UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Universitat Politècnica de Catalunya
Facultat d'informàtica de Barcelona

Grau en Enginyeria informàtica
Especialitat en enginyeria del software

Twitter Sentiment Analysis

Memòria

28 juny 2021

Ariadna de Arriba Serra

Director
Xavier Franch Gutiérrez

Co-director
Marc Oriol Hilari

Tutor
Fernando Barrabes Naval

Abstract

Sentiment analysis is a machine learning technique used to classify the level of sentiment from a text. It could be from the simplest classification of a message according to whether it expresses a positive or negative feeling, to the most complex that it is able to classify text into different emotions. It is a common technique that is used more and more in many companies and sectors, to see people's reaction in front of a new product or service and react to the predictions to offer what the client wants.

In this project, sentiment analysis will be applied for the classification of messages on social networks, a very common technique as it is where people express more their feelings without filters, related to Covid-19 and the pandemic we are living in. The aim is to conduct a study to see the emotional tendency of the population in this context.

Keywords: machine learning, emotion classification, sentiment analysis, natural language processing, social networks

Resum

El *sentiment analysis* és una tècnica de *machine learning* utilitzada per a classificar el grau de sentiment d'un text. Existeix des de la més simple classificació d'un missatge segons si expressa un sentiment positiu o negatiu, fins a la més complexa que és capaç de classificar en diferents emocions. És una tècnica que cada vegada és més utilitzada en moltes empreses i sectors, per veure la reacció de la població davant d'un nou producte o servei i reaccionar segons el que el client demana.

En aquest projecte, s'aplicarà el *sentiment analysis* per a la classificació de missatges en xarxes socials, una tècnica molt freqüent, ja que és on la població és més expressiva, relacionats amb la Covid-19 i la pandèmia que estem vivint. L'objectiu és realitzar un estudi per veure la tendència emotiva de la població en aquest context.

Paraules clau: aprenentatge automàtic, classificació d'emocions, anàlisis de sentiments, processament del llenguatge natural, xarxes socials

Resumen

El *sentimient analysis* es una técnica de *machine learning* utilizada para clasificar el grado de sentimiento de un texto. Existe desde la más simple clasificación de un mensaje según si expresa un sentimiento positivo o negativo, hasta la más compleja que es capaz de clasificar en diferentes emociones. Es una técnica que cada vez es más utilizada en muchas empresas y sectores, para ver la reacción de la población ante un nuevo producto o servicio y reaccionar según la demanda del cliente.

En este proyecto, se aplicará el *sentimient analysis* para la clasificación de mensajes en redes sociales, una técnica muy frecuente, ya que es donde la población es más expresiva, relacionados con la Covid-19 y la pandemia que estamos viviendo. El objetivo es realizar un estudio para ver la tendencia emotiva de la población en este contexto.

Palabras clave: aprendizaje automático, clasificación de emociones, análisis de sentimientos, procesamiento del lenguaje natural, redes sociales

Agraïments

Després de cinc anys dins al grau, el treball de final de grau ha estat l'última peça per acabar l'esforç i dedicació al llarg d'aquesta etapa. Durant tots aquests anys i sobretot aquests últims mesos no han sigut fàcils, però al final tot esforç té la seva recompensa. Voldria agrair tot el suport que m'han donat els amics més propers donant-me ànims i actitud per tirar endavant, sempre donant-me suport i aixecant-me els ànims en els moments més baixos. També a les meves companyes de pis que han viscut el meu dia a dia i han estat un pilar fonamental per a mi, estant en tot moment i quan més ho necessitava. Evidentment, agrair també a la meva família que, tot i la distància, segueixen fent-me costat en tot moment i en les decisions preses, fent-me sempre tirar endavant. I per descomptat, al Xavier i al Marc que han estat des de l'inici d'aquest projecte ajudant-me a créixer en l'àmbit professional i introduint-me en el món de la recerca en l'àmbit informàtic. Ja per acabar, aprofito per donar les gràcies a tot el professorat i personal docent que he anat tenint al llarg del grau i a la Universitat Politècnica de Catalunya en general per donar-me l'oportunitat d'estudiar aquest grau. Aquest no és un punt i final, sinó un punt i a part de tots els episodis i etapes que em queden. Espero que gaudiu del projecte que he realitzat igual que l'he gaudit jo al llarg d'aquests mesos.

Gràcies a tots.

Contingut

Llistat de figures

Llistat de taules

1 Contextualització i abast

1.1 Context

Aquest treball de final de grau d'enginyeria informàtica de la Facultat d'Informàtica de Barcelona pertany a l'especialitat d'enginyeria del software. El projecte consistirà en realitzar un estudi de recerca per analitzar els sentiments de la població de parla espanyola en relació amb el coronavirus (o Covid-19) a través de Twitter. Aquest estudi estarà realitzat en col·laboració amb el grup de recerca GESSI [1] al qual el director, Xavier Franch Gutiérrez, i codirector, Marc Oriol Hilari, en formen part.

1.2 Formulació del problema

El fenomen de les xarxes socials és quelcom estès arreu del món. Milers i milers de persones les utilitzen diàriament ja sigui per seguir a personatges influents, revisar notícies d'actualitat, mostrar activitats quotidianes o expressar com se senten davant d'una situació, sigui d'actualitat o no.

En aquests moments, el coronavirus és un fet d'actualitat que ha incidit directament sobre les nostres vides en poc més d'un any i que provocat una situació de pandèmia. Tenint tant a mà xarxes socials com Twitter, no és d'estranyar que diàriament la població parli sobre la Covid-19 i la situació de pandèmia que ens envolta.

A partir d'aquí és d'on sorgeix la idea d'aquest estudi. Ara bé, la tasca de classificar un fragment de text en una emoció, anomenada *sentiment analysis,* és quelcom molt ambigu i dispers. A més, si es té en compte el factor idioma, que en aquest cas és l'espanyol, el nombre de solucions existents son escasses i majoritàriament de pagament, tal com analitzarem a la següent secció.

Per atacar aquest hàndicap, la solució proposada és un sistema que permeti el monitoratge i classificació de *tweets* en una emoció concreta en temps real mitjançant tècniques d'aprenentatge automàtic i processament del llenguatge natural.

1.3 Conceptes

1.3.1 *Sentiment analysis*

L'anàlisi de sentiments (en anglès *sentiment analysis* o *opinion mining*) consisteix en determinar la tendència emocional d'un fragment de text mitjançant el processament del llenguatge natural (en anglès *NLP – Natural Language Processing*) i tècniques d'aprenentatge automàtic (en anglès *machine learning*). Podem distingir dos tipus molt clars de *sentiment analysis*: de polaritat i de classificació d'emocions. [2]

1.3.1.1 *Sentiment analysis* de polaritat

L'anàlisi de sentiments de polaritat consisteix en, tal com el nom indica, classificar un text segons la seva polaritat. La polaritat més simple que existeix és positiu/negatiu, però

normalment també se sol afegir el sentiment neutral, indicant que no hi ha una tendència clara cap a ninguna de les polaritats. A partir d'aquí, existeixen diferents variants com, per exemple, la polaritat extensa, que consta d'una classificació de cinc tipus (molt positiu, positiu, neutral, negatiu, molt negatiu). [3]

Per classificar un text en alguna de les polaritats, s'utilitza el mètode basat en regles, el qual implica una rutina bàsica del processament del llenguatge natural mitjançant una extensa llista de paraules classificades com a positives i negatives, o indicant un valor numèric de -1 a 1, per exemple, per indicar el grau de positivitat, negativitat o neutralitat del terme. L'algoritme analitza el contingut del text i busca a la llista quins termes coincideixen amb els del fragment. A partir d'aquí, calcula la freqüència de paraules positives i negatives (o n'extreu la polaritat a partir de la suma dels valors numèrics assignats a cada terme) i dicta el resultat segons el valor màxim. [4]

Aquest mètode és bastant recurrent per la seva senzillesa en la implementació però, ja que no té en compte el context del text, és un mètode poc flexible i imprecís. [5] Tot i això, és un mètode utilitzat sobretot en l'anàlisi de satisfacció de clients en un servei en concret, per exemple, o en crítiques de pel·lícules, llibres o sèries de televisió.

1.3.1.2 Classificació d'emocions

Aquesta segona variant, la classificació d'emocions, és un procediment més complex però alhora més precís. Existeixen moltes variants, ja que depèn del nombre d'emocions que vols obtenir a partir del classificador i quines esculls segons el conjunt de dades d'entrada. Les principals emocions, identificades com a bàsiques [6] segons Paul Ekman [7], un dels psicòlegs més influenciadors del segle XXI, resulten les següents: tristesa, por, alegria, enuig, sorpresa i aversió. En el camp de m*achine learning,* també es sol afegir el sentiment neutral o 'no rellevant' per textos que no mostren cap de les emocions anteriors.

Aquest tipus d'anàlisi de sentiments utilitza l'aprenentatge automàtic per obtenir resultats molt més precisos i exactes que en el cas de *sentiment analysis* de polaritat. En aquesta variant, s'aplica un tipus de *machine learning* anomenat aprenentatge supervisat i, més concretament, aplicant algoritmes de classificació.

1.3.2 *Machine learning*

El concepte d'aprenentatge automàtic (en anglès *machine learning*) es defineix com a un conjunt de mètodes computacionals que, a partir d'un conjunt d'informació, permeten millorar el rendiment o fer prediccions. Aquest conjunt d'informació pot correspondre's a un conjunt de dades etiquetades prèviament per un humà o altres variants d'informació que interaccionen amb l'entorn. En qualsevol cas, la mida i la qualitat de les dades és crucial per l'aprenentatge de la màquina. [8]

L'aprenentatge automàtic s'utilitza per resoldre molt tipus de problemes, entre els quals en destaquen els següents:

- Classificació de textos o documents
- Processament del llenguatge natural (en anglès *Natural Language Processing, NLP*)

- Aplicacions de processat per veu
- Aplicacions de visió per computador
- Entre d'altres

En el nostre cas, el que volem abastar és la classificació de textos i, més concretament, text extret de Twitter (també anomenat *tweet*). Per fer-ho, explicarem resumidament quins tipus d'algoritmes de *machine learning* existeixen i quin és el més adient per al nostre estudi.

Els algoritmes d'aprenentatge automàtic principals son els llistats a continuació [9]:

- **Aprenentatge supervisat** (en anglès *supervised learning*): l'algoritme genera una funció a partir de les dades d'entrada correctament etiquetades mitjançant un procés de *training*.
    - Regressió: utilitzat per predir valors numèrics.
    - Classificació: tècniques per predir valors categòrics o classes (en el nostre cas cada categoria serà una emoció).
- **Aprenentatge no-supervisat** (en anglès un*supervised learning*): conjunt d'algoritmes utilitzat per deduir inferències a partir de conjunt de dades no etiquetades.
- **Aprenentatge per reforç** (en anglès *reinforcement learning*): l'algoritme actua de manera similar a com aprenem els éssers humans. El model o algoritme aprèn contínuament del seu entorn mitjançant la interacció i obté una resposta positiva o negativa basada en l'acció presa. [10]

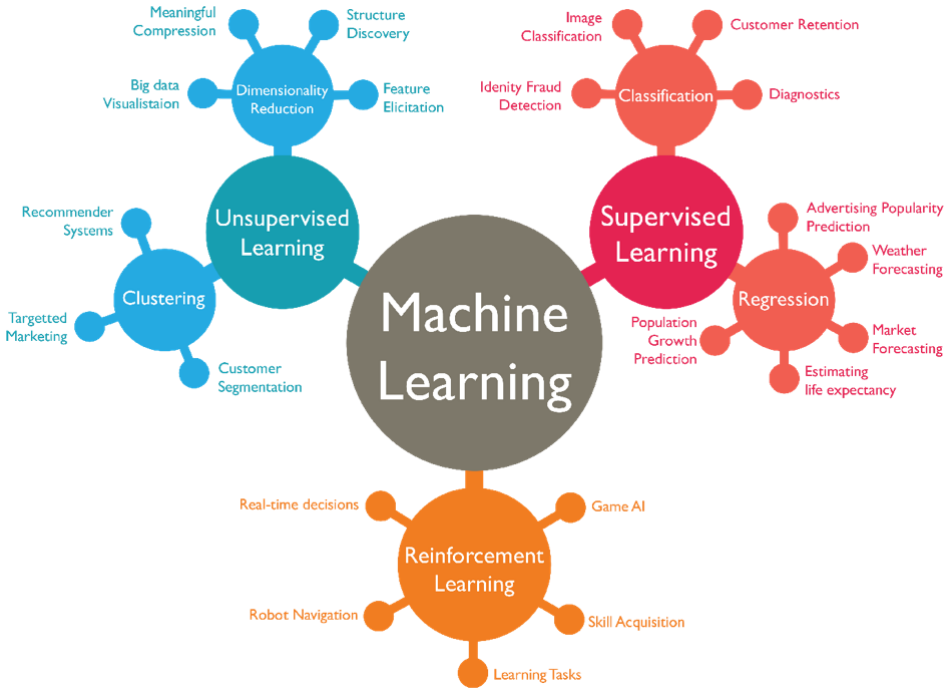

Figura 1.1 Tipus de machine learning [11]

1.3.3 *Natural Language Processing*

El processament del llenguatge natural és una branca de la intel·ligència artificial centrada en l'anàlisi de les comunicacions humanes i, en concret, del llenguatge. Consisteix a transformar el llenguatge natural en llenguatge formal comprensible pels computadors.

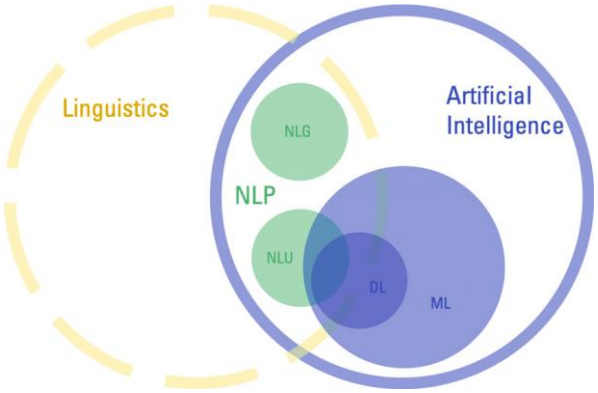

Figura 1.2 Processament del llenguatge natural en la intel·ligència artificial [12]

Per analitzar tasques que requereixen el processament del llenguatge natural, s'ha de passar per les següents fases o etapes [13]:

- **Anàlisi morfològica:** entendre el significat de les paraules a partir del seu morfema (fragment mínim d'una paraula capaç d'expressar el significat d'aquesta).
- **Anàlisi lèxica:** identificar i analitzar l'estructura de les paraules.
  - *Stemming*: consisteix a eliminar els sufixes de les paraules
  - *Lemmatization*: obtenir l'arrel les paraules.
- **Anàlisi sintàctica:** analitzar el sentit gramatical d'una oració a partir de la relació de les paraules que la componen.
- **Anàlisi semàntica:** fa referència a l'extracció del significat d'un text mitjançant l'aplicació d'algoritmes per interpretar les paraules i l'estructura de les oracions.
- **Integració del discurs:** focalitzat en la connexió entre oracions per entendre el significat complet del text, és a dir, entendre el context.
- **Anàlisi pragmàtica:** reinterpretar el significat real de l'oració tenint en compte l'entorn, incloent-hi les intencions i objectiu de l'orador.

És important tenir en compte les diferents etapes i tècniques de processament del llenguatge natural, ja que haurem d'utilitzar algunes d'elles per a la realització del sistema.

1.4 Actors implicats

Diferents actors (o persones interessades) estan implicats ja sigui directament o indirectament en aquest projecte. El llistat a continuació mostra de quins es tracten:

- **Desenvolupadora:** aquest projecte consta d'una sola desenvolupadora (jo mateixa, Ariadna de Arriba Serra) que es dedicarà a realitzar tota la part de

desenvolupament del codi necessari per a la realització de l'estudi i la seva documentació corresponents.

- **Director del projecte:** el director Xavier Franch, professor de la UPC i membre del grup de recerca GESSI (Grup d'Enginyeria del Software i dels Sistemes d'Informació) serà l'encarregat de fer el seguiment de l'estudi.
- **Codirector del projecte:** el codirector Marc Oriol, membre del grup de recerca GESSI (Grup d'Enginyeria del Software i dels Sistemes d'Informació) serà l'encarregat de fer el seguiment de l'estudi juntament amb el director del projecte.
- **GESSI (Grup d'Enginyeria del Software i dels Sistemes d'Informació):** ja que es tracta d'un projecte en col·laboració amb el grup de recerca GESSI, també el grup està interessat en què l'estudi vagi pel camí correcte i s'obtinguin resultats satisfactoris.
- **Altres grups o personal de recerca:** altres grups de recerca o persones que estiguin treballant en estudis semblants, poden utilitzar els resultats obtinguts per encaminar la seva feina en una direcció o una altra.

1.5 Abast

1.5.1 *Objectius*

El principal objectiu d'aquest estudi de recerca és veure la tendència emotiva de la població en relació amb la Covid-19 a través de Twitter. Evidentment, aquest és el focus principal del nostre estudi, però si ens endinsem en el projecte, veiem que sorgeixen els següents objectius i subobjectius:

- Obtenir *tweets* relacionats amb el coronavirus
  - Monitorar els *tweets* filtrant per idioma i tema (Covid-19) a través de l'API de Twitter
  - Classificar els *tweets* en una emoció de les seleccionades
- Construir una (o vàries) eina(es) de *machine learning* per classificar els *tweets*
  - Entrenar el model d'aprenentatge automàtic a partir del conjunt de dades prèviament obtingut i classificat
  - Construir una API per classificar un nou *tweet* amb el model entrenat

A continuació, a la Figura 1.3, es mostra un esquema de la solució dissenyada pel sistema que volem construir i que servirà per exemplificar i entendre millor els objectius explicats i els requeriments funcionals que explicarem en el següent punt.

