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*

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*

1.1. Explicatii:

1.1.0. Conversie audio:

*./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.

1.1.1 Parsarea dictionarului de foneme:

*./Prelucrarea datelor/parseDictionary.py*

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*

*./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.