NLP per a la resolució de tiquets

Irina Moreno Lahoz

Resum– Resum del projecte, màxim 10 línies. ....

Paraules clau– tiquets, NLP, Machine Learning, Inteligència Artificial, solució

Abstract– Versió en anglès del resum . ...

Keywords– tickets, NLP, Machine Learning, Artificial Intelligence, solution

• E-mail de contacte: irina.moreno@autonoma.cat
• Treball tutoritzat per: Débora Gil Resina (Ciències de la Computació)
• Curs 2022/23

1 INTRODUCCIÓ

AQUEST treball s'ha desenvolupat en el departament d'*IT* de l'empresa SEAT S.A., on es dediquen a la producció i la fabricació d'automòbils. En aquest departament, una de les tasques que tracten són les incidències que tenen els treballadors. Emmagatzemen dades sobre la informació de la incidència en format de tiquets en una base de dades. Aquestes incidències són una part molt important en el rendiment de l'empresa, ja que tenen un gran impacte en la producció i en el treball de l'empresa.

Es vol realitzar una ajuda a l'empresa amb la gestió de la solució d'aquests tiquets d'incidències.

1.1 Objectius

L'objectiu principal del projecte és poder donar una solució o una aproximació a la solució a la incidència que té l'usuari, sent una capa intermèdia entre l'usuari i la solució, i així, poder evitar acudir al servei de suport de tècnics propi de SEAT S.A. o minimitzar aquest accés.

D'una banda, es vol arribar a una solució molt més ràpida perquè l'usuari segueixi amb el seu treball amb total normalitat. I, d'altra banda, es vol minimitzar costos en els contractes de l'ajuda oferta pel servei de suport.

El treball se centrarà en l'estudi de mètodes de Processament del Llenguatge Natural (*NLP*, per les seves sigles en anglès) per trobar bones solucions a les incidències, i després es plantejarà com implementar aquest estudi als programes i processos de l'empresa i que passi a ser una bona ajuda, ja que pot pot tenir impacte económic sobre els contractes de servei i en el temps de desenvolupament i fabricació.

1.2 Planificació

Aquest projecte estarà dividit en un seguit de fases.

En primer lloc, es farà un estat de l'art sobre el món de NLP. Es buscaran els mètodes i les eines que existeixen i quines poden ser aplicades per arribar a l'objectiu d'aquest projecte.

En segon lloc, la primera cosa important a tenir en compte com a començament d'aquest projecte és entendre les dades amb els quals es tractarà i saber quina és la millor manera de tractar-les. Hi ha un volum de dades molt elevat. Es realitzaran una sèrie de filtres per eliminar aquelles dades que no són útils pel nostre mètode o aquelles que es troben lluny d'aconseguir el nostre objectiu. També es procedirà a dur a terme subconjunts de dades depenent de la relació que tinguin les solucions i per poder ajudar al mètode a trobar millors solucions.

Com a tercer punt, s'estudiaran tots els models, tècniques i eines que es poden fer servir per arribar a una bona solució. S'analitzaran i es triaran els que puguin donar millors resultats.

Per continuar, s'aplicarà a les dades el mètode triat.

A partir d'aquests resultats, també podem saltar a les fases anteriors i fer canvis en la preparació de les dades o buscar mètodes millors. Són 3 fases que es complementen.

Arribant al final del projecte, s'estudiarà la possibilitat de realitzar una nova interfície en l'aplicació que l'empresa utilitza en la generació dels nous tiquets. Es vol parlar amb caps i persones creadores de l'aplicació per a piular codi, crear i connectar tot el necessari.

Com a part final, es crearan les conclusions finals del projecte. Explicacions acompanyades de *dashboards* de l'anàlisis dels resultats.

Es treballarà en un entorn de Jupyter amb el llenguatge de Python.

A continuació, s'introdueix una taula que especifica el temps de dedicació a cada part d'aquest projecte.