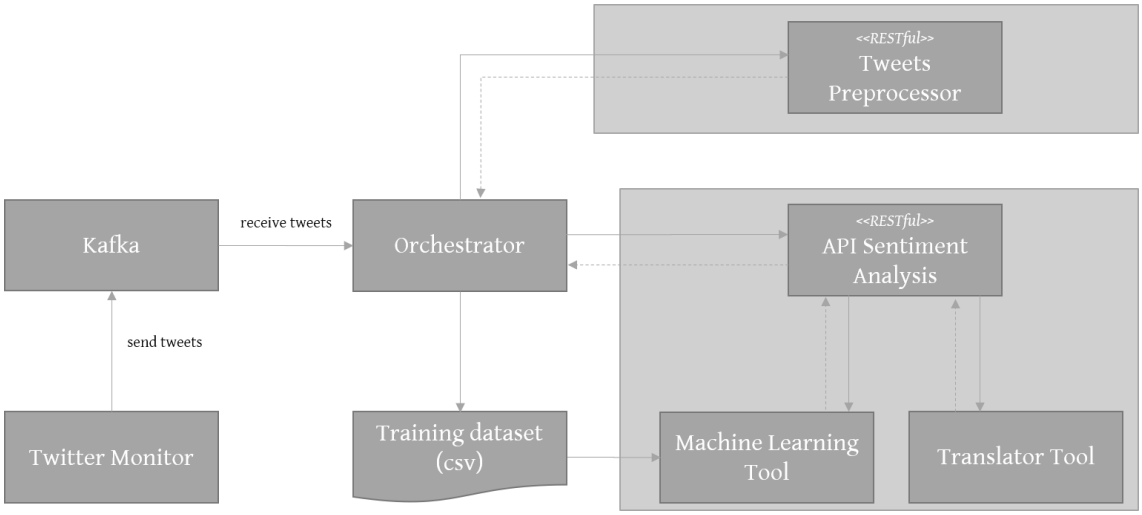

Figura 1.3 Esquema de la solució proposada (Elaboració pròpia)

1.5.2 *Requeriments*

1.5.2.1 Requeriments funcionals

Estretament relacionat amb els objectius, el nostre sistema ha de complir amb els següents requeriments funcionals que dividiré per elements o funcionalitats del sistema:

- Sistema general (*Orchestrator*)
  - El sistema ha d'estar integrat de tal manera que permeti l'anàlisi de *tweets* i la seva classificació en temps real.
- Monitor de Twitter
  - El monitoratge de *tweets* ha de permetre de col·lecció de *tweets* en temps real.
- Classificador
  - Els *tweets* recol·lectats han de passar per una fase de preprocessament del text previ a l'entrenament del model de *machine learning.*
  - Dades d'entrada pel *training*
    - El model de *machine learning* ha d'estar entrenat amb un conjunt de dades correctament etiquetades.
    - El conjunt de dades (en anglès *dataset* o corpus) ha de contenir text únicament en espanyol per entrenar el model. Si no és així, s'ha de traduir prèviament.
    - El corpus ha de contenir un nombre de mostres de cada emoció de manera equitativa
- *API*
  - L'API creada ha de poder ser utilitzada per tasques de traducció
  - L'API ha de permetre la classificació d'un fragment de text (*tweet*) en una de les emocions que definirem més endavant.

1.5.2.2 Requeriments no funcionals

Centrant-nos en els requeriments no funcionals o de qualitat, podem llistar els corresponents al nostre sistema seguint la plantilla Volere d'especificació de requeriments [14]:

- **Adaptabilitat:** el sistema ha de ser fàcilment adaptable els canvis que puguin sorgir durant el procés de desenvolupament com, per exemple, canvis de versions de serveis externs.
- **Disponibilitat:** l'API que permet la classificació dels *tweets* ha d'estar disponible en tot moment, sempre que es disposi de connexió a Internet.
- **Documentat:** el codi ha d'estar correctament documentat per facilitar la lectura o posterior reutilització.
- **Flexibilitat:** el sistema ha de ser capaç d'afegir, modificar o eliminar funcionalitats sense malmetre el sistema actual. Relacionat amb la flexibilitat, volem aconseguir que el sistema tingui un baix acoblament. Això permetria, per exemple, canviar de xarxa social per l'obtenció de text a classificar de manera ràpida i fàcil.
- **Mantenibilitat:** el sistema ha de tenir la capacitat de ser modificat en qualsevol moment de manera eficaç i efectiva, sigui per corregir-lo o millorar-lo.

1.5.3 *Obstacles i riscos*

Durant el desenvolupament del projecte, poden sorgir diferents imprevistos que posen el perill la correcta realització de l'estudi en el temps previst. Aquests riscos poden ser els següents:

- **Falta d'experiència:** al tractar-se d'un estudi que requereix coneixements d'aprenentatge automàtic i tenint en compte que es tracta d'un projecte de l'especialitat d'enginyeria del software i no de computació, serà necessari un període d'aprenentatge autònom per aprendre les tecnologies i coneixements necessaris per aplicar-ho.
- **Erros de programari:** relacionats amb el risc anterior, és possible que apareguin errors mentre s'està programant, sigui per falta d'experiència o per factors externs. És important tenir-ho en compte i deixar un temps previst per corregir els errors que puguin sorgir en el codi.
- **Limitació de temps:** tenint en compte que es tracta d'un projecte de final de grau amb una data d'entrega marcada, si no es planifica bé el projecte i es compleix amb una feina constant diària, el temps pot jugar en contra i pot ser que no s'assoleixin tots els objectius establerts a la data final.
- **Factors externs:** és possible que algun dels servidors que utilitzem per al nostre software, falli en algun moment. Aquest és un fet aliè i incontrolable per nosaltres, però que s'ha de tenir en compte per buscar una possible solució.

2 Estat de l'art i justificació

Una vegada definit el context del projecte, hem de veure quines eines existeixen actualment que facin anàlisi de sentiments. Després de realitzar un estudi de mercat per veure quines eines existents hi havia actualment, he pogut concloure que existeixen moltes eines d'anàlisi de sentiment de polaritat en anglès, però que el nombre es redueix quan es tracta de classificació d'emocions. Si ja passem a l'anàlisi de textos en espanyol, el nombre de recursos queda reduït al mínim i, a més, cap d'ells és de codi obert.

De les solucions trobades, voldria destacar les següents:

- **Lynguo** [15]: eina de monitorització i anàlisi de xarxes socials que realitza un anàlisi de sentiments d'una opinió, emoció o actitud a partir d'un text. Aquesta eina pot classificar el text d'entrada en vint categories d'emocions diferents.
- **Parallel Dots** [16]: es tracta d'una interfície de programació d'aplicacions (en anglès *Application Programming Interface*, *API*) on una de les seves funcionalitats principals és la classificació del text en sis emocions diferents mitjançant algoritmes d'aprenentatge automàtic. El classificador utilitzat ha estat entrenat prèviament utilitzant xarxes neuronals convolucionals a partir d'un *dataset* etiquetat amb l'emoció corresponent per un equip de ParallelDots [17].

A part de les solucions trobades existents en el mercat, hi ha estudis recents sobre l'anàlisi de sentiments per textos en espanyol. Concretament, m'agradaria destacar el TASS (Taller d'Anàlisi Semàntic en el SEPLN), un *workshop* d'anàlisi semàntic i de sentiments que es realitza cada any des del 2012 i on, en l'última edició (2020), constava de dues tasques principals: anàlisi de polaritat a tres nivells (positiu, negatiu i neutre) i detecció d'emocions [18].

Tenint en compte els pocs recursos disponibles per a l'anàlisi de sentiments de text en espanyol i valorant l'estudi fet al TASS 2020, podem contemplar la necessitat de realitzar una eina pròpia per aplicar la classificació d'emocions de textos en espanyol i, més concretament, de *tweets* relacionats amb el coronavirus que és el que ens interessa per aquest estudi.

3 Gestió del projecte

En aquesta secció es detallarà tot el que té a veure amb la gestió del projecte en diferents aspectes: la planificació temporal, la metodologia de treball per a la realització del projecte, la viabilitat econòmica i la sostenibilitat.

3.1 Planificació temporal

La duració d'aquest projecte serà de quatre mesos, aproximadament. La seva data d'inici va ser el passat 23 de febrer de 2021 i està prevista la data de la lectura i defensa per a l'última setmana de juny, és a dir, del 28 de juny al 2 de juliol de 2021.

Aquest projecte està previst realitzar-lo amb 478 h en total, més endavant es detallarà a què es dedicaran cada una d'aquestes hores. La càrrega de treball aproximada per setmana serà de 25 – 30 h, però pot variar segons es vagi avançant i/o les dificultats que vagin sorgint.

3.1.1 *Descripció de les tasques*

En aquest apartat descriuré les tasques del projecte necessàries per finalitzar el projecte en el temps establert. Per fer-ho, utilitzaré els termes i nomenclatura següent:

- **Tasca (T):** acció del projecte a realitzar per complir un o diversos objectius.
- **Èpiques (E):** cada èpica englobarà un conjunt de tasques relacionades per una temàtica comuna.
- **Iniciatives (I):** conjunt d'èpiques amb un objectiu comú.

I1: Documentació i comunicació

Aquesta iniciativa forma part de totes les tasques que tenen a veure amb la documentació i comunicació del projecte, tenint en compte la part de gestió, comunicació dins l'equip, documentació durant el projecte, defensa davant del tribunal, entre d'altres.

E1: Gestió de projecte

En aquesta èpica estaran englobades totes les tasques referents a la gestió del projecte. Les tasques que conté son les següents:

- **T1. Definició d'abast i contextualització:** definició del context del projecte i estat de l'art, introducció a conceptes tècnics i marcatge dels objectius i requeriments. Descripció de la metodologia utilitzada per a la gestió del projecte.
- **T2. Planificació temporal:** planificació temporal per a l'execució total del TFG, incloent-hi les diferents fases del projecte i els seus requeriments associats.
- **T3. Pressupost i sostenibilitat:** anàlisis de la sostenibilitat del projecte i pressupost necessari per a realitzar-lo.

- **T4. Integració document final:** integració de l'abast i context, planificació temporal i pressupost i sostenibilitat en un sol document.

E2: Documentació

En aquesta èpica hi haurà totes les tasques relacionades amb la documentació del projecte, llevat de les de gestió que s'han realitzat a l'inici.

- **T5. Informe de seguiment:** realització d'un informe a la meitat del projecte que reflecteix tot el que s'ha dut a terme i els avenços durant el projecte.
- **T6. Memòria:** document final que conté tota la realització del projecte.
- **T7. Presentació:** preparació de la presentació final com a suport de la defensa del TFG.

E3: Comunicació

Aquesta èpica consta de totes les tasques referents a la comunicació, sigui dins l'equip o externa.

- **T8. Reunions:** reunions setmanals de l'equip (director, codirector i jo) per veure els avenços del projecte.
- **T9. Defensa:** defensa del projecte en una sessió pública davant d'un tribunal que consta d'un president, secretari i vocal.

I2: Desenvolupament

Aquesta segona iniciativa implica tot el desenvolupament del projecte. Al tractar-se d'un projecte àgil, cada una de les èpiques i tasques passarà per una fase de *testing* i validació abans de completar-se al final de cada *sprint.* A la Taula 3.1 de la següent secció, es mostren les estimacions detallades on es reflecteix aquesta fase amb les hores dedicades al rol del tester.

E4: *Twitter monitor*

Aquesta èpica està dedicada a la realització d'una eina que permeti el monitoratge de *tweets* en temps real.

- **T10. Compte *Twitter Developer*:** creació d'un compte de Twitter Developer per tenir accés a l'API de Twitter.
- **T11. Connexió API Twitter:** programació d'un sistema que es connecti amb l'API de Twitter per obtenir els *tweets* en temps real aplicant-hi els filtres corresponents.

E5: API *Sentiment Analysis*

Èpica enfocada a la realització d'una API REST (de l'anglès *Representational State Transfer*) per a l'aplicació d'anàlisis de sentiments en un fragment de text.

- **T12. Creació *REST API Sentiment Analysis*:** crear una API REST pròpia que permeti generar *requests* per analitzar el sentiment d'un fragment de text i tasques de traducció.

- **T13. Connexió API Traductor:** connectar l'API pròpia amb una API que permeti la traducció d'un fragment de text, com per exemple Google o Microsoft.
- **T14. Connexió API ParallelDots:** connectar amb la versió *demo* de ParallelDots per comprovar que la nostra API funciona per analitzar un fragment de text. Posteriorment servirà per connectar-ho amb l'API pròpia del model (o models) de *machine learning.*

E6: *Orchestrator*

Aquesta èpica estarà focalitzada a la realització d'un sistema que englobi els altres sistemes creats prèviament (o posteriorment).

- **T15. Creació *Orchestrator*:** creació d'un sistema que faci de connexió amb totes les eines desenvolupades. Aquest sistema recaptarà els *tweets* (connectant amb el Kafka on hi haurà els *tweets* enviats pel monitor de Twitter) i aplicarà *sentiment analysis*, traduint-los prèviament si és necessari (connectant amb l'API pròpia creada).

E7: *Kafka Server*

En aquesta èpica es crearà la connexió amb el Kafka server per enviar i rebre els *tweets*.

- **T16. Connexió Kafka – *Twitter Monitor*:** connexió del monitor de Twitter amb el servidor Kafka per permetre que s'enviïn *tweets* a un tòpic de Kafka i així desacoblar el monitor del sistema.
- **T17. Connexió Kafka – *Orchestrator*:** connexió del *orchestrator* amb Kafka per poder rebre els *tweets* enviats prèviament pel monitor de Twitter.

E8: Eina de preprocessat

Èpica dedicada a la creació d'una API REST que aplicarà preprocessament de *tweets* per una millor classificació d'emocions en el model de *machine learning*.

- **T18. Preprocessat text:** creació d'un programa que permeti processar els *tweets* abans d'analitzar-ne el sentiment o entrenar el model de *machine learning*.
- **T19. Creació REST API Preprocessat *tweets:*** a partir del programa creat per preprocessar *tweets*, crear una API que permeti fer peticions externes.
- **T20. Millora en el preprocessat:** canvis en el programa per tal d'obtenir millors resultats i precisió en l'anàlisi del sentiment.

E9: Models *machine learning*

En aquesta èpica es crearà un (o més) model(s) de *machine learning* i tot el que això implica, és a dir, entrenament, validació i millora (i així iterativament fins a trobar un model amb resultats satisfactoris). També es crearà una API per fer peticions des de l'API d'anàlisis de sentiments prèviament creada.

- **T21. Creació model *machine learning*:** creació d'un model de *machine learning* enfocat a l'anàlisi de sentiments.

- **T22. Creació REST API model *machine learning*:** creació d'una API que permeti fer peticions de classificació d'un text en una emoció a partir del model creat.
- **T23. Entrenament model *machine learning*:** habilitar el programa perquè permeti l'entrenament del model a partir d'un conjunt de dades d'entrada.
- **T24. Validació model *machine learning*:** extreure mètriques i dades rellevants per comprovar la precisió i èxit del model.
- **T25. Millora model *machine learning*:** canvis en el programa per tal d'obtenir millors resultats i precisió en l'anàlisi del sentiment.

3.1.2 *Dependències*

En aquesta secció es mostra de forma visual les dependències entre tasques. A la primera imatge (Figura 3.1) es mostren les dependències corresponents a la iniciativa 1 (I1: Documentació i comunicació), mentre que a la segona i tercera (Figura 3.2 i Figura 3.3) es poden observar les dependències entre tasques de la iniciativa 2 (I2: Desenvolupament). Cal tenir en compte que no totes les tasques a realitzar tenen dependències entre elles.

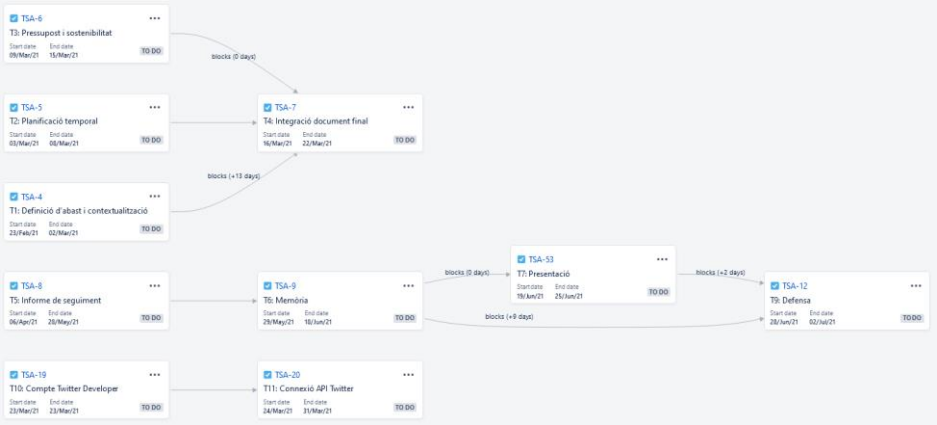

Figura 3.1 Dependències entre tasques de "I1: Documentació i comunicació" (Elaboració pròpia)

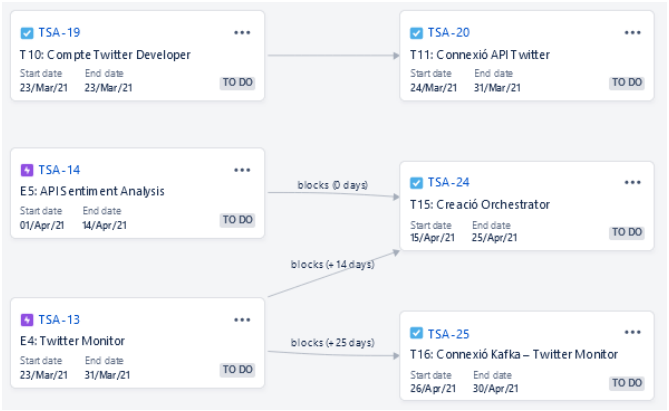

Figura 3.2 Dependències entre tasques de "I2: Desenvolupament" (Elaboració pròpia)

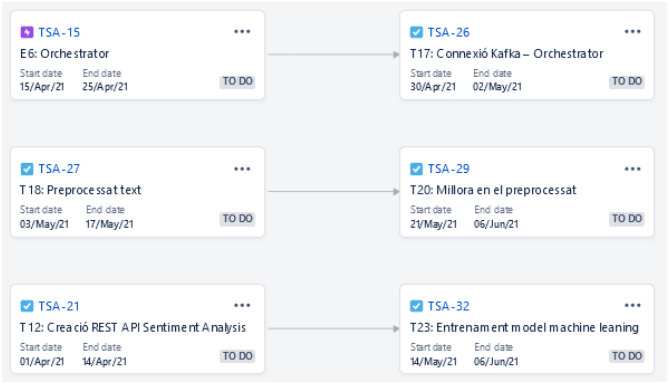

Figura 3.3 Dependències entre tasques de "I2: Desenvolupament" (Elaboració pròpia)

3.1.3 Recursos

En l'àmbit de recursos, podem distingir dues categories principals necessàries pel projecte: recursos humans i recursos materials.

