul invatat cu ajutorul metodei fit_transform
:return
:var validationSetVectors - matrix sparse - datele transformate in matrice, cu ajutorul a ceea ce contine deja vocabularul construit

Clasificatorul Multinomial Naive Bayes:

- conform documentatiei, modelul Multinomial Naive Bayes are avantajul teoremei lui Bayes, avand astfel cea mai mare probabilitate sa faca corect predictia.

- MultinomialNB() nu are niciun parametru, deoarece overfittingul se face la seturi de
  date de dimensiune mica.

Metoda fit:
:param training_set_vectors (sparse matrix) - datele pe care acestea le va clasifica
:param trainingSetLabels (lista de intregi) - datele conform carora vor fi clasificate cele de antrenare

Metoda predict
:param validationSetVectors (sparse matrix) datele pe care acesta le va clasifica conform clasficarii pe care a invatat-o deja ulterior, cand am apelat metoda fit.
:return np array care contine labelurile conform clasificarii (acestea vor fi 0 pentru propozitiile clasificate ca fiind moldovenesti si 1 pentru propozitiile clasificate ca fiind romanesti)

:var validationPredictions - labelurile pentru datele de validare dupa ce acestea au fost clasificate de model

Scop: Calcularea acuratetii pentru predictiile facute de modelul definit anterior. Principalul scop este ca aceasta sa fie 1. Asta inseamna ca toate predictiile au fost corecte, iar clasificatorul a avut o metoda buna prin care a invatat datele si a aplicat ceea ce a invatat.

Functie: *accuracy_score*
:param validationLabels - np array care contine labelurile corecte pentru datele de validare
:param validationPredictions - np array returnat de metoda predict a modelului, acesta contine clasificarea datelor de validare, in conformitate cu ceea ce a invatat

Alexandra Vacaru
Grupa 243

Astfel, am obtinut o acuratete de 0.73. Aceasta este destul de aproape de 1, ceea ce inseamna ca majoritatea predictiilor au fost corecte.

F1-SCORE = 0.74

Matricea de confuzie ne spune unde greseste modelul. Aceasta evalueaza acuratetea clasificarii.
Un element confusionMatrix[i][j] este egal cu numarul de propozitii care aveau labelul i, dar au fost prezise cu label j.

:var testingSetVectors - matrix sparse - datele de test transfomate in matrice
:var testingPredictions - labelurile pentru datele de test in urma clasificarii realizate de model

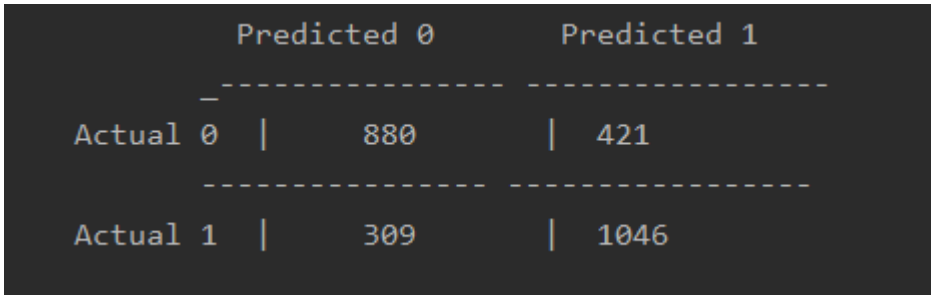
Aplicand acest algoritm pe datele de validare puse la dispozitie in proiect am obtinut un scorul

F1 = 0.74131821 . Am calculat si raportul de clasificare pentru datele de antrenare:

	precision	recall	f1-score	support
0	0.74	0.68	0.71	1301
1	0.71	0.77	0.74	1355
accuracy			0.73	2656
macro avg	0.73	0.72	0.72	2656
weighted avg	0.73	0.73	0.72	2656

Alexandra Vacaru
Grupa 243

Matricea de confuzie pentru acest set de date este urmatoarea:



	Predicted 0	Predicted 1
Actual 0	880	421
Actual 1	309	1046