* Entender el caso

Descarrego y descomprimeixo els fitxers adjunts, llegeixo la tasca i la entenc be i obro els fichers de dades per donar una ullada rápida a les dades i tenir una idea básica de com son.

15min

* Preparar labeler

Entenc que en algún momento haure de clasificar els casos, aixi que començo a preparar un script de Python amb tkinter, de nivel basic, que em mostri per pantalla el text que conte i tingui 3 botons per decidir la reposta a la pregunta: Is the customer asking for the status of his order? Unper dir que si, un per dir que no, i un per dir que no ho tinc clar i mirarlo posteriorment. Que finalment guardi la clasificacio donada.

1:30h

* Analisis de les dades

Decideixo anar amb google collab perque estic bastant acostumat i crec que queda be a la hora de comentar el text i separar en seccions, que despres pots amagar, pots tenir les dades pujades a github i no depens de instalacio de la maquina ni res. Aixi ajuda la replicabilitat.

Començo a crear el google collab, paso les dades a pandas y miro quants missatges tinc, quants camps de informacio.

La primera idea que tinc, tenint en compte que estic mes accostumat a tractar amb textos i Deep learning, es que probablement haure de mirar dentrenar un model de classificacio amb el text com a input. I en aquest cas, 1433 missatges poden ser justets per entrenar el model.

Nomes al mirarles visualment, ja m’adono que hi ha alguns camps que sembla que sempre tenen el mateix valor. Ho comprovo en codi i decideixo eliminar tots els camps que nomes tenen un valor, ja que no donen cap tipus de informacio sobre el tipus de missatge que es, aixi em quedo amb els camps de “Description”, “tags” i “CreatedAt”. Tenint en compte que aquest ultim deu ser autodescriptiu, aquest camp no dona informacio sobre el contingut del missatge aixi que tambe l’elimino. Penso que podre pasar paraules del text i els tags com a input del model.

20min

* Separar tags

Pero els tags consisteixen en un un array de tags de llargada variable, aixi que decideixo pasar a tenir tots els arrays una llargada uniforme mantenint la máxima que tenim, en aquest cas de 5, i omplint els arrays de menor longitud amb valors None. Despres de aixo separar cada tag del array en una columna del dataframe diferent mantenint el ordre, per si es important a la hora de clasificar.

40min

* Limpiar texto

Degut a que de primeres vaig pensar que entrenar un model de llenguatge, probablement amb una neural network, mes especificament un transformer, seria el procediment a seguir, en aquest punt descriure el que vaig estar fent.

Primer de tot vaig dedicarme a netejar el text de majuscules, punctuacio, números, simbols extranys. Tokenitzaar el text, lemmejar les paraules. Despres de una mica de bug fixing. Finalment analitzo la llargada dels missatges per normalitzarlos a una lllargada de 370 afegint Nan elements.

1:30h

* Investigation

Netejar el text de input es una part molt important en el cas del transformer, pero tot i aixi em ronda la idea que el numero de missatges no sera suficient, aixi que investigo una mica sobre data augmentation en aquest cas.

Durant aixo tambe m’adono que un punt molt important sera l’spelling. Una paraula mal escrita pot afectar la tokenitzacio i la subsequent identificacio de la paraula. Aixi que dedico una part important de temps a investigar sobre possibilitats de Spelling corrector i quina seria la millor opcio. Investigant i llegint algun paper com el de

<https://aclanthology.org/2020.lrec-1.228.pdf>

Em decideixo per LanguageTools, perque sembla que es capaç de tractar ambdos casos de errors per paraules unides i per concatenacio i sembla senzill de fer servir.

Penso que la millor manera seria que, un cop tingui el diccionari de aparicions de paraules en el meu training text, si una paraula no apareix en aquest pasarla primer pel corrector, i mirar si la detecta com a error, i si es el cas corregirla llavors, sino donarla ccom a descconguda.

1:15h

* Training and test

Abans de posarme a aplicar el corrector, per tal de canviar una mica de aires, deccideixo separar el conjunt en training i test. Em deccideixo per una distribucio 75%/25% i seleccionant les frases de forma random, ja que nose si hi ha alguna distribucio interna que desconec i m’interesa evitar.

10m

* Tags to onehot

Vull pasar els tags a onehot enccoding per tal de normalitzar la distanncia entre ells. Penso en com combinarlos ja que pasarlos al input de forma consecutiva podria suposar en un augment important de la llargada del input. Aixi que penso que el millor seria sumarlos en comptes de enganxarlos un darrere el altres.

Tambe decideixo que els tags que apareguin una unica vegada es combinaran en el valor Unknown, aixi no es consideraran per separat en lentrenament, ja que son casos masa especifics i poden afectar a un pitjor resultat i generalitzacio.

Aqui hi ha un canvi important ja que m’adono de que en comptes de aplicar un process complicat de cclassificacio de text, puc mirar de utilitzar els tags com a input de clasificacio i que funcioni correctament

2h

* Sample

Per tal de entrenar models de clasificacio decideixo reduir el numero de mostres a la mitat de forma random per tal de reduir el temps de labelear.