Recursos humans

Els recursos humans fan referència als rols necessaris que necessitarem per a la realització d'aquest projecte. En aquest cas jo assumiré tots o gran part dels rols, i son els següents:

- ***Product owner (PO)*:** representant del client i encarregat de gestionar el projecte.
- ***Scrum master (SM):*** líder de l'equip Scrum que supervisa i s'encarrega de que l'equip segueixi la metodologia, ajudant-lo quan sigui necessari i desenvolupant tasques organitzatives com la planificació de reunions.
- **Equip de desenvolupament**
  - **Arquitecte software (AS):** encarregat del disseny de l'arquitectura i els components del sistema.
  - **Dissenyador interfície (DI):** responsable de que el disseny de la interfície a desenvolupar sigui atractiva, fàcil d'usar i que compleixi amb els requisits no funcionals del projecte.
  - **Programador (P):** desenvolupador de la part tècnica del sistema, dictada per l'arquitecte software.
  - **Tester (T):** realitzador de proves i tests per assegurar-se del correcte funcionament del sistema, és a dir, que no generi errors i que segueixi el comportament esperat.

Recursos materials

A banda dels recursos humans, també hem de tenir en compte quins recursos materials necessitarem. En aquest cas, seran necessaris tant recursos hardware com recursos software.

- **Hardware:**
  - Portàtil: el desenvolupament del projecte es realitzarà amb un *MSI Prestige 14A10SC Intel® Core(TM) i7-10710U CPU* de 16GB de RAM.
- **Software:**
  - Eines de la FIB: Atenea (At), Racó (R)
  - Microsoft: Microsoft Word (W), Microsoft PowerPoint (Ppt)
  - GitHub (Git)
  - GMeet (GM)
  - Visual Studio Code (VS)
  - IntelliJ IDEA (IntelliJ)
  - Swagger (Swg)
  - Windows PowerShell (WinPS)
  - JIRA
  - Excel (Ex)
  - Zotero (Z)
  - Servidors del GESSI (SG)

3.1.4 *Estimacions i Gantt*

Estimacions

Codi	Descripció	Dedicació (h)	Rol						Recursos
			PO	SM	AS	DI	P	T	
I1	**Documentació i comunicació**	**149.5**	**14.5**	**144**	**0**	**0**	**0**	**0**	
E1	Gestió de projecte	60	2.5	57.5					
T1	Definició d'abast i contextualització	24	1	23					At, W, Z
T2	Planificació temporal	8	0.5	7.5					At, W, JIRA
T3	Pressupost i sostenibilitat	9	0.5	8.5					At, W, Ex
T4	Integració document final	19	0.5	18.5					At, W, R, Z, JIRA, Ex
E2	Documentació	80	3	77					
T5	Informe de seguiment	40	2	38					W, R, Z
T6	Memòria	30	1	29					W, R, Z
T7	Presentació	10		10					W, Ppt
E3	Comunicació	9.5	9	9.5					
T8	Reunions	9	9	9					GM
T9	Defensa	0.5		0.5					GM (o presencial)
I2	**Desenvolupament**	**328.5**	**0**	**0**	**30**	**5**	**188.5**	**100**	
E4	*Twitter monitor*	10.5	0	0	0	0	7.5	3	Git
T10	Compte *Twitter Developer*	0.5					0.5		-
T11	Connexió API Twitter	10					7	3	VS
E5	API *Sentiment Analysis*	48	0	0	10	5	24	9	Git, SG
T12	Creació *REST API Sentiment Analysis*	30			10	5	10	5	IntelliJ, Swg
T13	Connexió API Traductor	10					8	2	IntelliJ
T14	Connexió API ParallelDots	8					6	2	IntelliJ
E6	*Orchestrator*	40	0	0	10	0	20	10	Git
T15	Creació *Orchestrator*	40			10		20	10	IntelliJ
E7	*Kafka Server*	35	0	0	5	0	17	8	Git
T16	Connexió Kafka – *Twitter Monitor*	20			5		10	5	IntelliJ, WinPS
T17	Connexió Kafka – *Orchestrator*	15					7	3	IntelliJ, WinPS
E8	Eina de preprocessat	60	0	0	0	0	40	20	Git, SG
T18	Preprocessat text	30					30		VS
T19	Creació REST API Preprocessat *tweets*	10					10		VS
T20	Millora en el preprocessat	20						20	VS
E9	Models *machine learning*	135	0	0	5	0	80	50	Git, SG
T21	Creació model *machine learning*	40			5		35		VS
T22	Creació REST API model *machine learning*	5					5		VS
T23	Entrenament model *machine learning*	20					20		VS
T24	Validació model *machine learning*	30					20	10	VS
T25	Millora model *machine learning*	40						40	VS
TOTAL		**478**							

Taula 3.1 Taula d'estimacions de tasques (Elaboració pròpia)

Diagrama Gantt

A continuació, es mostra la planificació temporal detallada a partir del diagrama de Gantt. Tal i com indica a la llegenda, els colors groc, taronja i vermell indica el nivell de risc de cada tasca que es correspon amb baix (*low*), mitjà (*medium*) i alt (*high*), respectivament. Les tasques marcades en verd corresponen a les reunions setmanals. La resta de tasques o èpiques que engloben un conjunt de tasques estan marcades en gris.

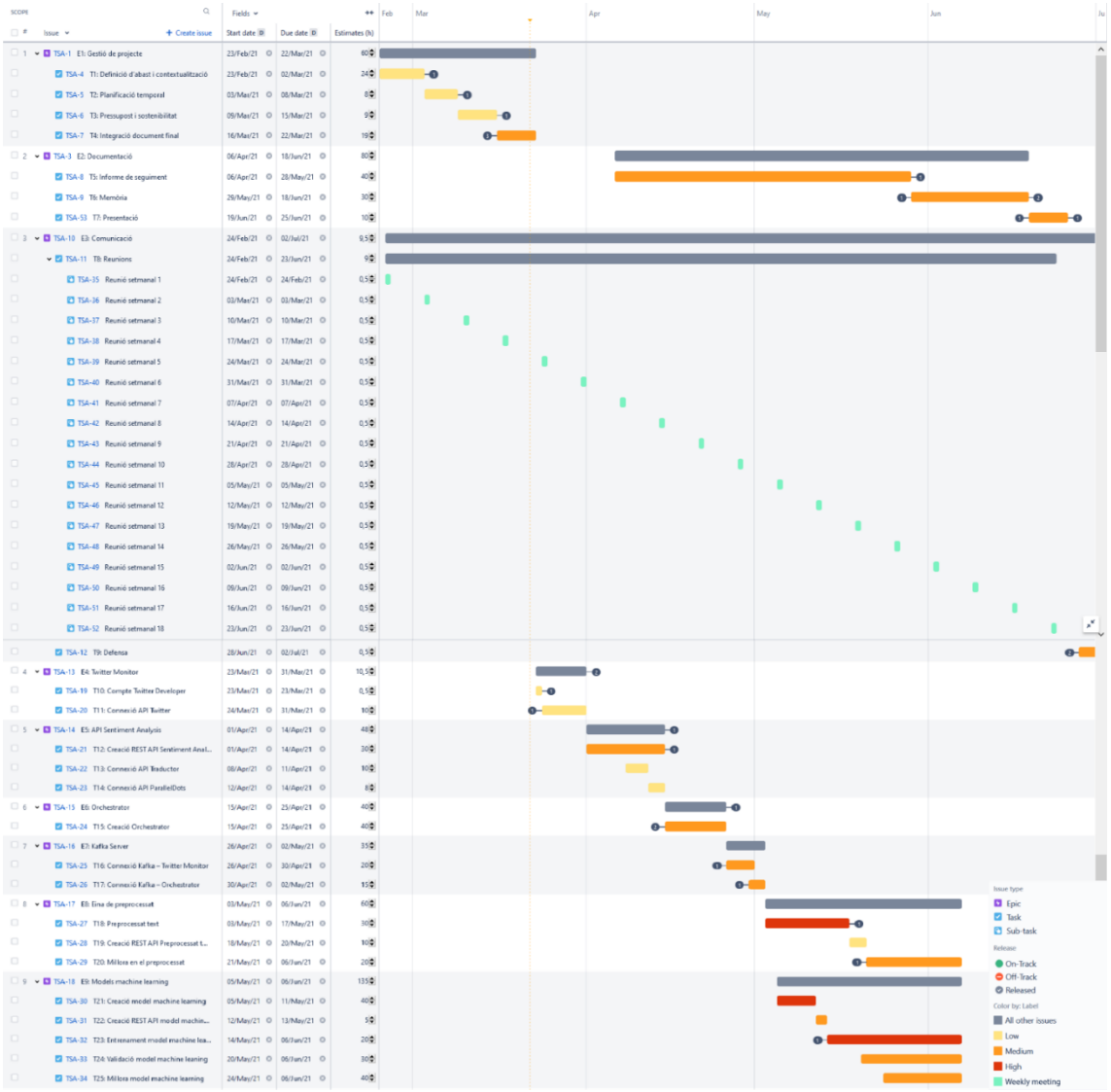

Figura 3.4 Diagrama de Gantt (Elaboració pròpia)

3.1.5 *Gestió del risc*

Per cadascun dels obstacles i riscos definits a l'abast del projecte s'ha de tenir un compte un pla alternatiu que el solucioni i una estimació del temps que prendria aplicar-ho.

Obstacles i plans alternatius

En aquest apartat, analitzarem cadascun dels riscos i proposarem la solució alternativa:

- **Falta d'experiència**
  - **Alternativa:** donat que la idea d'aquest projecte ja està vista des de fa uns mesos, s'ha tingut en compte i s'han adquirit els coneixements necessaris per a realitzar-lo. Tot i això, tret que sóc novícia en el tema, poden sorgir problemes que em prenguin més temps a resoldre que un expert. Aquest temps addicional d'aprenentatge autònom i/o resolució de dubtes amb un expert dins el grup de recerca, per exemple, s'ha de tenir en compte.
  - **Impacte:** mig
  - **Recursos addicionals:** temps (hores extres)
- **Erros de programari**
  - **Alternativa:** dedicació d'hores addicionals per trobar una solució a l'error trobat.
  - **Impacte:** mig
  - **Recursos addicionals:** temps (hores extres)
- **Limitació de temps**
  - **Alternativa:** si per algun dels riscos del projecte s'augmenta el nombre d'hores de dedicació, s'ha de reestructurar el projecte i planificar les tasques de tal manera que es compleixi amb la data d'entrega establerta.
  - **Impacte:** baix
  - **Recursos addicionals:** temps (hores extres)
- **Factors externs**
  - **Alternativa:** disposar d'una versió del software en local per executar-ho en el moment de mostrar el seu funcionament.
  - **Impacte:** mig
  - **Recursos addicionals:** software

3.1.6 *Canvis en la planificació*

Inicialment el projecte estava pensat per a ser desplegat a una plataforma *cloud* com, per exemple, Heroku. Actualment, tenim a disposició els servidors del grup GESSI que ens permeten desplegar el sistema. Això afecta lleugerament als recursos materials i la gestió del risc. Aquests canvis ja han estat contemplats i modificats en els apartats anteriors.

En relació al diagrama de Gantt i les estimacions, el projecte segueix el transcurs previst i no han sorgit imprevistos que afectin negativament en el compliment d'aquest en el temps fixat inicialment.

3.2 Metodologia de desenvolupament

Pel desenvolupament del projecte, utilitzaré la metodologia àgil i més concretament *Scrum*. *Scrum* és un *framework* que facilita el treball en equip en la gestió de projectes permetent trobar solucions adaptades a problemes complexes de manera dinàmica. Aquest mètode de treball es basa en tres aspectes principals [19]:

- **Transparència**: tots els membres implicats tenen una visió clara i global del projecte en tot moment.
- **Inspecció**: es duen a terme reunions de seguiment freqüents que permeten tenir un control de tot el procés i detectar problemes a priori.
- **Adaptació**: els membres de l'equip s'adapten als canvis què puguin sorgir per completar cada etapa amb èxit.

A continuació, es mostra un esquema (veure Figura 3.5) per clarificar la metodologia i procediment que segueix *Scrum*, seguit de les definicions de cada concepte.

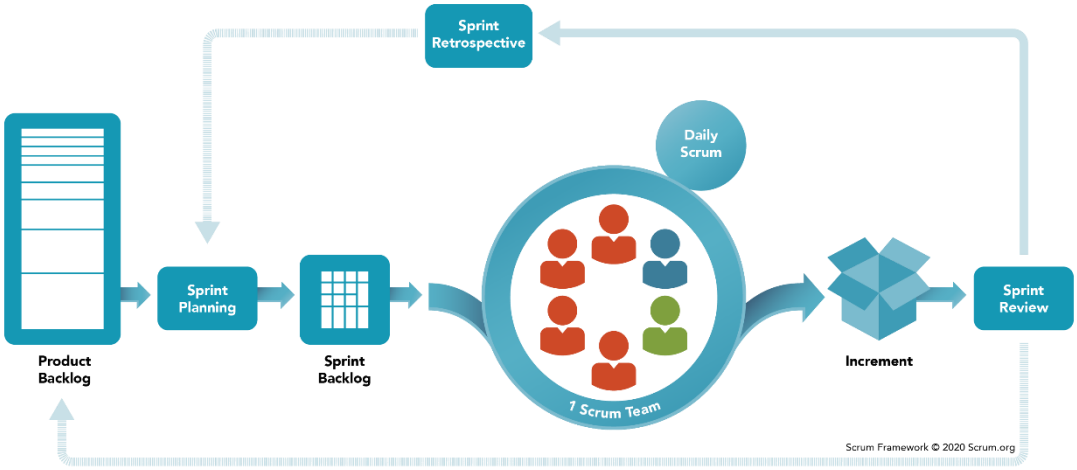

Figura 3.5 Scrum framework [20]

Per aquest projecte (i qualsevol projecte *Scrum*), hi ha una sèrie de conceptes clau a tenir en compte [19]:

- ***Sprint:*** cada una de les fases en les quals es divideix el projecte de durada limitada on les idees conceptuals prenen valor.
- ***Sprint Planning:*** inicia cada *sprint* assegurant que inclou quines son les tasques o objectius a realitzar, el valor d'aquests i com s'aconseguiran.
- ***Daily Scrum:*** tasca diària de 15 minuts on es fa una introspectiva del treball realitzat durant el dia.
- ***Sprint Review:*** consisteix a inspeccionar el *sprint* realitzat i determinar les adaptacions a realitzar en tasques futures.

- ***Sprint Retrospective:*** l'equip inspecciona com s'ha desenvolupat el darrer *sprint* en relació amb eines, processos, *definition of done*, entre d'altres; i s'analitza els problemes que hi ha hagut durant la seva realització.

Tenint clars els conceptes anteriors, cal mencionar una sèrie d'artefactes que formen part de cada un dels *sprints* i de tot el procés:

- ***Product Backlog:*** llista ordenada de tot el necessari per millorar el producte. Cada un dels elements del *product backlog* es fragmenta en tasques més petites i precises. El *producte backlog* ha de ser fidel al *product goal* (objectiu del producte o projecte).
- ***Sprint Backlog:*** està constituït per un subconjunt d'elements del *product backlog*. El *sprint backlog* es construeix seguint el *sprint goal*, que correspon a l'objectiu únic del *sprint*.
- **Increment:** pas endavant cap al *product goal.* Pot haver-hi un o més increments durant un *sprint*. En el *sprint review* es consoliden els increments realitzats. Cada increment ha de complir amb la *DoD (Definition of Done)*, una descripció formal de les mesures de qualitat a satisfer.

3.2.1 *Eines de seguiment*

1.5.3.1 JIRA

JIRA [21] és un software que permet mantenir un seguiment del projecte seguint metodologies àgils com *Kanban* o *Scrum* (que és el que utilitzarem en aquest cas). Permet d'una manera molt visual tenir un control de tot el taulell *Scrum*, definir els *sprints*, marcar les èpiques i tasques assignant-hi els *story points* corresponents, tenir un *roadmap* del projecte, generar reports del treball realitzat, entre d'altres. Ofereix un ventall molt ampli d'opcions adaptat a les metodologies àgils.

1.5.3.2 GitHub

GitHub [22] és una eina de control de versions que ens permetrà tenir diferents repositoris accessibles i tenir un control durant la realització del codi.

1.5.3.3 Google Drive

Google Drive [23] és un servei d'emmagatzematge d'arxius al núvol que permetrà tenir documents compartits amb el director i codirector del projecte d'una manera més dinàmica i eficient i permetrà l'edició en temps real de documents en col·laboració, si s'escau.

1.5.3.4 Google Meet

Google Meet [24] és un servei de comunicació de vídeo i àudio que permetrà realitzar les diferents reunions amb el director i codirector de projecte.

3.3 Gestió econòmica

En la gestió econòmica del projecte, s'han de tenir en compte diversos factors. S'han d'identificar els costos del projecte per posteriorment estimar els costos finals i realitzar el pressupost i també s'ha de tenir en compte un control de gestió per, quan finalitzi el projecte, veure el cost real respecte l'estimat.

3.3.1 *Identificació de costos*

Primerament, hem de detallar els costos relacionats amb recursos humans, recursos materials (de hardware i software), les despeses generals com electricitat o internet i el cost dels imprevistos que puguin sorgir.

Recursos humans

Per a cadascun dels rols identificats en el projecte, he especificat el salari base per hora segons Indeed [25] per poder calcular-ne posteriorment el cost total.

Rol		Salari base (€/h)
Product owner (PO)		18,55 €
Scrum master (SM)		15,66 €
Equip de desenvolupament	Arquitecte software (AS)	19,60 €
	Dissenyador interfície (DI)	12,56 €
	Programador (P)	13,71 €
	Tester (T)	18,77 €

Taula 3.2 Salari/h per rols (Elaboració pròpia)

A la taula a continuació (Taula 3.3) es mostra el cost per activitat i el cost total tenint en compte les hores dedicades a cada activitat i el salari de cada un dels rols.