Fases	Temps a dedicar
Estat de l'art	2 dies
Gestió de les dades	5 setmanes
Disseny experimental	4 setmanes
Estudi d'integració	2 setmanes
Conclusions	1 setmana

TAULA 1: TAULA DE PLANIFICACIÓ

2 PREPARACIÓ DE LES DADES

L'empresa SEAT S.A. té una gran quantitat de dades emmagatzemades, dades de tota mena. En aquest projecte es tractarà amb el dataset on es guarda tota la informació sobre els tiquets d'incidències. Aquest dataset s'actualitza cada dia i va creixent de mida a mesura que va passant el temps, al dia es poden arribar a generar 800 nous tiquets. El tiquet més antic que hi ha emmagatzemat és de la data 26/08/2015 i el dataset té una mida de 858400 tiquets aproximadament (primera setmana que es treballa amb la base de dades).

Primerament, s'eliminen aquells registres rellevants del dataset que estan buits, com per exemple algun tiquet que no tingui solució o que no tingui *CI*. Visualitzem per cada *CI* les quantitats de tiquets que hi ha en un histograma mostrat en la figura 1.

El conjunt de dades consta d'un número molt elevat de tiquets. Cada tiquet conté molts camps emmagatzemats, però ens quedem amb els següents, ja que són els que tenen més rellevància:

- Ticket Type: si es tracta d'un incident o d'una sol·licitud.
- CI[1] ID: combinació única per a la identificació dels tiquets.
- CI Name: nom de l'aplicació on es troba l'incident.
- Title: títol que l'usuari li dona a l'incident.
- Description: descripció que l'usuari realitza en descriure el seu problema.

1 Les sigles CI (*Configuration Item*) provenen de la llibreria ITIL (*Information Technology Infrastructure Library*), un marc de millors pràctiques per a la gestió de serveis de tecnologia de la informació. Un element CI correspon a qualsevol component o part d'una infraestructura de tecnologia de la informació que ha de ser gestionada i controlada per a garantir la prestació de serveis d'IT d'alta qualitat.

- Solution: es descriu la solució al problema de l'usuari.
- Open Time: data i hora la qual el tiquet ha estat creat.
- Resolved Time: data i hora la qual el tiquet ha estat solucionat.
- Current Status: informació sobre el que s'esta fent en aquell moment amb el tiquet.
- Current Assignment: grup al qual ha estat assignat aquest incident.
- Prio: camp numèric que indica la rapidesa en què la incidència s'ha de solucionar.
- Imapct: camp numèric que indica l'impacte de la incidència dins del treball en SEAT.

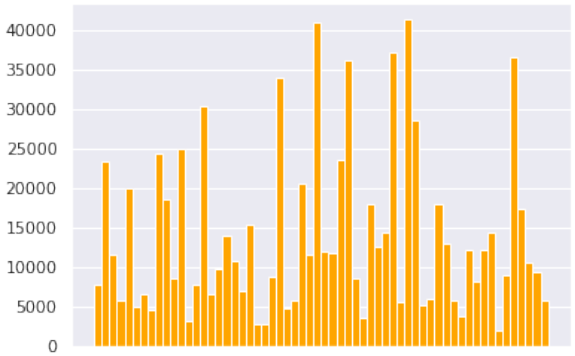

Fig. 1: Histograma CIs

Com podem veure en el gràfic anterior, hi ha una gran diferència en els volums de dades dels diferents *CI*. Estudiant amb més profunditat les dades, s'eliminen els *CI* que contenen menys de 6 registres, ja que poden provocar soroll en el nostre model d'aprenentatge perquè no tenen suficients solucions amb les quals treballar i pot provocar no arribar a solució.

En segon lloc, ens hem de fixar amb el que realment treballarem, la solució dels tiquets. L'atribut 'Solució' és un camp que pot ser generat directament per una persona i també pot ser que una part es crei automàticament. Per aquesta raó, procedim a fer una neteja i eliminarem les parts generades automàticament i, posteriorment, eliminarem aquelles solucions on el text sigui massa curt. S'aplica el filtratge on només seleccionem les solucions les quals la longitud del text supera els 29 caràcters. Amb aquest filtratge, la base de dades s'ha reduït de mida uns 75000 tiquets, aproximadament.

