Modulul de Prelucrarea Datelor

0. Overview

Rolul modulului de prelucrare a datelor in cadrul proiectului este de a normaliza inputul altor componente la un singur format pentru un design SOLID.

O alta contributie a modulului este antrenarea unui model lingvistic statistic bazat pe n-grame cu backoff folosit in recunoasterea vocii.

1. Detalii de implementare

1.0 Cuprins (fisiere):

1.0.0. Conversie audio:

*./Prelucrarea datelor/stripAudio.py*

*./Prelucrarea datelor/stripAudio\_pydub.py*

1.0.1. Parsarea dictionarului de foneme:

*./Prelucrarea datelor/parseDictionary.py*

1.0.2. Antrenarea modelului lingvistic:

*./Prelucrarea datelor/lm-gen/buildCorpus.py*

*./Prelucrarea datelor/lm-gen/wikiParse.py*

*./Prelucrarea datelor/lm-gen/httpDownloader.py*

*./Prelucrarea datelor/lm-gen/mkvocab.py*

*./Prelucrarea datelor/lm-gen/constants.py*

*./Prelucrarea datelor/lm-gen/wikiParse\_alternative/wikiParseAleckej.java*

1.1. Explicatii:

1.1.0. Conversie audio:base URL for the Wikipedia pages

*./Prelucrarea datelor/stripAudio.py:*

Acest fisier incapsuleaza functionalitatea modulului responsabila cu conversia fisierelor audio la format audio RAW sau WAV de sample rate 16Khz, 16-bits per frame si un singur canal audio.

Functia suporta formatele MP3, WAV si RAW.

Functionalitatea mentionata este expusa prin functia:

*def convertFileToRaw(inFilePath, outFilePath)*

*inFilePath* este calea catre fisierul audio ce se vrea convertit iar *outFilePath* este calea catre rezultatul convertirii. Daca la calea *outFilePath* exista deja un fisier, acesta va fi inlocuit.

*outFilePath* (si *inFilePath*) contin extensia fisierului.

Aceasta functie poate fi apelata direct de la linia de comanda prin rularea scriptului: primul si al doilea argument vor reprezenta *inFilePath* si *outFilePath* respectiv.

*./Prelucrarea datelor/stripAudio\_pydub.py:*

O versiune, veche a a scriptului stripAudio.py. Aceasta versiune nu este folosita actual insa expune aceeasi functionalitate ca si stripAudio.py. Diferenta intre cele doua este libraria de manipulare audio folosita.

1.1.1 Parsarea dictionarului de foneme:

*./Prelucrarea datelor/parseDictionary.py*

Acest script primeste ca parametru o cale catre un fisier. Se incearca citirea fisierului dat, iar in cazul in care nu apare eroare se citeste fiecare linie, se normalizeaza(se elimina spatiile) apoi se asociaza fiecarui cuvant reprezentarea in foneme. Acest script contine urmatoarele functii:

*def domain (tokens)*

Aceasta functie preia primul element din tokens care reprezinta cuvantul si il returneaza.

*def image (tokens)*

Aceasta functie preia al doilea element din tokens care reprezinta scrierea in foneme si il returneaza.

*def normalizeLine (line)*

Aceasta functie primeste ca parametru o linie din dictionar si returneaza linia in care toate spatiile sunt inlocuite cu un singur caracter whitespace.

*def parseDictionaryExplicit (path, domain, image)*

Aceasta functie construieste dictionarul, fiecarui cuvant i se asociaza reprezentarea acestuia in foneme. In cazul in care calea nu este gasita se va afisa un mesaj de eroare.

*def parse\_dictionary (path)*

*Aceasta functie primeste ca parametru o cale catre un fisier si returneaza dictionarul in care cheia este cuvantul iar valoarea este reprezentarea in foneme.*

*def parse\_dictionary\_reverse (path)*

*Aceasta functie primeste ca parametru o cale catre un fiseier si returneaza dictionarul in care cheia este reprezentarea in foneme iar valoarea este cuvantul.*

1.1.2. Antrenarea modelului lingvistic:

*./Prelucrarea datelor/lm-gen/buildCorpus.py:*

Acest script, odata apelat, incepe construirea unui corpus in limba romana folosind baza de date a wikipedia. Scriptul atinge acest rezultat utilizand functii prezente celelalte module Python prezente in director:

*httpDownloader* pentru descarcarea bazei de date wikipedia

*wikiParse* pentru eliminarea metadatelor si generarea corpusului

*mkvocab* pentru generarea unui vocabular pe baza corpusului (folosit in unele metode de creare a modelului lingvistic)

*./Prelucrarea datelor/lm-gen/wikiParse.py*

Scriptul elimina elementele redundante si extrage textul dintr-un fisier xml cu informatii de pe Wikipedia.

Resursa folosita ca model: <https://github.com/dbpedia/fact-extractor/blob/master/lib/WikiExtractor.py>, modificat astfel incat sa parseze limba romana (diacriticele), sa elimine orice caracter care nu este litera, cifra sau spatiu si sa transforme numerele din reprezentarea lor in baza 10 in string.

Input: Document xml ce poate fi descarcat de pe: <https://dumps.wikimedia.org/rowiki/> . Exemplu de apelare: pathToPython wiki.py --infn path\rowiki-latest-pages-articles.xml .

Output: Un text document numit wiki.txt.