Codi	Descripció	Dedicació (h)	Rol						Cost
			PO	SM	AS	DI	P	T	
I1	**Documentació i comunicació**	**149.5**	**14.5**	**144**	**0**	**0**	**0**	**0**	**2524.02 €**
E1	Gestió de projecte	60	2.5	57.5					946.83 €
T1	Definició d'abast i contextualització	24	1	23					378.73 €
T2	Planificació temporal	8	0.5	7.5					126.73 €
T3	Pressupost i sostenibilitat	9	0.5	8.5					142.39 €
T4	Integració document final	19	0.5	18.5					298.99 €
E2	Documentació	80	3	77					1261.47 €
T5	Informe de seguiment	40	2	38					632.18 €
T6	Memòria	30	1	29					472.69 €
T7	Presentació	10		10					156.60 €
E3	Comunicació	9.5	9	9.5					315.72 €
T8	Reunions	9	9	9					307.89 €

T9	Defensa	0.5		0.5					7.83 €
I2	**Desenvolupament**	**328.5**	**0**	**0**	**30**	**5**	**188.5**	**100**	**4593.37 €**
E4	*Twitter monitor*	10.5	0	0	0	0	7.5	3	159.14 €
T10	Compte *Twitter Developer*	0.5					0.5		6.86 €
T11	Connexió API Twitter	10					7	3	152.28 €
E5	API *Sentiment Analysis*	48	0	0	10	5	33	9	719.23 €
T12	Creació *REST API Sentiment Analysis*	30			10	5	10	5	489.75 €
T13	Connexió API Traductor	10					8	2	147.22 €
T14	Connexió API ParallelDots	8					6	2	82.26 €
E6	*Orchestrator*	40	0	0	10	0	20	10	657.90 €
T15	Creació *Orchestrator*	40			10		20	10	657.90 €
E7	*Kafka Server*	35	0	0	5	0	17	8	481.23 €
T16	Connexió Kafka – *Twitter Monitor*	20			5		10	5	328.95 €
T17	Connexió Kafka – *Orchestrator*	15					7	3	152.28 €
E8	Eina de preprocessat	60	0	0	0	0	40	20	923.80 €
T18	Preprocessat text	30					30		411.30 €
T19	Creació REST API Preprocessat *tweets*	10					10		137.10 €
T20	Millora en el preprocessat	20						20	375.40 €
E9	Models *machine learning*	135	0	0	5	0	80	50	2133.30 €
T21	Creació model *machine learning*	40			5		35		577.85 €
T22	Creació REST API model *machine learning*	5					5		68.55 €
T23	Entrenament model *machine learning*	20					20		274.20 €
T24	Validació model *machine learning*	30					20	10	461.90 €
T25	Millora model *machine learning*	40						40	750.80 €
TOTAL		**478**							7117.38 €

Taula 3.3 Cost per activitats i rols (Elaboració pròpia)

Recursos materials

Hardware

Per a calcular el cost total dels recursos hardware utilitzats, hem de tenir en compte el preu base del material i l'amortització calculada a partir de la fórmula a continuació, tenint en compte el cost i la vida útil dels recursos segons la Taula 3.4 i els dies laborables del 2021 [26]:

$$Amortització = \frac{cost}{vida\ útil\ x\ dies\ laborables/any\ x\ dedicació\ diària}\ x\ duració\ projecte\ (h) = \frac{1298.33€}{4\ anys\ x\ 251\frac{dies}{any}\ x\frac{5h}{dia}}\ x\ 478h \approx \mathbf{123.63}\ €$$

Recurs	Cost	Vida útil	Amortització
MSI A10SC-067XES i7-10710U	1298.33€	4 anys	123.63€

Taula 3.4 Cost de recursos hardware (Elaboració pròpia)

Software

Per a la realització d'aquest projecte de recerca, tots els recursos software utilitzats son gratuïts o s'ha utilitzat la versió limitada o de prova gratuïta.

Despeses generals

A banda dels recursos humans i materials, també s'han de tenir en compte altres despeses més indirectes com podria ser el llum, l'aigua, el lloguer de l'edifici on es treballa, entre d'altres. En aquest cas, donada la situació de la Covid-19, tots els participants de projecte estem teletreballant. Per aquest motiu, calcularé les despeses d'electricitat [27], tenint en compte la potència del portàtil utilitzat [28], i Internet [29] de la llar per a una sola persona i hi afegiré un 10% del director i un altre 10% pel codirector del projecte, per tenir en compte la seva part en el projecte.

$$Elecricitat = \frac{0.0410€}{kWh}\ x\ 0.227kW\ x\ 478h = 4.45\ €$$

$$Internet = \frac{30.95€}{mes}\ x \frac{12mesos}{1any} x \frac{1any}{365dies}\ x \frac{1dia}{24h}\ x\ 478h = 20.26\ €$$

Recurs	Cost	Dedicació personal	Cost total
Electricitat	4.45 €	1 + 0.1 + 0.1	5.34 €
Internet	20.26 €	1 + 0.1 + 0.1	24.31 €
TOTAL			**29.65 €**

Taula 3.5 Cost de despeses generals (Elaboració pròpia)

Imprevistos

En aquest projecte, per resoldre els imprevistos que puguin haver-hi la solució és dedicar-hi més hores. Aquestes hores extres dedicades a cada risc o imprevist es tradueixen amb un augment de dedicació del personal i, per tant, un cost addicional al projecte. En el cas dels factors externs, només és necessari software extra que és gratuït, per tant, no hi haurà cost addicional. A la taula següent (Taula 3.6) es mostra el detall de dedicació extra en hores per a cada risc i com afecta al cost:

Risc	Dedicació extra per rol (h)						Cost	Probabilitat	Cost total
	PO	SM	AS	DI	P	T			
Falta d'experiència			5		20	20	747.60€	55%	411.18 €
Errors de programari					30		411.30€	60%	246.78 €
Limitació de temps		10					156.60 €	10%	15.66€
Factors externs							-	5%	-
TOTAL									**673.62 €**

Taula 3.6 Cost d'imprevistos (Elaboració pròpia)

3.3.2 *Estimació de costos*

Per sintetitzar el pressupost global del projecte, hem de tenir en compte tots els costos calculats a la secció anterior. A la taula següent (Taula 3.7), podem veure de forma resumida el cost total del projecte:

Recurs		**Cost**
Recursos humans		7117.38 €
Recursos materials	Hardware	123.63 €
	Software	-
Despeses generals		29.65 €
Imprevistos		673.62 €
Subtotal		7944.28 €
	Contingències (15%)	1191.64 €
TOTAL		**9135.92 €**

Taula 3.7 Estimació total de costos (Elaboració pròpia)

3.3.3 *Control de gestió*

Per finalitzar la secció de gestió econòmica, cal fer un control de les possibles desviacions que pugui haver entre el pressupost calculat i el cost real del projecte. Per a cada secció explicada anteriorment, s'han de tenir en compte les següents desviacions:

- Desviacions recursos humans:
  - $Desviació_{preu-recursos\ humans} = (cost\ estimat - cost\ real)\ x\ consum\ real\ (h)$
  - $Desviació_{consum-recursos\ humans} = (consum\ estimat - consum\ real)\ x\ cost\ real$
- Desviacions recursos materials:
  - $Desviació_{hardware} = cost\ estimat - cost\ real$
- Desviacions imprevistos:
  - $Desviació_{imprevistos} = cost\ estimat - cost\ real$
- Desviacions totals:
  - $Desviació_{total} = cost\ total\ estimat - cost\ total\ real$

Durant la realització del projecte, s'haurà d'anar anotant les hores reals dedicades a cada tasca per poder calcular el cost real del projecte. En finalitzar-lo, podrem comprovar les desviacions produïdes i el cost real per la realització total del projecte.

3.4 Informe de sostenibilitat

3.4.1 *Autoavaluació enquesta*

Si bé és cert que la sostenibilitat no és el primer que es té en compte en dur a terme un projecte informàtic, és molt important fer una anàlisi prèvia de com afectarà el nostre projecte al medi ambient donat que estem en un punt que els humans consumim més recursos dels que el planeta pot generar. Però no només s'ha de tenir en compte aspectes ambientals sinó també econòmics i socials.

Tot i que durant el meu pas pel grau s'han tractat diverses vegades la sostenibilitat en projectes en diferents competències transversals d'assignatures, després de realitzar l'enquesta d'EDINSOST, m'he adonat que hi ha molts aspectes que desconec o que no tinc suficientment en compte en iniciar un nou projecte. Principalment, tot i valorar-ho en els projectes elaborats, no tinc els coneixements suficients en quan a indicadors per mesurar detalladament cadascun dels tres àmbits que engloba la sostenibilitat. Tot i això, he pogut adonar-me que no fer un estudi econòmic, social i mediambiental pot tenir un gran impacte negatiu en els nostres projectes.

3.4.2 *Dimensió econòmica*

En referència a la gestió econòmica, he calculat el pressupost d'acord amb el salari base per a cada rol. Evidentment, com que es tracta d'un treball de final de grau individual (tot i estar amb col·laboració amb el grup de recerca GESSI), gran part dels rols els assumeixo jo mateixa i el salari és per una sola persona i no per rols.

Tal com vaig comentar a l'estat de l'art, es tracta d'una solució *open-source* accessible a tothom a través d'una API. Per tant, tenint en compte que les solucions existents al mercat son totes de pagament, a banda dels costos calculats anteriorment i de manteniment, aquest producte serà accessible de manera gratuïta per a tots els *stakeholders*.

3.4.3 *Dimensió ambiental*

En tractar-se d'un projecte software i tenint en compte que s'utilitzen els mínims recursos materials, l'impacte ambiental no és molt elevat. L'únic impacte ambiental directe és el de l'electricitat consumida pel dispositiu utilitzat. Tot i això, m'agradaria destacar que el portàtil utilitzat no serà exclusivament dedicat aquest projecte, sinó que serà un producte utilitzat durant més anys dins el grup de recerca.

Un altre factor a tenir en compte és el transport. En aquest moment es realitza teletreball, però quan passi l'estat de pandèmia i es torni al treball presencial, s'ha de tenir en compte l'impacte al medi ambient del transport. És per això, que per reduir-lo al màxim, evitarem agafar vehicle propi i utilitzar el transport públic. I, en cas que sigui possible, s'intentarà anar caminant o en bicicleta per reduir encara més l'impacte.

Ja per acabar, comentar que tot i que desconec el detall dels recursos utilitzats en les solucions existents, puc confirmar que al tractar-se de grans empreses l'impacte ambiental és molt més elevat que el causat en aquest projecte, ja que d'entrada consumeixen més recursos d'electricitat al disposar de més treballadors al càrrec.

3.4.4 *Dimensió social*

Per últim, és important valorar l'impacte social del projecte. En l'àmbit personal m'ha obert moltes portes, tals com l'oportunitat de treballar en un grup de recerca i d'ampliar els meus coneixements del grau i en una part d'una altra especialitat, com és adquirir coneixements de *sentiment analysis* i *machine learning*.

D'altra banda, aquest projecte de recerca toca un tema de molta actualitat, la Covid-19, pel que considero que aquest estudi pot ajudar socialment, ja que per xarxes socials la gent expressa realment com se sent en aquell moment. Si bé és cert que es tracta d'un projecte majoritàriament tecnològic, també consta d'una gran part social, ja que mesura l'impacte psicològic i la tendència afectiva d'un tema que afecta globalment com és el coronavirus.

Finalment, tal com vaig comentar a l'estat de l'art, no hi ha cap eina que abordi l'anàlisi de sentiments en espanyol de codi obert, però encara menys que tracti el coronavirus, ja que és un tema molt concret i recent.

3.5 Lleis i regulacions

Aquest projecte es tracta d'un projecte de recerca i no d'un producte comercialitzat. Per a realitzar determinades proves s'han utilitzat diferents *datasets* d'accés públic. Això s'ha de tenir en compte, ja que si s'utilitzés en qualsevol moment per a la realització d'un paper o si se'n fa menció en la memòria, s'haurà d'incloure la citació corresponent.

Per a la formació del propi *dataset* hem obtingut les dades a partir de l'API de Twitter. Aquestes dades son dades públiques i accessibles a tothom. Ara bé, si en un futur l'eina desenvolupada es volgués comercialitzar o aplicar en un cas concret en el qual s'utilitzessin dades confidencials ja sigui, per exemple, per una empresa o per a la realització d'un estudi, s'hauria de tenir en compte la Llei Orgànica de Protecció de Dades [30].

D'altra banda, aquest document i tot el desenvolupament del projecte estan regulats per la Llei de la Propietat Intel·lectual [31]. Respecte a la documentació, la universitat ha de respectar els drets d'autor i pot fer-ne difusió sempre que sigui per motius acadèmics i/o de recerca.

Per últim, el software del projecte estarà publicat a GitHub i serà de codi obert. Cada repositori estarà sota la llicència de Apache 2.0 [32].

4 Descripció tècnica

Aquesta secció està orientada a la descripció tècnica del sistema desenvolupat on, concretament, estarà inclosa l'arquitectura detallada, els patrons de disseny utilitzats i la descripció de cada un dels components que formen el sistema.

4.1 Arquitectura

La Figura 4.1 mostra l'arquitectura del sistema on el component principal implementat és l'*Orchestrator*. L'*Orchestrator* és el responsable de mantenir el flux d'informació entre tots els subsistemes del *framework* desenvolupats: el *Twitter Monitor*, el *Kafka server*, l'API de *Tweets Preprocessing* i l'API de *Sentiment Analysis*, que està connectada a l'API del traductor de Microsoft per a tasques de traducció i als models de *machine learning* desenvolupats.

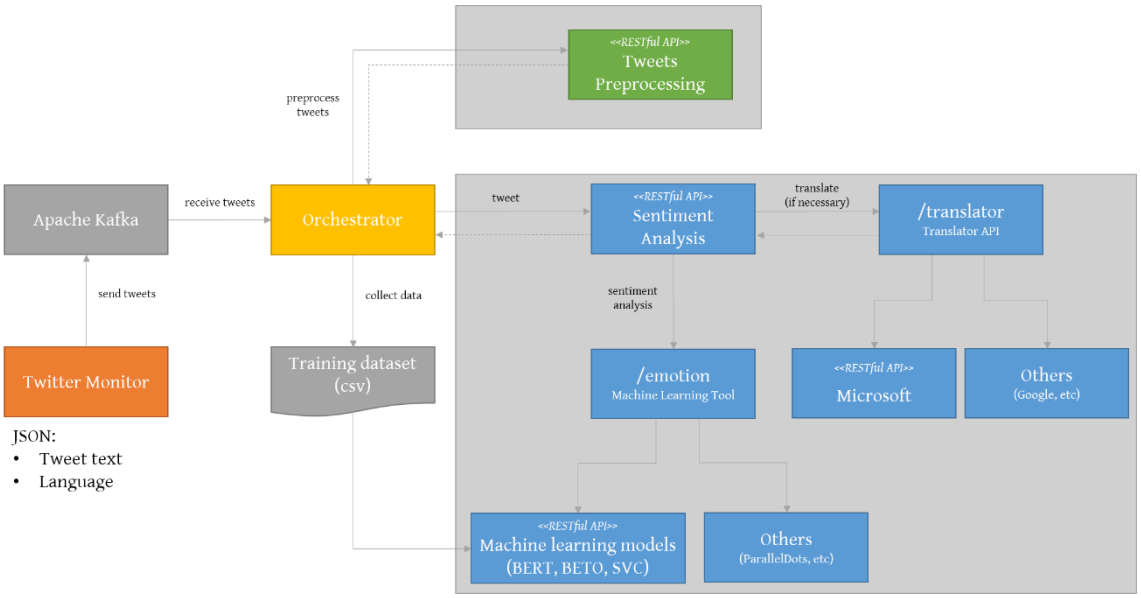

Figura 4.1 Arquitectura del sistema

El comportament del sistema és el següent: el *Twitter Monitor* rep els *tweets* en temps real mitjançant una crida a l'API de Twitter i aplicant els filtres corresponents, com per exemple, l'idioma dels *tweets*. En el moment que rep els *tweets*, el monitor els envia al Kafka server i aquests son rebuts per l'*Orchestrator* que fa una crida a l'API de *Tweets Preprocessing* per aplicar el preprocessat en cada *tweet* rebut. En aquest punt, l'API de preprocessat aplica algunes tècniques comunes de NLP com comprovar que una paraula existeix al diccionari de l'idioma desitjat o eliminant *hashtags* i mencions a usuaris innecessaris pel nostre objectiu, amb la finalitat d'obtenir un text més "nèt" i comprensible per a la màquina.

Una cop el sistema ha obtingut i processat tots els *tweets*, hi ha dues opcions possibles a seguir. La primera tracta de generar un fitxer CSV des de l'*Orquestrator* per entrenar els nostres models d'aprenentatge automàtic, i l'altra consisteix en aplicar l'anàlisi del sentiments mitjançant el model entrenat. En ambdós casos, el sistema comprova si els *tweets* estan en l'idioma especificat i, si no ho estan, fa una crida a l'API de traducció per traduir el text. L'API de traducció creada és una *API RESTful* genèrica que crida a una altra API de traducció existent al mercat. En el nostre cas, hem usat Microsoft Translator però, degut a la seva implementació, es pot canviar fàcilment per qualsevol altra eina de traducció.

L'última etapa correspon en fer una *request* a l'API de *Sentiment Analysis* per aplicar l'anàlisi de sentiments als *tweets* i obtenir la probabilitat de cada emoció. En aquest cas, i de manera similar a l'API del traductor, aquesta API crida a una altra API per classificar el text en emocions. Es pot fer crides tant a una API pública d'anàlisi de sentiments, com a les APIs creades amb els models d'aprenentatge automàtic desenvolupats. Aquests models d'aprenentatge automàtic son, per una banda, un model basat en BERT (de l'anglès *Bidirectional Encoder Representations from Transformers*) en la seva versió multi-idioma i BETO, una versió espanyola de BERT. D'altra banda, he creat un model basat en SVM (*Support Vector Machine*). Cadascun d'ells s'explicarà detalladament més endavant.

4.2 Descripció dels components

A continuació, detallarem cada un dels components del sistema. Cada un dels components ha estat implementat i desenvolupat cobrint les necessitats del projecte. L'únic subsistema ja existent, i que per tant no hem implementat, ha estat el *Apache Kafka*. Tots els repositoris de cada component estan disponibles a GitHub (veure Annex C).

4.2.1 *Twitter Monitor*

El monitor de Twitter és una eina desenvolupada per a realitzar tot el monitoratge dels *tweets* en temps real. Es tracta d'una aplicació Java on la seva funcionalitat principal es basa en la connexió amb el servidor de Kafka per a la recol·lecció de *tweets* en temps real a través de l'API de Twitter. Per fer-ho, hem seguit els passos següents:

1. Registrar un nou compte de Twitter Developer
2. Creació d'un projecte anomenat "COVID-19_Stream_v2" dins del portal de desenvolupador
3. Construir una *request* utilitzant un dels *endpoints* de l'API de Twitter, en aquest cas, utilitzarem "Twitter API v2".