Quan ja tenim tota la base de dades preparada, es procedirà a realitzar diferents datasets amb els quals es treballarà i es veurà quin dona millors resultats.

Primerament, es mira de separar les dades depenent del seu *CI*, ja que pot ser que existeixin diferents aplicacions que tinguin un mateix problema, però la solució a cada problema sigui totalment diferent. A més a més, es mirarà la quantitat de tiquets que es generen a partir de cada *CI*, pel fet que pot ser que no hi hagi suficients incidents d'aquell *CI*. No s'ha de crear, modificar ni eliminar cap camp al dataset que tenim, simplement considerarem el nom de cada *CI* en el paràmetre per fer l'agrupació.

Com següent punt, es procedirà a fer agrupacions basantse en una variable anomenada *Service Tower*, que son unes categories on es classifiquen les diferents aplicacions que estan relacionades o pertanyen al mateix departament per a fer anàlisis en conjunt. Fent aquesta agrupació, es vol aconseguir tenir en compte en l'aprenentatge tots els *CI*. Es crea un dataset nou, idèntic al principal, però amb un camp extra anomenat 'Service Tower'.

Aquesta informació es troba en un altre dataset, així que no hi ha la mateixa cantitat de dades. S'ha afegit una categoria extra on es troben emmagatzemats aquells *CI* que no han pogut ser classificats. En el gràfic de barres de la figura 2 mostrem la canitat de diferents *CI* que es troben en aquestes categories, en aquest cas uns 913 *CI*.

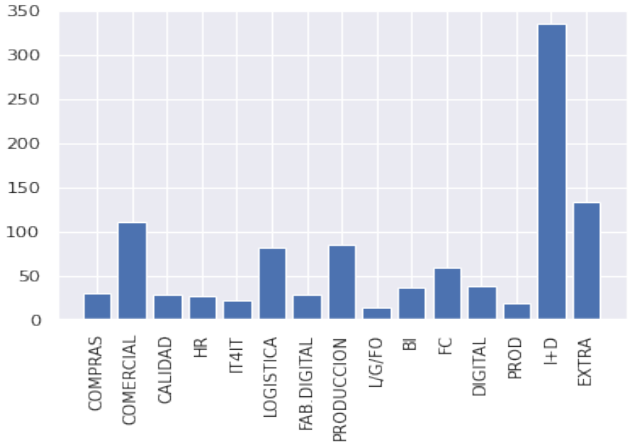

Fig. 2: CI en Service Towers

Amb la creació d'aquests 2 datasets passarem a la part de metodologia, on buscarem quins són els millors mètodes i les millors tècniques per posar en pràctica en el nostre conjunt de dades.

3 METODOLOGIA

En quest projecte, la Intel·ligència Artificial té gran importància. La *IA* és una branca de la informàtica que busca crear màquines que imitin la intel·ligència humana per a fer tasques i puguin millorar conforme la informació que recopilen. La màquina rep dades, les processa i respon a elles.

Per a això, la *IA* empra una àmplia varietat de tècniques: aprenentatge automàtic, el processament del llenguatge natural, la visió artificial i l'aprenentatge profund.

Ens centrarem en el Processament del Llenguatge Natural (*NLP*), que és una àrea d'estudi centrada en com els ordinadors entenen el llenguatge humà, l'interpreten i processen. El *NLP* és realment la interfície entre la ciència informàtica i la lingüística. Per tant, es basa en la capacitat de la màquina per a interactuar directament amb els humans.

3.1 Aplicacions, Tècniques i Eines de NLP

NLP és utilitzat per moltes aplicacions com assistents virtuals (*chatbots*), traducció automàtica de textos, recuperació d'informació, classificació de textos, detecció de sentiments, resum de textos, reconeixement d'entitats, anàlisi de veu, etc.