Optiuni:

--compress: arhiveaza fisierul de output folosind bzip

--infn: pentru a specifica fisierul de input

-h, --help: afiseaza optiunile ce pot fi folosite

*./Prelucrarea datelor/lm-gen/wikiParse\_alternative/wikiParseAleckej.java:*

Aplicatie alternativa (fata de wikiParse.py) pentru parsarea si eliminarea metadatelor din baza de date Wikipedia descarcata. Actual, acest program e nefolosit.

*./Prelucrarea datelor/lm-gen/httpDownloader.py*

Acesta este un simplu script folosit la descarcarea bazei de date wikipedia. El contine o singura functie:

*def downloadHTTP(url)*

Aceasta functie primeste ca parametru un URL iar odata apelata, creeaza in folderul curent un fisier identic cu cel aflat la URL-ul dat.

*./Prelucrarea datelor/lm-gen/mkvocab.py*

Acest script expune functia *createVocabulary* care, primind prin *inFilePath* o cale catre un fisier text ce contine cuvinte separate prin spatiu, construieste la *outFilePath* un fisier text cu numarul de aparitii a fiecarui cuvant gasit in *inFilePath*.

Fisierul este sortat descrescator dupa numarul de aparitii si are forma:

*<cuvant1> <aparitii1>*

*<cuvant2> <aparitii2>*

*<cuvant3> <aparitii3>*

*./Prelucrarea datelor/lm-gen/constants.py*

Un simplu fisier folosit pentru stocarea constantelor. Singurul uz este stocarea URL-ului la baza de date wikipedia.

2. Librarii, executabile, dependente externe:

2.0. sox:

Sox este o librarie pentru manipulat fisiere si semnale audio. Pentru instalare trebuie sa descarcati ultimul installer de la linkul:

<https://sourceforge.net/projects/sox/files/sox/?fbclid=IwAR0fcXqzqot4RV5AcAwrZTToIa_J_MmGTYmUqaKcTmgqNJJfSLJtGUh6XCI>

Dupa descarcare, acesta trebuie rulat. Urmati pasii din installer. La finalizarea instalarii, e recomandat ca variabilei de sistem PATH sa ii fie adaugata calea catre executabilele sox.

Pentru suport MP3, trebuie sa descarcati urmatoarele doua librarii:

<https://app.box.com/s/tzn5ohyh90viedu3u90w2l2pmp2bl41t?fbclid=IwAR3evraKs67KSFdDro_gybuALUlN_IL2JWe5eMHhnyuiRoP-V1QX98sZd3k>

Acestea trebuie puse in directorul cu executabilele sox.

2.1. SRI LM:

SRI LM este un toolkit pentru construit si aplicat modele lingvistice statistice, in principiu pentru uz in recunoasterea vocii, tagging statistic si segementare, si traducere automata. Adresa oficiala este:

<http://www.speech.sri.com/projects/srilm/>

SRI LM nu este o dependenta. Proiectul poate rula deoarece acesta include un model lingvistic pre-generat. Insa daca se doreste experimentarea/testarea crearii de modele lingvistice, sursele acestuia pot fi descarcate aici:

<http://www.speech.sri.com/projects/srilm/download.html>

Comanda utilizata in construirea modelului lingvistic inclus este:

*ngram-count -kndiscount -interpolate -order 9 -text wiki.txt -lm output.lm*

Aceasta comanda va genera un model lingvistic statistic 9-gram in format ARPA.

Note: Pentru crearea corpusului (wiki.txt) se va rula scriptul *buildCorpus.py*.

Formatul pe disc a acestui model este:

*\data\*

*ngram 1=n1*

*ngram 2=n2*

*...*

*ngram N=nN*

*\1-grams:*

*p w*

*...*

*\2-grams:*

*p w1 w2*

*...*

*\N-grams:*

*p w1 ... wN*

*...*

*\end\*

Formatul ARPA pentru modele N-gram cu backoff incepe cu un header, introdus de keywordul \data\, care expune numarul de N-grame de fiecare lungime. Urmand aceasta, N-gramele sunt printate cate una pe linie, grupate pe sectiuni dupa lungime, fiecare sectiune incepand cu keywordul \N-gram:, unde N este lungimea N-gramelor ce urmeaza. Fiecare linie de N-gram incepe cu logaritmul (baza 10) a probabilitatii conditionale p a N-gramului respectiv, urmat de cuvintele w1...wN ce compun N-grama. Keywordul \end\ conclude reprezentarea modelului.

Ponderile de backoff sunt necesare doar pentru acele N-grame care formeaza un prefix pentru un N-gram mai lung in acel model. In particular, N-gramele de cel mai inalt ordin ar fi fara folos.

Din moment ce log(0) (minus infinitate) nu are o reprezentare portabila, astfel de valori sunt mapate la un numar negativ foarte mic dar aceasta valoare este (-99 in SRILM) este reprezentata ca log(0) cand este citita inapoi in memorie.

2.2. pydub:

Pydub este o librarie pentru manipulat fisiere si semnale audio folosita in primele versiuni ale *stripAudio.py.* Curent, este inlocuit cu sox si nu reprezinta o dependenta. Insa daca se doreste utilizarea *stripAudio\_pydub.py* in locul *stripAudio.py*, instalarea pydub se face ruland de la linia de comanda:

*pip install pydub*