Per construir la sol·licitud dels *tweets* a l'*endpoint* desitjat, que en aquest cas és el Filtered Stream ja que ens permet obtenir *tweets* filtrats segons els criteris que especifiquis (mitjançant les *rules*) en temps real [33], els passos a seguir son els següents:

1. Creació d'una *rule*

   Les *rules* estan construïdes a partir de diferents operadors utilitzats per coincidir amb determinats atributs dels *tweets*.
   Per aquest cas particular, hem utilitzat l'operador `context` amb una valor específic creat pels desenvolupadors de Twitter per obtenir tots els *tweets* relacionats amb la Covid-19 [34]. A més, hem aplicat un filtre per idioma en espanyol i dos més per evitar obtenir respostes de *tweets* i *retweets.*

   ```
   context:123.1220701888179359745 -is:retweet -is:reply lang:es
   ```

2. Afegir una *tag* a la *rule*

   Cada *rule* té associada una *tag* que la identifica, ja que el *Filtered Stream* permet aplicar varies *rules* a l'hora d'obtenir els *tweets.* En aquest sistema només tenim una *rule* etiquetada de la següent manera: `covid-19`.

3. Afegir la *rule* creada al *stream*

   Una vegada creada la *rule*, s'ha d'afegir al *stream*, juntament amb la seva etiqueta. Això estarà reflectit en una funció anomenada `setupRules`.

4. Autenticar la request

   El *Filtered Stream* requereix autenticació Bearer Token OAuth 2.0 proporcionada pel portal de desenvolupadors de Twitter. Aquest *token* s'ha d'enviar al crear les *rules* i al fer la connexió amb el *stream*.

5. Identificar i especificar els camps que volem que ens retorni

   El següent pas és identificar quins camps volem que ens retorni el *stream* i crear la *url* especificant els camps desitjats. En aquest cas concret, la *url* creada és la mostrada a continuació:

   ```
   https://api.twitter.com/2/tweets/search/stream?tweet.fields=id,create
   d_at,text,author_id,lang,geo,source&expansions=geo.place_id&place.fie
   lds=country_code,full_name
   ```

   Així doncs, els camps seleccionats a retornar son: identificador del *tweet*, data de creació, text, identificador de l'autor, idioma i localització (identificador, codi de país i nom complet), si la conté.

6. Connexió del stream

   Per últim, només queda realitzar la connexió amb el *stream* perquè ens retorni els *tweets* amb els camps especificats. Aquest procés estarà englobat en una funció anomenada `connectStream`.

El procediment complet està reflectit en el codi Java contingut en el GitHub del projecte.

4.2.2 *Kafka server*

El Kafka és un sistema especialitzat en el processament de fluxos de dades en temps real, per recopilar grans quantitats de dades o realitzar anàlisis en temps real (o ambdues coses). Es tracta d'un sistema de missatgeria basat en el mètode *publish-subscribe*, que consisteix en l'enviament de missatges entre processos, aplicacions i servidors [35]. En el nostre cas, és el monitor de Twitter qui envia els missatges (*producer*) i l'*Orchestrator* el que els rep (*consumer*).

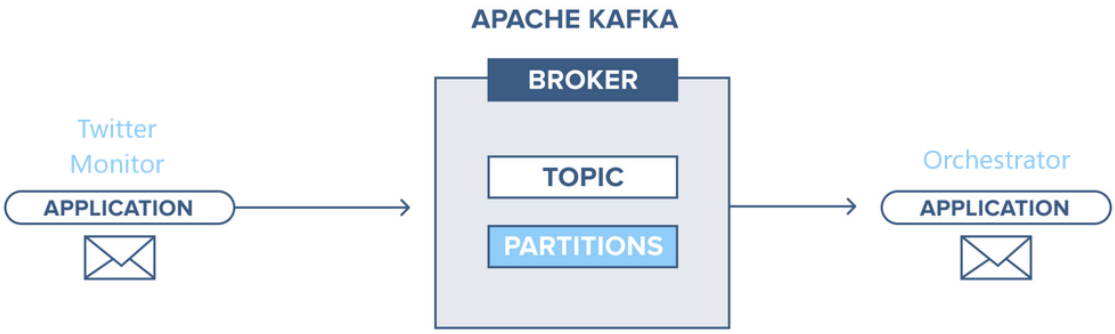

Figura 4.2 Enviament de missatges Kafka entre dues aplicacions [35]

Tal i com mostra la Figura 4.2, els missatges o *records* son enviats a un *topic* concret i son emmagatzemats durant un temps especificat. Tots els *topics* del Kafka estan continguts en el que s'anomena *broker* que, bàsicament, és qui gestiona totes les sol·licituds dels clients i manté les dades dins el clúster (conjunt de *brokers*). Un clúster pot contenir un o múltiples *brokers* (Figura 4.3). La gestió dels *brokers*, *topics* i usuaris dins el clúster, així com la gestió de l'estat del clúster, es realitzada pel Zookeeper.

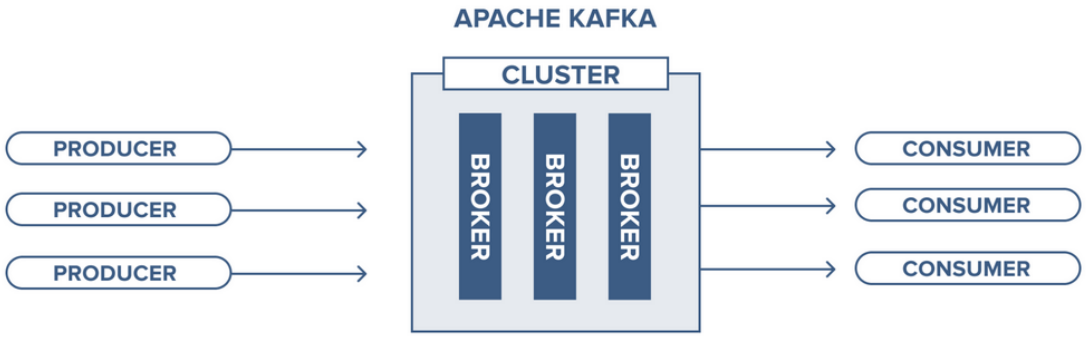

Figura 4.3 Kafka clúster [35]

Per tal de posar en marxa el sistema i que el monitor de Twitter pugui enviar correctament els *tweets* al Kafka, executat en local per aquest projecte, primerament s'ha d'haver configurat i inicialitzat l'entorn seguint els passos següents:

1. Obtenir Kafka

   El primer pas a realitzar, és instal·lar Kafka al nostre sistema. Per fer-ho, ens dirigim a la pàgina oficial de Kafka i descarreguem l'arxiu tgz i el descomprimim a la carpeta desitjada. Un cop descomprimit, ens dirigim a la carpeta a través de la interfície de comandes *Windows PowerShell*.

2. Inicialitzar l'entorn

   Per inicialitzar el Kafka, hem d'executar les següents comandes:

```
.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties

.\bin\windows\kafka-server-start.bat .\config\server.properties
```

3. Crear un *topic*

   Una vegada inicialitzats els serveis del Kafka, hem de procedir a la creació del *topic* per a la recol·lecció de missatges que, en aquest cas, es corresponen als *tweets*:

```
.\bin\windows\kafka-topics.bat --create --topic sentiment-analysis --
bootstrap-server localhost:9092
```

4. Rebre missatges

   Una vegada creat el *topic*, ja podem posar en marxa el monitor de Twitter perquè enviï els *tweets* al *topic*. Per veure els missatges enviats al Kafka pel monitor, hem d'executar la següent comanda:

```
.\bin\windows\kafka-console-consumer.bat           --bootstrap-server
localhost:9092 --topic sentiment-analysis --from-beginning
```

4.2.3 *Orchestrator*

L'*Orchestrator* és el motor central del sistema desenvolupat. Aquest consta de dues funcionalitats principals, les quals tenen en comú el flux que segueixen però difereixen en l'objectiu final. Ambdues funcionalitats segueixen els passos següents:

1. **Obtenció de *tweets*;** es recullen els *tweets* enviats al Kafka a través del monitor de Twitter.
2. **Traducció de text (si s'escau):** es comprova si el text està en l'idioma desitjat (espanyol en aquest cas). En cas que estigui en un altre idioma, l'*Orchestrator* fa una crida a l'API de *Sentiment Analysis* per traduir el text en qüestió.
3. **Preprocessament de *tweets*:** una vegada tots els *tweets* estan en espanyol, el sistema crida a l'API de *Tweets Preprocessing* per aplicar el preprocessat de *tweets*.

A partir d'aquí, cada tasca segueix un curs diferent segons la funcionalitat a realitzar en aquell moment:

- **Recol·lecció de dades** (veure Figura 4.4)**:** aplicat el preprocessat, el sistema genera un arxiu csv que conté el text original del *tweet* i el text processat. Aquest fitxer s'utilitzarà posteriorment per a realitzar l'entrenament dels models de *machine learning* desenvolupats.
- **Anàlisi de dades** (veure Figura 4.5)**:** aquesta segona tasca es centra en l'aplicació d'anàlisis de sentiments del text. L'aplicació genera un arxiu csv amb el text original del *tweet* en espanyol i les probabilitats resultants de l'API de *Sentiment Analysis* corresponents a cada emoció.

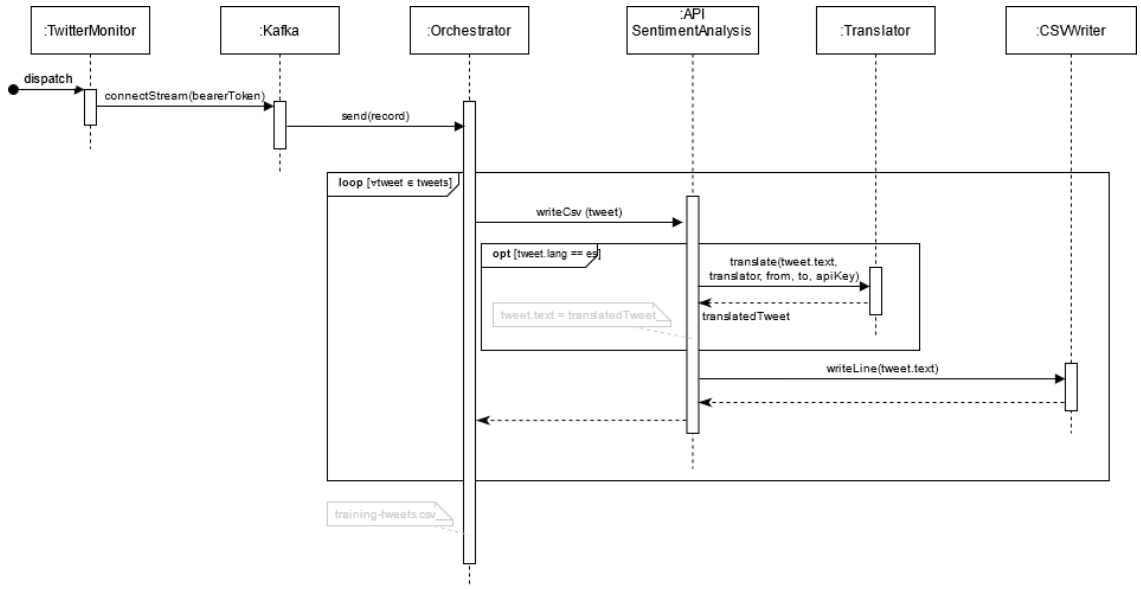

Figura 4.4 Diagrama de seqüència de recol·lecció de dades (Elaboració pròpia)

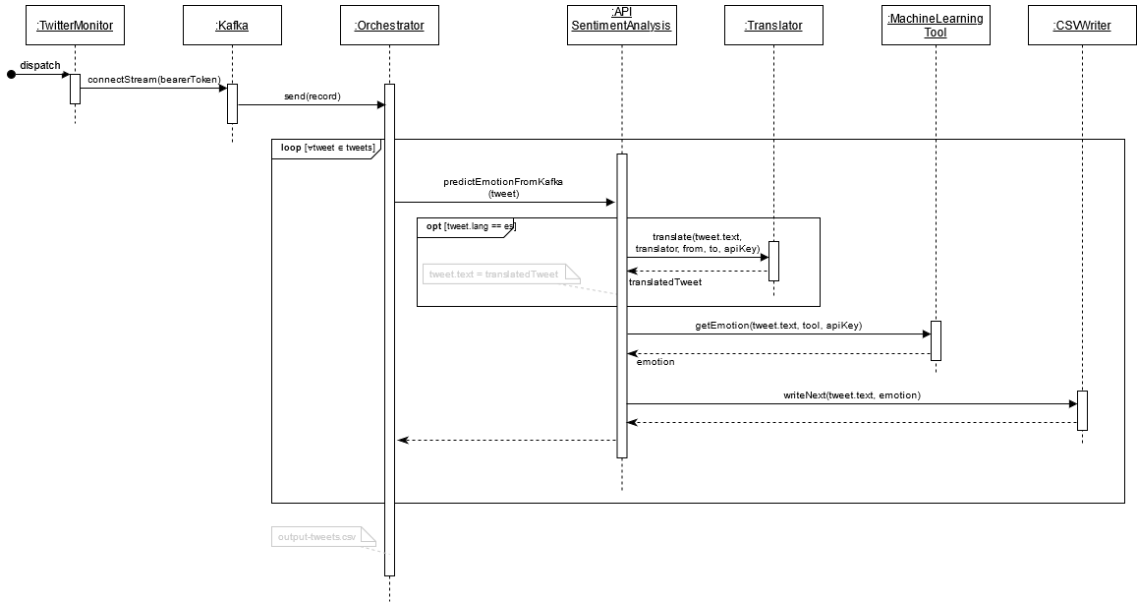

Figura 4.5 Diagrama de seqüència d'anàlisi de dades (Elaboració pròpia)

4.2.4 API *Sentiment Analysis*

L'API de *Sentiment Analysis* és un punt clau en el sistema. Es tracta d'una API creada per connectar amb diferents serveis, tant externs com d'elaboració pròpia, de traducció i de anàlisis de sentiments. Aquestes dues funcionalitats treballen de la següent manera (veure Annex A1):

- **Traducció:** quan un usuari fa una petició de traducció, el sistema fa una petició a l'eina especificada i retorna el resultat a l'usuari.
- **Anàlisis de sentiments:** un usuari fa una petició seleccionant l'eina de *machine learning* que desitja utilitzar, tant eina externa com qualsevol dels models de *machine learning*

pròpiament desenvolupats. El sistema retorna un JSON amb les probabilitats de cada emoció sobre el text analitzat.

Aquesta API està dissenyada aplicant dos patrons de disseny: MVC (Model – View – Controller) i patró factoria. El primer patró està relacionat amb l'estructura del sistema i ajuda a mantenir una separació entre les diferents capes de l'aplicació. Addicionalment, amb l'aplicació del patró factoria oferim un desacoblament del sistema que ens permet afegir nous mètodes de traducció i models de *machine learning* de manera ràpida i aplicant poques modificacions al codi existent.

Patrons de disseny

Els patrons de disseny utilitzats per aquesta API son els següents:

Patró factoria

És un patró de disseny creacional orientat a mecanismes de creació d'objectes per incrementar la flexibilitat i reutilització del codi [36] en el qual tenim una superclasse, que en aquest cas és una interfície (però podria ser classe abstracte o una classe de Java), amb múltiples subclasses [37]. En el cas d'aquesta API, he aplicat aquest patró per a la creació dels objectes del model de *machine learning* per a la classificació d'emocions, tal i com es reflexa la Figura 4.6.

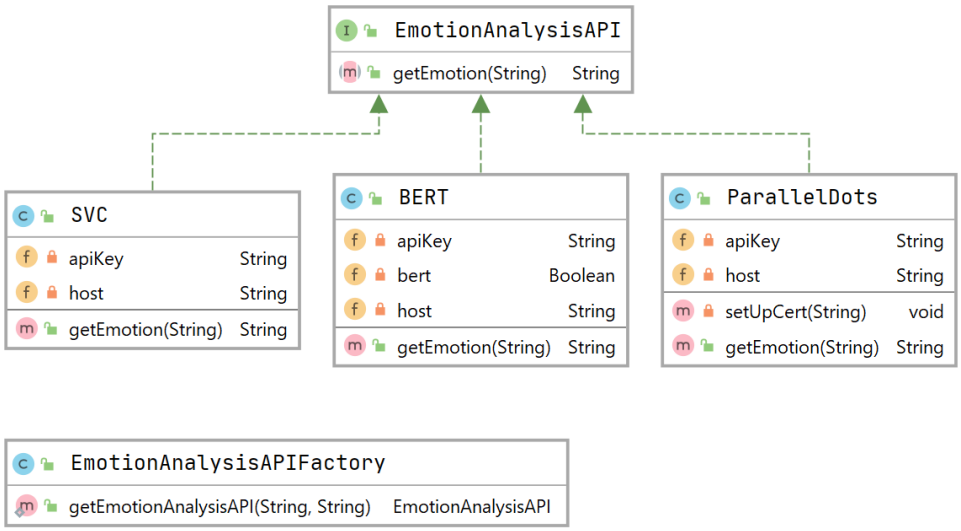

Figura 4.6 Patró factoria EmotionAnalysisAPI (Elaboració pròpia)

De manera similar, he aplicat el patró per a la creació dels objectes per realitzar les tasques de traducció, tal i com es mostra a la Figura 4.7.