Entre totes aquestes possibles aplicacions, el treball se centrarà en la classificació de textos i recuperació d'informació, ja que a partir d'un text, haurà de detectar les paraules més rellevants i, finalment, acabarà classificant aquell text en una solució o una altra.

NLP utilitza una sèrie de tècniques que s'utilitzen per a processar, analitzar i comprendre el llenguatge humà mitjançant l'ús d'algorismes i models computacionals.

Les tècniques més populars, i en les que el treball es centrarà, són:

- Tokenització: procés de dividir un text en unitats més petites anomenades *tokens*. Els *tokens* poden ser paraules, signes de puntuació o números.
- Eliminar *StopWords*: les *StopWords* són paraules que s'eliminen del text per a reduir la quantitat de paraules amb les quals es treballa i centrar-se en les paraules clau o significatives del text.
- Eliminar signes de puntuació: eliminem els signes de puntuació perquè no són rellevants i així ens centrem en les paraules.
- *Stemming*: procés de reduir les paraules a la seva forma base, com ara convertir 'corriendo' en 'corre'. És una tasca important per a la normalització de text.
- Lematització: és similar al stemming, però en lloc de reduir les paraules a la seva forma base, les converteix a la seva forma de diccionari, com ara convertir 'corriendo' en 'correr'.
- Clustering de text: És una tasca d'aprenentatge no supervisat que consisteix a agrupar els documents de text en grups basats en la seva similitud.

Aquestes tècniques són les que s'utilitzaran en aquest treball, però existeixen moltes més, com per exemple: Anàlisi morfològica, *Tagging part of speech* (PoS), *Bag of Words*, Anàlisi de sentiments, Classificació de text, entre d'altres.

Calen eines per poder posar en pràctica totes les tècniques que utilitza NLP per a les diferents aplicacions. Les biblioteques de NLP són conjunts de recursos de programari i dades lingüístiques reconstruïts que permeten als desenvolupadors de programari crear aplicacions de processament automàtic del llenguatge natural.

Les biblioteques de NLP més populars inclouen NLTK, spaCy, Gensim, TextBlob, Stanford CoreNLP i el Natural Language Toolkit de Google, entre altres. Aquestes biblioteques solen ser de codi obert i es poden fer servir de manera gratuïta, encara que pot haver-hi restriccions en relació amb el seu ús comercial.

3.2 Models de NLP

La llibreria scikit-learn (sklearn) és una de les més populars per a l'aprenentatge automàtic en Python, i també és àmpliament utilitzada en NLP.

A continuació s'expliquen alguns dels mètodes de la llibreria sklearn que s'utilitzaran per determinar quin és el millor a aplicar en aquest treball:

- MultinomialNB de Naive Bayes: és un model probabilístic que s'utilitza per a la classificació de text i altres tasques de NLP.
- LogisticRegression de Linear Model: és un model de classificació lineal que s'utilitza per a la classificació de text i altres tasques de NLP.
- RandomForestClassifier de Ensemble: és un model d'aprenentatge de conjunt que s'utilitza per a la classificació de text i altres tasques de NLP.
- LinearSVC de SVM: és un model d'aprenentatge supervisat que s'utilitza per a la classificació de text i altres tasques de NLP.
- CountVectorizer de Feature Extraction: és una tècnica d'extracció de característiques que converteix un conjunt de documents de text en una matriu de termes de document. És útil per a la construcció de models de classificació basats en text.
- TfidfVectorizer de Feature Extraction: és una tècnica d'extracció de característiques que converteix un conjunt de documents de text en una matriu de característiques ponderades segons la freqüència inversa de document. És útil per a la identificació de paraules clau importants en un conjunt de documents.

Els models que s'utilitzaran en aquest treball són els següents, encara que existeixen molts més, com per exemple LabelEncoder, OneHotEncoder, TfidfTransformer, entre d'altres.

4 Disseny Experimental

En aquesta fase es comença a posar en pràctica les tècniques explicades anteriorment perquè la màquina processi el llenguatge humà i es faran les prediccions de les solucions a partir de diferents models.