5min

* Labeling

Acabo de arreglar el programa de tkinter de llabeling perque s’adapti a com tinc les dades guardades actualment i em poso a labelearles.

A més, en el cas que aixi de tornar a la hipotesis que necesito clasificar segons el text, aixo em donara informacio visualment sobre casos tipics que podria netejar del text.

Em fixo que hi ha algun cas on es una mica ambigu la classificacio, pero intento fero lu millor possible

2:15h

* Separar training i eval

Separo training i eval en 80/20, per tal de evaluar els propers models, quedanmte una distribucio final de 60% train / 15% eval / 20% test.

5m

* The models

Aqui penso quines possibilitats tinc de models per entrenar, per probar que tal funcionen i quedarme amb la millor al final. Em deccideixo per entrenar un SVM, un RandomForest i una Neural Network. Intuitivament crec que la que funcionara millor sera el SVM ja que normalment necesita menys dades que altres models per obtenir bons resultats i les clases estan ben delimitades. Tambe penso que la neural network probablement no obtindra gaire bons resultats ja que disposem de poques dades, a mes havent fet el sampling.

Mentres preparo la svm penso que al tenir poques dades de entrenament, hauria de aplicar kfolding en lentrenament, aixi que ho faig. Preparo el svm i el random forest

1:10h

* The situation

Aqui em dedico un moment a definir clarament la situacio, i quina seria la millor metrica per evaluar.

La situacio que jo tinc al cap es que hi ha el classificador que classifica si es o no un missatge del tipus “where is my order”, i que en el cas que no ho sigui s'enviara a un altre classificador (probablement en aquest cas huma) que dira a on ha danar aquest missatge. Amb aixo tinc clar que els errors critics es que classifiqui un cas 0 com a 1, ja que en aquest, com se ma demanat que classifiqui nomes aquest cas, faig la suposicio que es perque despres hi ha un proces automatic que pot donar problemes si la clasificacio es incorrecte, en canvia la inversa si un 1 es classifica com a 0, el huma que hi hagi reclassifcantlos el podra reclassificar com a “where is my order” i enviar al process adecuat.

Amb aixo defineixo que la metrica mes important es la precisio, ja que valora el numero de correctes reals dintre dels que ha donat com a correctes el model. Per evitar casos on el models directament classifiqui com a 0 tots els casos, els entrenare amb accuracy i la tindre en compte com a segon metrica en la evaluacio.

20m

* The models(2)

Definida la situacio decideixo posar un threshold de confiança a la sortida dels models. Si la confiança no es superior a un valor que haure de definir, la prediccio es donara com a invalida i s’enviara a 0 (classificador huma).

30h

Segueixo implementant en random forest i la NN, aquesta ultima basada en el model en <https://machinelearningmastery.com/building-a-binary-classification-model-in-pytorch/>

Aplico una seed a tots els models per tal de poder tenir sempre els mateixos resultats si els entrenes again.

1:10h

* Evaluation data

Faig servir les dades de evaluacio per comprovar que els models funcionin correctament i per deccidir quin nivell de threshold de confiança seria el millor.

Les dades les mostro com una matriu de confiança per tal que siguin mes entendibles visualment.

M’adono que els tres models en el seu millor threshold obtenen el mateix resultats. Em miro quines frases son aquestes en les que falla, per si fos cosa que potser no estan del tot be lebelejades segons el text, pero tot i que siguin uns casos una mica ambigus crec que els casos estan ben labelejats.

25min

* Some minor problems

Per tal de mantenir la independencia entre les seccions, i poder carregar els models en el futur per evaluar el conjunt de test intento descarregar els models i ser capaç de carregarlos again, i el mateix pel labelBinarizer fet servir per fer el onehot enccoding als tags. Aixo resulta donarma una mica mes de problemes i per aixo li dedico un temps.

1h

* Test data

Em dedico a netejar les dades de test, per tal de tenirles en el mateix format que les de entrenament. Les paso pels models i observo els resultats. En el cas de test svm i random forest segueixen donant els amteixos resultats, pero la neural network ja no hi ha empitjjorat. Sobretot em fixo en que ens trobem amb un cas critic i l’examino. Segons el text sembla que esta ben lebelejat, tot i que si em fixo que en els labels que te apareix el “where-is-my-order” que suposo que normalment defineix casos positius, aixi que considero que es un error a la hora de quan han afegit els tags.

He provat de fer visualitzacio del les decisions del model, pero per tal de finalitzar el analisis amb un temps menor i no haver de implementar pca i similars, he decidit no fer-ho. Pero si disposesim de mes temps es podria fer.

45min

* Best model

Finalment selecciono el millor model i em deicideixo pel SVM. Tant el SVM com el Random Forest obtenen els mateixos resultats, aixi que el temps de processament es un valor a tenir molt en compte. SVM triga 0.008s a fer els 179 casos de test, mentre que RF triga 0.0224s, cosa que no es molt significativa amb aquest numeros de casos. Pero si volguessim aplicar-ho en una situacio real on podrien arribar molts mes casos aquest temps pot suposar un increment significatiu.

15min

* Extras

Com a extra, comprovo el codi i l’acabo de netejar. I escric aquest document

2h