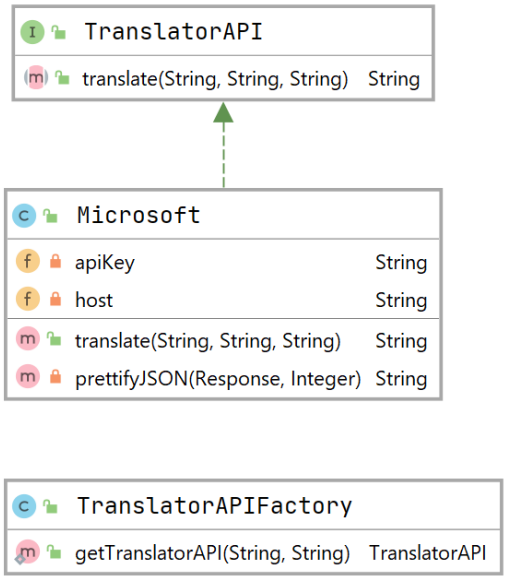

Figura 4.7 Patró factoria TranslatorAPI (Elaboració pròpia)

Patró MVC (Model – Vista – Controlador)

Patró de disseny utilitzat per separar la lògica de les diferents capes en unitats independents de la nostra aplicació . Tenim tres components que constitueixen aquest patró [38]:

- **Model:** component central del patró que conté l'estructura de dades. Per aquesta API no disposarem de base de dades ja que el que fa aquesta API es cridar a altres APIs (de traducció i de *sentiment analysis*). L'usuari introduirà el text que necessita traduir o analitzar.
- **Vista:** interfície gràfica d'usuari que conté les funcionalitats per interactuar directament amb l'usuari. En aquest cas tenim Swagger com a UI (de l' anglès, *User Interface*).
- **Controlador:** connecta la vista amb el model convertint les dades d'entrada de l'usuari en dades que aquest sol·licita. Per aquest cas particular, per exemple, l'usuari envia un fragment de text i el controlador s'encarrega d'enviar les dades al model que crida a diferents APIs que retornen l'anàlisi d'emocions corresponent.

4.2.5 API *Tweets preprocessing*

L'API de *Tweets Preprocessing*, tal i com el nom indica, és una API desenvolupada pel preprocessat de *tweets* aplicant diverses tècniques de processament del llenguatge natural per "netejar" el text. Alguns exemples de *tweets* després d'aplicar el preprocessat es mostren a la Taula 4.1. Aquests mètodes aplicats i tècniques de NLP son els següents:

- **Eliminar URLs:** son molts els *tweets* que conten URLs que amplien el contingut del text. A l'hora d'aplicar *sentiment analysis* aquests enllaços no es visiten i només causen "soroll" al text. Per aquest motiu convé eliminar-les.
- **Eliminar mencions:** és usual que a Twitter els usuaris mencionin a altres usuaris. Aquestes mencions no aporten informació i, per tant, poden ser eliminades.
- **Eliminar números:** pel context de *sentiment analysis* els números no tenen rellevància en la majoria de casos. Si que és cert que influeix de manera quantitativa, per exemple,

si es parla del nombre de contagis per COVID-19. En aquests casos però, com el nombre sempre es variant, a cada número se li aplicaria un *token* diferent i crearia més confusió que claredat. Així doncs, eliminarem tots els números que apareixen al text.

- **Reemplaçar emoticones:** les emoticones son molt utilitzades i importants en l'anàlisi de sentiments. És molt probable que el text sigui neutral però hi hagi una emoticona que expressi felicitat i ajudi a reconèixer que l'usuari està content. Per aquest motiu, hem reemplaçat les emoticones per la emoció que expressen. Per exemple:
  - :) → *happy*
  - :(→ *sad*
- **Reemplaçar *emojis*:** de la mateixa manera que les emoticones, alguns *emojis* ens aporten sentiments al text. Tots aquells que aporten informació els hem substituït per text i els que no aporten valor, els hem eliminat.
- **Reemplaçar abreviacions:** en tots els idiomes de manera col·loquial s'utilitzen expressions abreviades, sobretot en xarxes socials. En el cas de l'espanyol hem agafat una llista de les més comunes i les hem substituït per la paraula corresponent. Per exemple, algunes de les més típiques son: *q (que), bn (bien), vdd (verdad), tqm (te quiero mucho)* o *pti (para tu información)*.
- **Reemplaçar riures:** el cas del riures en xarxes socials és quelcom complicat de generalitzar, ja que hi ha moltes variants. En aquest cas, he agafat totes les paraules que comencen per 'ja', 'je', 'ji', 'jo' i les he substituït per 'jajaja' (després de comprovar que el terme no existeix al diccionari per evitar reemplaçar paraules existents), per obtenir una generalització del riure.
- **Eliminar signes de puntuació:** per analitzar els sentiments no és necessari mantenir els signes de puntuació (punts, comes, interrogants, hashtags, etc.) ja que no donen valor a les emocions per si soles. En alguns casos, alguns signes de puntuació (per exemple, signes d'exclamació) poden emfatitzar la emoció del text que expressen. Degut a que no hi ha una generalització normativa perquè la màquina detecti aquests casos de manera clara, hem decidit eliminar-los per a no crear més confusió a la màquina.
- **Eliminar caràcters repetits:** en xarxes socials és comú que no es respectin les normes ortogràfiques i que els usuaris escriguin allargant les paraules repetint caràcters per emfatitzar, especialment al final de les paraules (per exemple, *holaa* o *graciaas!*). Aquests caràcters addicionals es poden eliminar.
- **Lematitzar:** una tècnica molt comuna en NLP és la lematització. Bàsicament consisteix en transformar totes les conjugacions verbals que apareixen al text en l'infinitiu del verb en qüestió. Per exemple, de la oració '*Hoy jugaremos a pelota*' el verb *jugaremos* es quedaria en *jugar*.
- **Eliminar *stopwords*:** una altre tècnica molt utilitzada és la d'esborrar *stopwords*, és a dir, paraules que no aporten cap mena d'informació útil. En Python hi ha llibreries que conten un llistat de totes les comunes en cada idioma. Per exemple, en espanyol algunes d'elles son *'del', 'ella'* o *'entonces'*. En la següent secció parlarem en detall de totes les llibreries i tecnologies utilitzades per a cada subsistema.
- **Eliminar espais en blanc:** l'últim pas a realitzar és eliminar els espais en blanc extres que pugui contenir el *tweet*, o que hagin pogut quedar al reemplaçar o eliminar paraules en els passos previs.

El funcionament de l'API comença amb una *request* de l'usuari on envia el text original del *tweet*. El sistema aplica el processament del text explicat anteriorment i retorna el text processat.

Tweet	Tweet Preprocessat
@AntonioMautor Mi idea primera era hacerlo, pero como dijeron que posiblemente no había vacunas para todo el mundo, tengo 45 y como si el barco se hundiera, primero grupos de riesgo ... los mayores de 65, etc etc ...\	idea primero ser hacer posiblemente no haber vacuna mundo tener si barco hundir primero grupo riesgo mayor etcétera etcétera
"\"El mejor argumento que he leído hasta ahora, desde luego 😂 https://t.co/JzjTuea53D\""	mejor argumento haber leer luego felicidad
"\"Mil gracias por compartir 😭 https://t.co/TbsakUc5ky\""	gracias compartir felicidad

Taula 4.1 Exemple de preprocessament de tweets (Elaboració pròpia)

4.2.6 APIs Models *machine learning*

Les últimes APIs desenvolupades son les corresponents als models de *machine learning*. Les tres APIs tenen el mateix funcionament. L'usuari fa una petició d'anàlisis del text a través de l'*endpoint* corresponent. A partir d'aquí, el model aplica l'anàlisi del text utilitzant el model indicat prèviament entrenat i retorna un JSON amb cadascuna de les emocions i la seva probabilitat.

Pel càlcul de les probabilitats de cada emoció, s'aplicarà la funció *softmax*, una funció matemàtica que agafa el vector d'entrada de nombres reals i ho normalitza en una distribució proporcional als exponencials dels nombres d'entrada. Els nombres d'entrada poden ser negatius o majors a 1 però, després d'aplicar el *softmax*, la suma de cadascun dels elements sumarà 1. La funció aplicada és la següent:

$$softmax\left(Z_j\right) = \frac{e^{Z_j}}{\sum_{k=1}^{K} e^{Z_k}} \; for\; j = 1, \dots, K$$

En el cas de les xarxes neuronals (veure Figura 4.8), es fa una mapeig de les dades de sortida que no estan normalitzades a la distribució de probabilitats de les classes de sortida (emocions) de les capes finals del classificador [39].

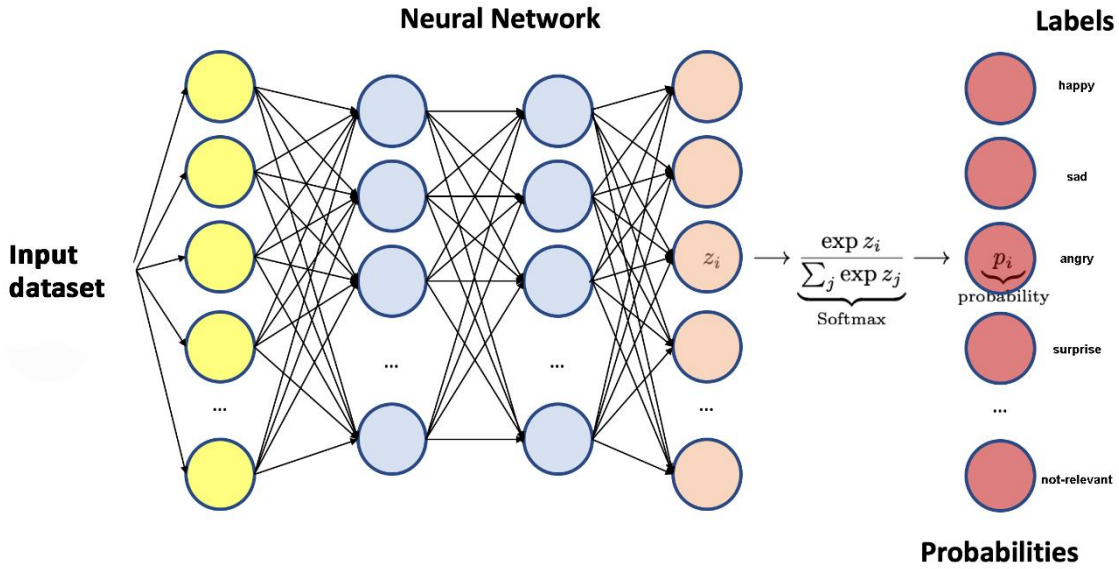

Figura 4.8 Softmax aplicat a una xarxa neuronal [40]

API BERT

BERT és una tècnica de *machine learning* desenvolupada per Google basada en l'arquitectura *transformer* (veure Figura 4.9). El model *transformer*, un model de *deep learning* relativament nou, és capaç de lidiar amb problemes de processament del llenguatge natural. BERT és un *transformer* [41] bidireccional, tal i com el seu nom indica, pre-entrenat amb un gran corpus format pel Toronto Book Corpus i Wikipedia [42]. Mitjançant capes d'atenció (de l'anglès *attention masks*), el model codifica cada paraula d'una oració durant la fase d'entrenament i ha de predir les paraules prèviament emmascarades (un 15% de cada oració) utilitzant NSP (de l'anglès *Next Sentence Prediction*) per predir i entendre el context del text, així com la connexió entre frases [43].

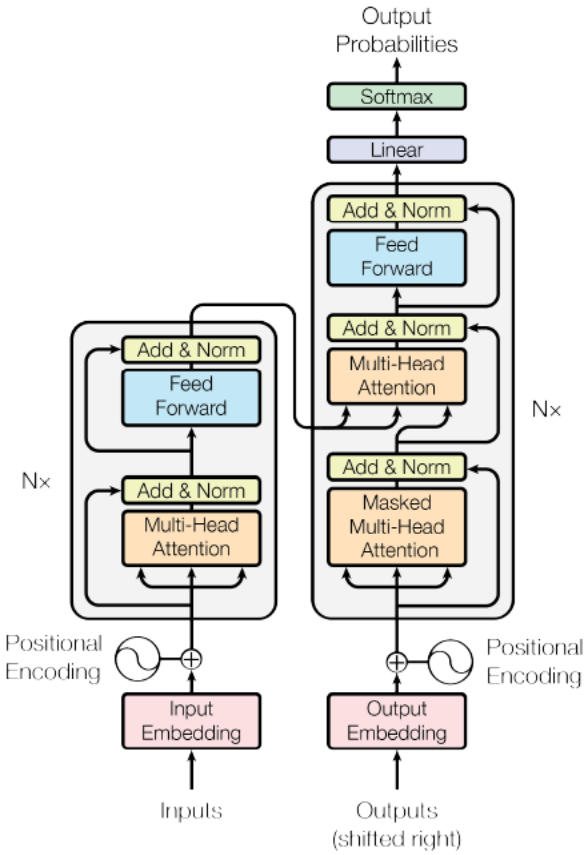

Figura 4.9 Arquitectura del model transformer [41]

Per aquesta API tenim dues variants que escollirà l'usuari a l'hora de fer la *request*. Aquestes variants son les següents:

- **BERT *Multilanguage*:** és un model BERT entrenat en 104 idiomes diferents, incloent l'espanyol, amb un gran *dataset* extret de Wikipedia [42]. L'*accuracy* extreta per aquesta variant en la fase de pre-entrenament és força bona en la majoria de casos, tal i com mostra la taula següent:

System	English	Chinese	Spanish	German	Arabic	Urdu
BERT - Translate Train Cased	81.9	76.6	77.8	75.9	70.7	61.6

Taula 4.2 5 Accuracy extreta del GitHub original de BERT Multilanguage [44]

- **BETO:** aquesta segona variant és especifica d'idioma, ja que ha estat entrenada amb un *dataset* de mida similar al BERT però amb text únicament en espanyol. Es va realitzar un estudi on els resultats del *benchmark* obtinguts van ser els següents:

Task	BETO-cased	BETO-uncased	Best Multilingual BERT
POS	98.97	98.44	97.10 [2]
NER-C	88.43	82.67	87.38 [2]

MLDoc	95.60	96.12	95.70 [2]
PAWS-X	89.05	89.55	90.70 [8]
XNLI	82.01	80.15	78.50 [2]

Taula 4.3 Resultats BETO vs BERT Multilanguage [45]

API SVC

SVC (de l'anglès *Support Vector Machine*) és un conjunt de models de *machine learning* supervisat associat a algoritmes enfocats a diferents tasques d'aprenentatge com la classificació. Per resoldre problemes de classificació, he utilitzat SVC (de l'anglès *Support Vector Classifier*) [46] el qual està basat en la llibreria LIBSVM [47], una llibreria per a tasques SVM disponible en diversos llenguatges de programació. En aquest cas, hem aplicat el cas típic de classificació multi-classe que implica una primera fase d'entrenament per a l'obtenció del model i una fase d'avaluació mitjançant la predicció de dades a partir del model entrenat prèviament [48].

4.3 Obtenció del *dataset*

Per a l'obtenció del *dataset*, hem recollit un total de 3346 *tweets* a través del monitor de Twitter creat. A partir d'aquí, hem etiquetat manualment cada *tweet* mitjançant un exhaustiu procés entre dues persones que, en aquest cas, hem estat el Marc Oriol i jo mateixa. Inicialment, fèiem una reunió setmanal per arribar a un acord per a cada emoció en la qual diferíem però finalment, després de mesurar l'*inter-rater agreement* per veure la concordança, vam observar que era més fiable mantenir només els *tweets* on coincidíem des d'un primer moment. Les emocions seleccionades primerament per etiquetar els *tweets* van ser les següents: *angry*, *disgust*, *happy*, *sad*, *surprise* i *not-relevant*. Després d'unes setmanes, vam eliminar el *disgust* ja que gairebé no hi havia cap *tweet* amb aquesta emoció.

Una vegada eliminats els *tweets* que no coincidíem i els *tweets* nuls després d'aplicar el preprocessat, hem obtingut un conjunt de 2164 *tweets*, els quals segueixen la distribució següent:

Emoció	Nombre de mostres
angry	628
happy	289
not-relevant	787
sad	436
surprise	24

Taula 4.4 Distribució de tweets (Elaboració pròpia)

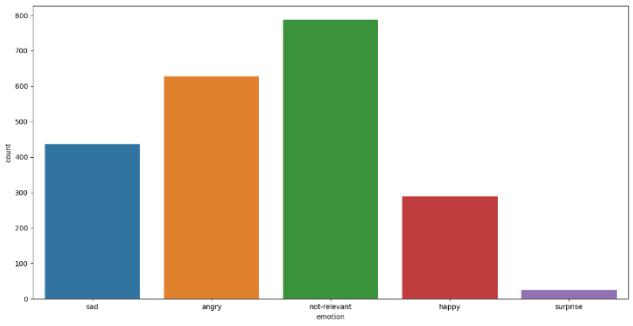

Figura 4.10 Distribució de tweets (Elaboració pròpia)

5 Entorn de desenvolupament

En aquest apartat descriuré quines tecnologies i eines he utilitzat per a la realització del projecte, tenint en compte que la integritat del sistema s'ha realitzat amb els llenguatges de programació Python i Java.

5.1 Tecnologies

Les eines software utilitzades per a la realització del projecte son les següents:

- **IDEs i *frameworks***
  - **Java 8** [49]**:**
    - **IntelliJ** [50]**:** entorn de desenvolupament utilitzat per al desenvolupament dels subsistemes desenvolupats en Java: *Twitter Monitor, Orchestrator, API Sentiment Analysis.*
    - **SpringBoot** [51]**:** *framework* de codi obert per facilitar el desenvolupament d'aplicacions Java.
    - **Maven** [52]**:** software utilitzat per a la gestió i construcció de projectes Java que utilitza un POM (de l'anglès *Project Object Model*) per a descriure el software, els components utilitzats i les seves dependències.
    - **Javadoc** [53]**:** utilitat de Oracle per a la creació d'una pàgina HTML amb la documentació del projecte a partir del codi font.
  - **Python (v3.9):**
    - **Visual Studio Code** [54]**:** editor de codi optimitzat pel desenvolupament d'aplicacions web. En aquest cas, pel desenvolupament de les APIs dels models de *machine learning* i la del preprocessament de *tweets*.
    - **Flask** [55]**:** *framework* minimalista per a Python utilitzat per a la ràpida creació d'aplicacions amb les mínimes línies de codi.
- **API *requests* i documentació:**
  - **Postman** [56]**:** plataforma col·laborativa pel desenvolupament de APIs. En aquest cas, la utilitzarem per a la validació de les APIs creades ja que permet fer sol·licituds HTTP.
  - **Swagger** [57]**:** eina de codi obert per a dissenyar, construir, documentar i utilitzar serveis RESTful, en particular al cas, l'API de *Sentiment Analysis* Java.

5.2 Llibreries utilitzades

A continuació llistarem les llibreries utilitzades per a cada un dels llenguatges utilitzats i dels subsistemes desenvolupats.