4.1 Preparació del text

Primerament, es comença fent una neteja i una adaptació en els diferents textos que tenim a tractar, així la màquina el processa i l'entén.

Iniciem aplicant el mètode de tokenitzar important el mòdul 'word_tokenize' de la llibreria 'NLTK'. El que fa aquest mètode exactament és separar el text paraula per paraula i ho emmagatzema en una llista.

Seguidament, eliminem els signes de puntuació de cada element d'aquesta llista. Es mira caràcter a caràcter i, si es troba en una llista anomenada 'punctuation' carregada de la llibreria 'string', vol dir que és un signe de puntuació i s'ha d'eliminar.

Com a següent pas, eliminem les *StopWords*. Importem una llista de paraules sense gran valor, com per exemple 'el','la', 'de', 'en', 'por', 'con', 'para', entre d'altres; del mètode 'corpus' de la llibreria 'NLTK' i busquem i eliminem si alguna d'aquestes paraules es troba en el text de 'Title' i 'Description'. A aquesta llista afegirem manualment una sèrie de paraules que considerem de valor nul i podrien

estar presents en el text, com per exemple 'gracias', 'martorell', 'saludos' i 'hola'.

Afegirem un nou atribut al nostre dataset que serà 'Text', on es trobarà emmagatzemada una llista de paraules que són la unió de l'atribut 'Title' i 'Description'.

A continuació es mostra una comparació de 2 textos. El primer, figura 3, és el text original que conté la base de dades, i el segon, figura 4, és el mateix text, però se li ha aplicat els mètodes anteriors per poder treballar millor amb ell.

Reset password yes; Yes - ACCESO PORTAL
Buenos días,
Necesitaría que me hicierais un reset del password del yes, creo que se me ha bloqueado la cuenta.
Gracias!

Fig. 3: Text abans de la neteja

['Reset', 'password', 'yes', 'Yes', 'ACCESO', 'PORTAL', '—', 'Buenos', 'días', 'Necesitaría', 'hicierais', 'reset', 'password', 'yes', 'creo', 'bloqueado', 'cuenta']

Fig. 4: Text després de la neteja

Per acabar, treballarem amb la llibreria 'spacy', que tracta de normalitzar paraules a la seva forma bàsica o arrel. S'inicialitza carregant el model *spaCy* per l'idioma espanyol i s'extreuen les paraules en forma de diccionari. El resultat acaba sent la mateixa llista que ja teniem, però amb les paraules actualitzades, com podem veure en la figura 5.

TEXT EXEMPLE SPACY

Fig. 5: Text després d'aplicar mètode spaCy

4.2 Comparació dels Models

4.3 Predicció de la solució

5 INTEGRACIÓ

App Service Desk SEAT és una aplicació creada pel departament on l'usuari emplena una sèrie de camps i crea un tiquet d'incidències. En aquest moment són enviats al servei de suport per a la seva solució i és guardat a la base de dades.

Implementacions per dur a terme: abans de donar-li una possible solució, es faran una sèrie de preguntes (arbre de preguntes) a l'usuari que ajudin el model a augmentar l'èxit (percentatge de probabilitat final) de trobar la solució, detallar un llistat d'accions per a ajudar al servei de suport a acostar-se i arribar abans a la solució, crear el script que en executar-se solucioni el problema de l'usuari.

6 CONCLUSIONS

AGRAÏMENTS

REFERÈNCIES

[1] http://en.wikibooks.org/wiki/LaTeX

[2] https://www.holded.com/es/blog/aplicaciones-inteligencia-artificial-negocios

[3] https://es.wikipedia.org/wiki/Inteligencia_artificial

[4] https://www.unir.net/marketing-comunicacion/revista/nlp-procesamiento-lenguage-natural/

[5] https://www.aprendemachinelearning.com/procesamiento-del-lenguaje-natural-nlp/

[6] https://datascientest.com/es/nlp-natural-language-processing-introduccion

[7] https://chat.openai.com/chat

APÈNDIX

A.1 Secció d'Apèndix

... .

A.2 Secció d'Apèndix

... .