5.2.1 *Java*

Per a la construcció dels subsistemes desenvolupats en Java (Orchestrator, Twitter Monitor, API Sentiment Analysis) hem necessitat les llibreries llistades a continuació:

- **Spring Fox** [58]**:** conjunt de llibreries Java per a l'automatització de les especificacions de les APIs de manera comprensible per als humans i la màquina. Springfox examina l'aplicació en temps d'execució basant-se en les configuracions Spring, l'estructura de classes i les anotacions Java.
- **Okhttp** [59]**:** permet realitzar operacions HTTP de manera senzilla i eficient en entorns Java i Android.
- **Json Simple** [60]**:** facilita l'ús de objectes JSON en la codificació i descodificació d'aquests dins el codi Java.
- **Jakarta Activation** [61]**:** extensió de la plataforma Java per treure profit de serveis estàndards com determinar el tipus de dades o descobrir operacions disponibles. Ens servirà per poder accedir a javax [62] que permetrà l'ús de SSL (de l'anglès Secure Socket Layer) per a la connexió amb ParallelDots.
- **Kafka Clients** [63]**:** proveeix accés al Apache Kafka i el processament de streams a alt nivell. El codi del Producer estarà contingut al Twitter Monitor i el codi del Consumer al Orchestrator.
- **gson** [64]**:** usada per a convertir objectes Java a la seva representació en JSON, i a la inversa. Serà útil per a convertir els *tweets* rebuts com a *strings* a objectes JSON.
- **opencsv** [65]**:** llibreria Java que permet treballar fàcilment amb arxius CSV. L'utilitzarem per a generar l'arxiu per a l'entrenament del model a partir dels *tweets* rebuts a través del Kafka.
- **SLF4J** (de l'anglès *Simple Logging Facade for Java*) [66]**:** simple abstracció per a diferents *frameworks* de *logging* que permet a l'usuari final entrar en el *framework* desitjat en temps de desenvolupament. En farem ús al accedir al Kafka server.
- **FasterXML Jackson Core** [67]**:** útil per al processament de JSON. Les principals classes utilitzades son JsonFactory per a construir el JSON *parser* i en JsonGenerator per a la generació d'instàncies.
- **emoji-java** [68]**:** llibreria per a facilitar l'ús d'*emojis* en les aplicacions Java.
- **Apache HTTP Client** [69]**:** proporciona una llibreria per a la implementació del client amb els estàndards i recomanacions HTTP més recents de manera eficient, actualitzada i plena de funcionalitats.
- **org.json :** exposa com fer l'intercanvi de documents JSON a objectes Java i com generar nous documents JSON a partir de classes Java.

La Taula 5.1 exposa breument les llibreries utilitzades per a cada sistema i la versió usada en cada component.

Llibreria	Versió	Sistemes
Spring Fox	2.9.2	API Sentiment Analysis
JSON simple	1.1.1	API Sentiment Analysis, Orchestrator, Twitter Monitor
okhttp	4.7.2	API Sentiment Analysis, Orchestrator, Twitter Monitor
Jakarta Activation	1.2.2	API Sentiment Analysis
Kafka Clients	2.6.0	Orchestrator, Twitter Monitor
gson	2.8.6	Orchestrator, Twitter Monitor
opencsv	3.3	Orchestrator
SLF4J	1.7.30	Orchestrator, Twitter Monitor
FasterXML Jackson Core	2.4.0	Orchestrator, Twitter Monitor
emoji-java	5.1.1	Orchestrator

Apache HTTP Client	4.5.12	Twitter Monitor
org.json	20200518	Twitter Monitor

Taula 5.1 Llibreries Java (Elaboració pròpia)

5.2.2 Python

Per al desenvolupament del codi Python, he fet ús de varies llibreries, totes elles *open-source*, que ajuden i faciliten el desenvolupament del codi. Aquestes llibreries son les següents:

- **emojis** [70]**:** conté una base de dades de *emojis* basada en la llibreria gemoji [71]. Aquesta llibreria la utilitzarem per obtenir tots els *emojis* en un llistat continguts en el text Python. Posteriorment, comprovarem que estiguin a l'arxiu CSV que conté els *emojis* traduïts a text i, si s'escau, substituirem l'*emoji* per l'emoció corresponent en text. En cas contrari, eliminarem l'*emoji*.
- **Jupiter Widgets (ipywidgets)** [72]**:** proporciona elements HTML interactius per a *Jupiter notebooks* i *IPython kernel*. En aquest cas, ens permet visualitzar d'una forma atractiva l'avenç en el procés del *training* dels models BERT i BETO mitjançant una barra de progrés.
- **matplotlib** [73]**:** permet crear visualitzacions estàtiques, animades i interactives en Python, com ara gràfiques. Per aquest projecte ens servirà per visualitzar la distribució dels *tweets* en gràfic de barres i les matrius de confusió.
- **nltk** [74]**:** plataforma que permet desenvolupar programes Python per lidiar amb dades del llenguatge natural. Conté uns 50 recursos lèxics i de corpus a més de un conjunt de llibreries per al processament del llenguatge natural. L'utilitzarem principalment per a obtenir les *stopwords* en espanyol i així poder eliminar-les.
- **NumPy** [75]**:** proveeix objectes vectorials multidimensionals, objectes derivats (com vectors i matrius emmascarats) i un conjunt de rutines per realitzar ràpides operacions amb vectors com, per exemple, operacions matemàtiques, lògiques o d'ordenació. En farem un ús recurrent en varies ocasions com trobar un nombre màxim, concatenar vectors, calcular rangs, entre d'altres.
- **pandas** [76]**:** permet utilitzar fàcilment i amb un alt rendiment estructures de dades i eines per a l'anàlisi d'aquestes. Ens serà útil per llegir el fitxer CSV i estructurar les dades obtingudes del fitxer.
- **Pyenchant** [77]**:** llibreria que engloba un conjunt de llibreries i programes ortogràfics. Permet l'accés a llibreries ortogràfiques incloent algunes especialitzades en un idioma en particular, com és l'espanyol en aquest cas. Ens permetrà comprovar si una paraula existeix o no al diccionari espanyol.
- **PyTorch** [78]**:** utilitzada per a tasques d'aprenentatge automàtic basada en la llibreria Torch [79] i utilitzada per a aplicacions de visió artificial i processament del llenguatge natural, entre d'altres. L'utilitzarem en la fase d'entrenament i validació dels model BERT i BETO.
- **scikit-learn** [80]**:** eines simples i eficients per l'anàlisi predictiu de dades i tasques d'aprenentatge automàtic. Està dissenyada per operar amb les llibreries NumPy, SciPy i matplotlib. Inclou diversos algoritmes de classificació, com serà el cas del model SVC, regressió i anàlisis, incloent algoritmes com Support Vector Machines, Random Forest o K-means.

- **Scipy** [81]**:** conté eficients rutines numèriques fàcils d'utilitzar per a l'usuari com integració numèrica, interpolació, optimització, àlgebra lineal i estadístiques. En farem ús a l'API de BERT per a l'aplicació de la funció *softmax*.
- **seaborn** [82]**:** llibreria per a la visualització de dades basada en matplotlib. Proveeix una interfície d'alt nivell per mostrar gràfiques estadístiques de manera atractiva per a l'usuari. L'utilitzarem per a comptar el nombre de mostres de cada emoció per posteriorment mostrar la distribució amb matplotlib.
- **stanza** [83]**:** col·lecció d'eines precises i eficients per a diferents idiomes, com eines d'anàlisis sintàctic i de reconeixement d'entitats. Proporciona model de processament del llenguatge naturals per a l'idioma escollit. Ens servirà per a la lematització del text.
- **Transformers** [84]**:** proporciona milers de model pre-entrenats per dur a terme tasques en text, com ara classificació, extracció d'informació o traducció, en més de 100 idiomes diferents. Proporciona arquitectures per a la comprensió (NLU, de l'angles *Natural Language Understanding*) i generació (NLG, de l'anglès *Natural Language Generation*) del llenguatge natural. L'utilitzarem per obtenir els models pre-entrenats BERT Multilingual i BETO.

La Taula 5.2, resumeix les llibreries utilitzades amb la seva versió corresponent i quins sistemes dels desenvolupats en Python (API Tweets Preprocessing, API BERT i API SVC) l'estan utilitzant.

Llibreria		Versió	Sistemes
emojis		0.6.0	API Tweets Preprocessing
Jupiter Widgets (ipywidgets)		7.6.3	API BERT
matplotlib		3.4.1	API BERT, API SVC
nltk		3.5	API Tweets Preprocessing
numpy		1.20.2	API Tweets Preprocessing, API SVC
pandas		1.2.3	API Tweets Preprocessing, API BERT, API SVC
pyenchant		3.2.0	API Tweets Preprocessing
PyTorch	torch	1.8.1+cpu	API BERT
	torchaudio	0.8.1	
	torchvideo	0.9.1+cpu	
scikit-learn		0.24.1	API Tweets Preprocessing, API BERT, API SVC
scipy		1.6.2	API BERT
seaborn		0.11.1	API BERT
stanza		1.2	API Tweets Preprocessing
transformers		4.5.0	API BERT

Taula 5.2 Llibreries Python (Elaboració pròpia)

6 Validació

En aquesta secció, exposaré els mètodes utilitzats per a la validació general del correcte funcionament del sistema. Per fer-ho, hem validat cadascun dels components per separat. Primerament, hem comprovat que el monitor de Twitter rebés correctament els *tweets* i els enviés correctament al Kafka que hem executat localment. Un cop l'enviament dels missatges eren correctament processats pel Kafka, vam assegurar que eren rebuts per l'Orchestrator. A partir d'aquí, el Orchestrator tenia dos funcionalitats que ja hem explicat anteriorment:

- Generar un arxiu CSV amb els *tweets* preprocessats i traduïts a l'espanyol, si era necessari.
- Aplicar *sentiment analysis* cridant a l'API creada per això.

En els següents dos apartats, explicarem com hem validat cadascuna de les APIs desenvolupades i el procediment seguit per a la millora dels models de *machine learning*.

6.1 Validació APIs

Per a la comprovació del correcte funcionament de cadascuna de les APIs, he utilitzat el software Postman que permet enviar *requests* HTTP i obtenir les dades de resposta corresponents. Vam crear una col·lecció amb cadascuna de les crides per a cada API i una vegada executada la crida, comprovava la sortida, analitzant si obtenia el resultat esperat. Si hi havia un error, consultava la consola del Visual Studio Code on estava executant l'API localment i així reiteradament fins aconseguir el resultat esperat.

A continuació, mostrem un exemple on, a partir d'un text en concret, s'obtenen els diferents resultats (per informació addicional de configuració dels paràmetres veure Annex A):

- **Text:**
  - **Traducció:**

    "After more than one week at home in quarantine, I'm free 😃😃"

  - **Sentiment analysis:**

    "A ver si se termina ya la pandemia, estoy harta del covid 😭"

- **Resultats:**
  - API Sentiment Analysis (veure A1):
    - Traduir text

```
{
      "translation": "Después de más de una semana en casa
      en cuarentena, estoy libre 😃😃 ",
      "from": "en",
      "to": "es"
}
```

- Aplicar sentiment analysis
  - ParallelDots


```
{
    "emotion": {
        "Angry": 0.280472462,
        "Happy": 0.1648395387,
        "Bored": 0.0392264591,
        "Fear": 0.1757421422,
        "Sad": 0.1837037203,
        "Excited": 0.1560156777
    }
}
```


  - Models machine learning (veure APIs abaix)

- API Tweets Preprocessing (veure A2)


```
{
    "translation": "Después de más de una semana en casa en
cuarentena, estoy libre 😀😀 ",
    "from": "en",
    "to": "es"
}
```


- API BERT (veure A3)
  - BERT Multilingual


```
{
    "emotions": {
        "angry": 0.3091,
        "happy": 0.11399,
        "not-relevant": 0.42155,
        "sad": 0.14403,
        "surprise": 0.01132
    }
}
```


  - BETO


```
{
    "emotions": {
        "angry": 0.71442,
        "happy": 0.02399,
        "not-relevant": 0.09899,
        "sad": 0.14578,
        "surprise": 0.01682
    }
}
```

- API SVC (veure A4)


```
{
    "emotion": {
        "angry": 0.00192,
        "happy": 1e-05,
        "not-relevant": 0.10221,
        "sad": 0.88774,
        "surprise": 0.00812
    }
}
```


6.2 Validació models *Machine Learning*

Per a la validació dels models de *machine learning*, hem realitzat el que s'anomena *fine-tuning* de híper-paràmetres que, bàsicament, consisteix en ajustar alguns paràmetres del model per tal d'obtenir millors resultats.

En el cas del BERT i el BETO, els híper-paràmetres que hem ajustat son els següents:

- **Learning rate:** Híper-paràmetre que controla quant canvia el model envers a l'error estimat cada vegada que s'actualitzen els pesos del model. Escollir un valor òptim és difícil ja que un valor molt baix por provocar molta lentitud en el procés de *training* i un valor molt alt pot fer que el model divergeixi en comptes de convergir en la solució (veure Figura 6.1) [85].

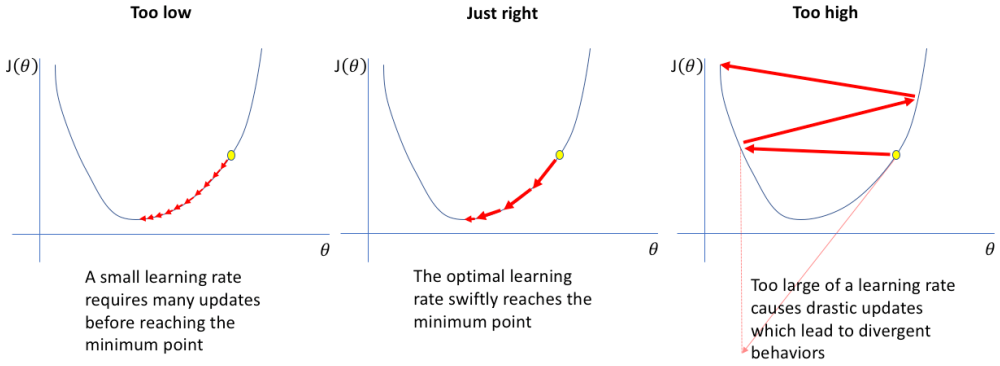

Figura 6.1 Ajustant l'híper-paràmetre "learning rate" [86]

- **Epsilon**: Nombre de valor molt baix utilitzat per prevenir qualsevol divisió per zero en la implementació [87].
- **Number of epochs:** Híper-paràmetre que indica el nombre de vegades que el model visita íntegrament el corpus de *training*. En aquest cas, l'ajustem per controlar el *weight decay*, que segueix la següent fórmula:

$$weight\ decay = \frac{learning\ rate}{number\ of\ epochs}$$

Hem de tenir cura a l'hora d'escollir el nombre d'èpoques ja que el *weight decay* és un paràmetre que s'utilitza per a prevenir l'*overfitting* i mantenint un baix valor és pot evitar l'esclat de gradients.

Pel cas del *Support Vector Classifier*, els paràmetres a ajustar son diferents als que tenim pel BERT i BETO. En aquest cas, utilitzarem RandomSearch, de la llibreria sklearn, que comprova de manera aleatòria diferents combinacions dels paràmetres especificats amb els valors establerts. Els paràmetres que ajustarem son els següents:

- ***C:*** paràmetre de penalització per cada error causat. Ha de ser estrictament positiu. Els valors seleccionats son els següents: 0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000.
- ***Kernel*:** tipus d'hiperplà usat per a la separació de les dades. Hi ha quatre tipus diferents: 'linear' (lineal), 'poly' (polinòmic), 'rbf' (de l'anglès *radial basis function*) i 'sigmoid'.
- ***Gamma*:** paràmetre per hiperplans no lineals. Quant més alt és el valor, més s'ajusta a les dades de *training.* El problema però, és que un valor molt alt pot causar *overfitting*. Hem escollit els valors següents: 0.0001, 0.001, 0.01, 0.1, 1, 'scale'.
- ***Class weight*:** pes específic per a cada classe. Per simplificar hem escollit únicament entre dos valors: 'balanced' o *None*. El mode balancejat ajusta automàticament els pesos de les classes de manera inversament proporcional a la freqüència de cada classe en el *dataset* d'entrada. El mode *None*, atorga a cada classe un pes de valor 1.
- ***Decision function shape*:** Estratègia usada a l'hora de classificar entre les diferents classes del model. Hi ha l'estratègia 'ovo' (de l'angles *one vs one*) o 'ovr' (de l'anglès *one vs rest*).

7 Resultats

Per a cadascun dels models desenvolupats, exposarem els resultats obtinguts després d'haver aplicat el *fine-tuning* per a l'obtenció del model òptim.

7.1 Mesures de rendiment

Per a la validació dels models de *machine learning*, utilitzarem les mesures *accuracy*, *f1-score*, *precision* i *recall*. Abans d'explicar cada una d'elles, aclarirem uns conceptes que ens serviran per explicar les mesures més endavant [88].

True → / *Predicted ↓*	Class 0	Class 1	...	Class N
Class 0	a	d	...	g
Class 1	b	e	...	h
...	...	...	...	...
Class N	c	f	...	i

Taula 7.1 Exemple matriu de confusió (Elaboració pròpia)

- **True Positives (TP):** valors de cada classe predits correctament, és a dir, que el valor predit coincideix amb el valor real de la classe.
- **False Positives (FP):** valors predits com a classe X quan corresponen a altres classes.
- **False Negatives (FN):** valors de la classe X predits com a altres classes.
- **Correct Predictions in Total (CPT):** valors totals predits correctament, és a dir, que el valor predit coincideix amb el valor real de la classe.

$$CPT = TP_{class\ 0} + \ldots + TP_{class\ N}$$

- **Incorrect Predictions in Total (IPT):** valors totals predits incorrectament, és a dir, que el valor predit no coincideix amb el valor real de la classe.

$$IPT = (FP_{class\ 0} + \ldots + FP_{class\ N}) + (FN_{class\ 0}) + \ldots + FN_{class\ N})$$

Aclarits els conceptes anterior, anem a definir les mesures que obtindrem al testejar els nostres models [89]:

- ***Accuracy*:** ràtio de totes les mostres predites correctament entre el nombre total de mostres.
  - *Fórmula:*

$$Accuracy = \frac{CPT}{CPT + IPT}$$

  - *Aclariments:*
    - Bona mesura per *datasets* balancejats
    - En *datasets* no balancejats, pot donar una alta *accuracy* tot i tenir un mal model o *dataset*.
- ***Precision*:** ràtio de totes les mostres predites correctament entre el nombre total de mostres predites com a aquella classe (correctament o incorrectament).

- *Fórmula:*

$$Precision_{class\ 0} = \frac{TP_{class\ 0}}{TP_{class\ 0} + FP_{class\ 0}}$$

$$Precision = AVG\ (precision_{class\ 0}, \dots, precision_{class\ 0})$$

$$= \frac{precision_{class\ 0} + \cdots + precision_{class\ N}}{N}$$

- *Aclariments:*
  - Respon a la pregunta (exemple): Quants *tweets* dels etiquetats com a 'happy' mostren realment el sentiment de felicitat?
  - Una *precision* alta significa una taxa baixa de falsos positius (FP)

- ***Recall*:** ràtio de totes les mostres predites correctament entre el nombre total de mostres que pertanyen a aquella classe.
  - *Fórmula:*

$$Recall_{class\ 0} = \frac{TP_{class\ 0}}{TP_{class\ 0} + FN_{class\ 0}}$$

$$Recall = AVG\ (recall_{class\ 0}, \dots, recall_{class\ 0}) = \frac{recall_{class\ 0} + \cdots + recall_{class\ N}}{N}$$

  - *Aclariments:*
    - Respon a la pregunta (exemple): Quants tweets que mostren felicitat han estat etiquetats com a 'happy'?

- **F1-Score:** mitjana ponderada del *precision* i el *recall*, tenint en compte tant els falsos positius com els falsos negatius.
  - *Fórmula:*

$$F1\ Score = \frac{2 \times (precision \times recall)}{precision + recall}$$

  - *Aclariments:*
    - Més útil que la *accuracy*, sobretot en *datasets* poc balancejats (la *accuracy* funciona millor si els falsos positius i els falsos negatius tenen un cost similar).
    - Si el cost dels falsos positius i els falsos negatius son molt diferents, és millor mirar la *precision* i el *recall.*

7.2 Resultats BERT i BETO

Per a validar si hem obtingut un bon model durant la fase d'entrenament i validació, extraiem dues mesures que son la *training loss* i *validation loss*. La pèrdua (en anglès *loss*) és un valor numèric que indica la penalització per una mala predicció. Si la predicció del model fos perfecte, aquest valor seria zero [90].

Tenint en compte el seu significat, la Taula 7.2 mostra resumidament com interpretar-ne els resultats durant l'entrenament del model. La interpretació per obtenir el model òptim correspondria a *"Good fitting"*.

Interpretation	Training loss	Validation loss
Underfitting	alt	alt
Overfitting	baix	alt
Good fitting	baix	lleugerament superior a *training loss*
Unknown fitting	alt	baix

Taula 7.2 Interpretació del training i validations loss (Elaboració pròpia)

Tal i com hem explicat anteriorment, els dos models han passat per una etapa de *fine-tuning* en la qual hem ajustat diferents paràmetres fins a obtenir el model òptim pel nostre *dataset*. Després de realitzar varies proves resumides en la **Error! Reference source not found.** amb diferents configuracions, hem obtingut el model definitiu.

- **Test 1:** *accuracy* prou bona, *good fitting*
- **Test 2:** *good fitting* però *accuracy* inferior al primer test
- **Test 3:** *learning rate* massa petita
- **Test 4:** *good fitting* però *accuracy* inferior al primer test (molt similar al primer test)
- **Test 5:** *overfitting* en la majoria de *epochs*
- **Test 6:** *epsilon* massa gran

A continuació, es mostra la configuració dels paràmetres del model òptim que correspon al primer test, a més del *training* i *validation loss* per a cada una de les *epochs* (veure Taula 7.3).

- **Learning rate:** 10^{-4}
- **Epsilon:** 10^{-4}
- **Epochs:** 10

Una vegada tenim la configuració adient, hem de seleccionar quin dels models obtinguts durant l'etapa de *training* és el més òptim.

BERT	BETO
Epoch 1 Training loss: 1.4190936471734727 Validation loss: 1.272063902446202 F1 Score (Weighted): 0.3772607914933663 Accuracy: 0.47575057736720555	Epoch 1 Training loss: 1.3292364265237535 Validation loss: 1.1599654299872262 F1 Score (Weighted): 0.43916980834321995 Accuracy: 0.535796766743649
Epoch 2 Training loss: 1.232545908008303 Validation loss: 1.0867045266287667 F1 Score (Weighted): 0.5492450678651604 Accuracy: 0.5981524249422633	Epoch 2 Training loss: 1.0962842575141363 Validation loss: 0.9970219646181379 F1 Score (Weighted): 0.6178727041248874 Accuracy: 0.625866050808314
Epoch 3 Training loss: 1.0318500058991569 Validation loss: 1.0470541545322962 F1 Score (Weighted): 0.5671333621966883 Accuracy: 0.5958429561200924	Epoch 3 Training loss: 0.8792334709848676 Validation loss: 0.9334965859140668 F1 Score (Weighted): 0.6327924587766909 Accuracy: 0.6443418013856813

Epoch 4 Training loss: 0.8837058948619025 Validation loss: 0.9722376550946917 F1 Score (Weighted): 0.6254013272587085 Accuracy: 0.6443418013856813	Epoch 4 Training loss: 0.7022646708147866 Validation loss: 0.9301177774156842 F1 Score (Weighted): 0.6664535400059269 Accuracy: 0.6766743648960739
Epoch 5 Training loss: 0.7786130506013121 Validation loss: 0.9478264961923871 F1 Score (Weighted): 0.6421177750435741 Accuracy: 0.6558891454965358	Epoch 5 Training loss: 0.5642643393948674 Validation loss: 0.9207701086997986 F1 Score (Weighted): 0.6651930744470659 Accuracy: 0.6720554272517321
Epoch 6 Training loss: 0.6871629740510669 Validation loss: 0.9425788010869708 F1 Score (Weighted): 0.6380173543691507 Accuracy: 0.6466512702078522	Epoch 6 Training loss: 0.4632206777376788 Validation loss: 1.0113985708781652 F1 Score (Weighted): 0.6574938807973506 Accuracy: 0.6605080831408776
Epoch 7 Training loss: 0.600846857896873 Validation loss: 0.9546443564551217 F1 Score (Weighted): 0.6460346915344719 Accuracy: 0.651270207852194	Epoch 7 Training loss: 0.3491776335452284 Validation loss: 1.0277880259922572 F1 Score (Weighted): 0.666161845315874 Accuracy: 0.6697459584295612
Epoch 8 Training loss: 0.5108701639941761 Validation loss: 0.9723458119801113 F1 Score (Weighted): 0.6352252216628854 Accuracy: 0.6420323325635104	Epoch 8 Training loss: 0.2850670662841627 Validation loss: 1.0928485138075692 F1 Score (Weighted): 0.6643512412885477 Accuracy: 0.6674364896073903
Epoch 9 Training loss: 0.45926921176058905 Validation loss: 0.9954503434044975 F1 Score (Weighted): 0.6396431657740442 Accuracy: 0.6443418013856813	Epoch 9 Training loss: 0.23027623551232473 Validation loss: 1.110574790409633 F1 Score (Weighted): 0.6503389326837806 Accuracy: 0.6535796766743649
Epoch 10 Training loss: 0.40277119034102987 Validation loss: 1.0022327048437936 F1 Score (Weighted): 0.6349724986219559 Accuracy: 0.6420323325635104	Epoch 10 Training loss: 0.20197792231504405 Validation loss: 1.103320164339883 F1 Score (Weighted): 0.6634145997686355 Accuracy: 0.6674364896073903

Taula 7.3 Resultats model òptim per a BERT i BETO (Elaboració pròpia)

Tal i com hem explicat prèviament, per a l'obtenció d'un model ben ajustat cal tenir una *training loss* baixa i una *validation loss* lleugerament superior. En aquest cas, pel model BERT Multlingual es produeix a la *Epoch 4* i pel BETO a la *Epoch 3*.

Per a veure en detall les mètriques de cadascun d'ells, la Taula 7.4 Resultats BERT i BETO (Elaboració pròpia) reflecteix els resultats obtinguts per a BERT Multilingual i BETO, i les seves corresponents matrius de confusió.

	Accuracy	***F1-score***	***Precision***	***Recall***
BERT Multilingual	0.64434	0.47414	0.72202	0.47022
BETO	0.67667	0.52039	0.74932	0.51007

Taula 7.4 Resultats BERT i BETO (Elaboració pròpia)

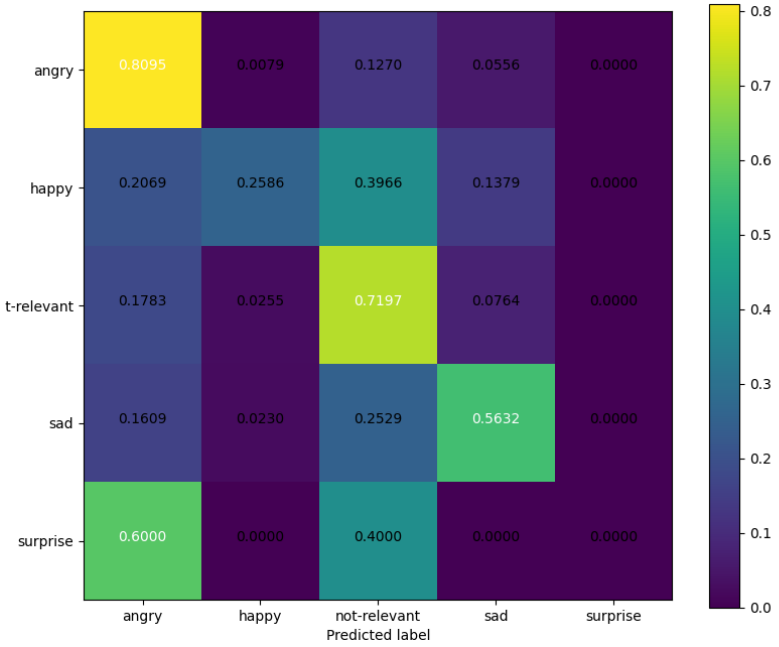

Figura 7.1 Matriu de confusió BERT (Elaboració pròpia)

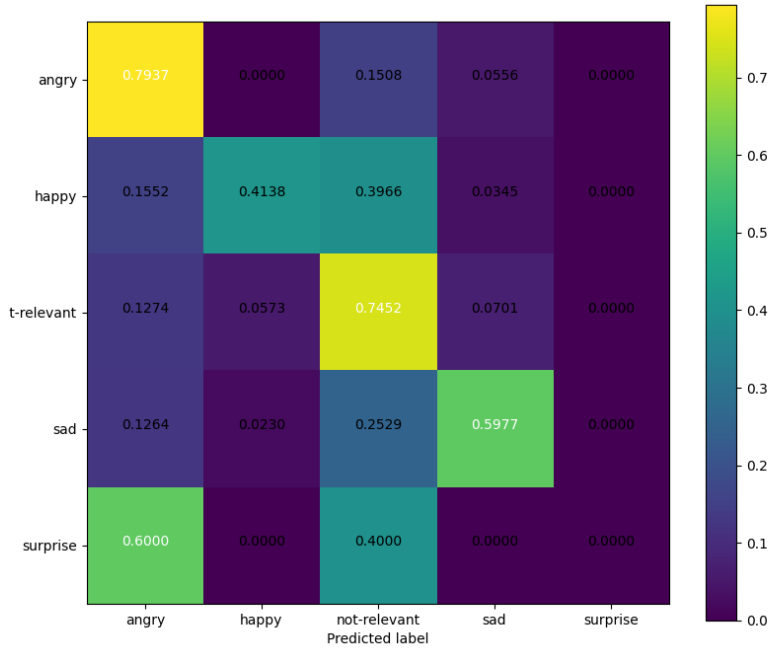

Figura 7.2 Matriu de confusió BETO (Elaboració pròpia)

Tal i com podem observar, ambdós models donen molt resultats similars. Tot i així, tenint en compte les mètriques de la **Error! Reference source not found.**, podem afirmar que el BETO és lleugerament millor ja que ha estat entrenat únicament amb text en espanyol.

Si observem les matrius de confusió (veure Figura 7.1 i Figura 7.2), podem veure que on el model té més error és en la detecció d'emocions *happy* i *surprise*. Això és degut a que hi ha molt poques mostres d'aquestes dues classes i el model no té prou dades per aprendre a classificar-les correctament. En el cas ideal, el model hauria d'estar balancejat en nombre de *tweets* per emoció. En el nostre cas no és una tasca fàcil, ja que no controlem com la gent reacciona a la Covid-19 a través de les xarxes socials.

7.3 Resultats SVC

Tal i com hem explicat a la secció anterior, el sistema creat per al model SVC comprova automàticament quin és el model que obté una millor precisió per als híper-paràmetres especificats a ajustar. Un cop executat el sistema de cerca del model, els paràmetres candidats han estat els següents:

```
SVC(C=1, degree=1, gamma=0.01, kernel='sigmoid', 'class_weight'=None,
'decision_function_shape'='ovr')

CalibratedClassifierCV(base_estimator=SVC(C=1, degree=1, gamma=0.01,
                                          kernel='sigmoid'), cv='prefit',
method='sigmoid')
```

Passada l'etapa d'entrenament del model i del calibrador del classificador, hem obtingut les mesures següents:

Accuracy	*F1-score*	*Precision*	*Recall*
0.28176	0.19878	0.19748	0.20294

Taula 7.5 Resultats SVC (Elaboració pròpia)

Com podem veure, les mètriques ofereixen un rendiment molt més baix que els models BERT i BETO. Això es pot deure a que aquests dos models han estat entrenats prèviament mentre que l'única informació que ha rebut el model SVC és el *dataset* que nosaltres li hem aportat. Aquest factor és veu molt més clarament a la matriu de confusió (veure Figura 7.3), on reflexa la carència d'equitat entre classes i qualitat de les dades. El model no detecta en cap cas l'emoció de sorpresa i la resta d'emocions les etiqueta, en la majoria de casos, com a 'angry' o 'not-relevant', ja que son les emocions que contenen un major nombre de mostres.

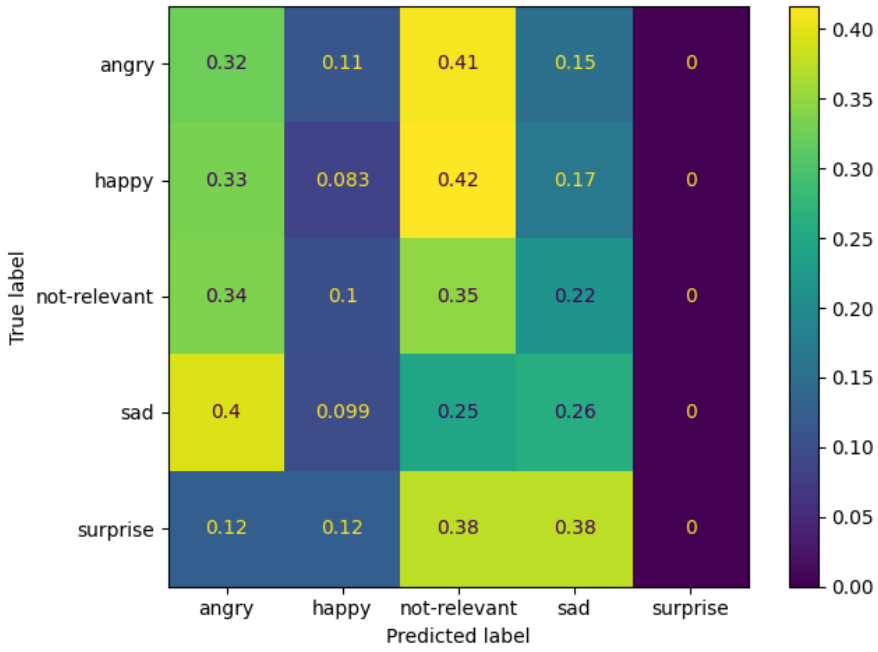

Figura 7.3 Matriu de confusió SVC (Elaboració pròpia)

8 Conclusions i treballs derivats

Un cop desenvolupat aquest projecte durant aquests mesos, puc afirmar que el camp de *machine learning* és quelcom ambigu i que requereix de molt temps i recursos. Entrant en el detall del sistema i els models escollits, mentre que el BETO i el BERT obtenen uns resultats prou acurats pel conjunt de dades obtingut, el model SVC ha resultat un fracàs. S'ha de tenir en compte, que tant el BETO com el BERT estan pre-entrenats amb milers de dades per aprendre del llenguatge natural. Així doncs, no és d'estranyar que quan l'entrenem per a classificar emocions obtingui millors resultats que el SVC, ja que "només" ha d'aprendre tasques de classificació, d'emocions en aquest cas. En canvi, el model SVC només rep les dades proporcionades pel nostre corpus. Degut a que el nombre de mostres no és molt elevat i està poc balancejat, els resultats obtingut no m'han sorprès. Era d'esperar que el model predigues erròniament en molts casos i confongués entre les diferents classes. El que també era predictible és la incapacitat de predir l'emoció de sorpresa. Aquest factor es produeix en qualssevol dels casos, ja que disposem un nombre molt petit de mostres (gairebé insignificant) per a aquesta emoció.

Si reprenem els objectius del projecte i de l'estudi de recerca, la nostra fita principal era veure com la població reacciona al coronavirus a través de xarxes socials mitjançant la obtenció de missatges de Twitter relacionats amb la Covid-19 i construir un sistema que ens permetés fer la classificació de manera automàtica. Si bé és cert que hem complert satisfactòriament tots les objectius, ja que em pogut observar que, en general, la població reacciona amb tristesa i enuig i hem pogut construir un bon sistema que compleix amb tots els requisits fixats; en el camp de *machine learning* sempre hi ha possibilitats de millora. Com hem comentat anteriorment, desenvolupar un sistema que sigui capaç de classificar en emocions un text és quelcom molt complex, ja que en moltes ocasions ni els humans som capaços de fer-ho.

Com a treball futur, es podrien aplicar algunes millores per obtenir un model més acurat i precís. Com hem pogut veure durant la realització del projecte, les dades obtingudes per a l'entrenament del model son molt importants. Un primer pas per a la millora del model podria ser l'obtenció de més dades i més balancejades entre classes. Per això però, segurament hauríem d'ampliar el context i no centrar-ho només en la Covid-19, ja que hem vist que predomina el sentiment d'enuig i tristesa, i aquest factor no el controlem nosaltres. Es contempla l'opció d'ampliar el context dels *tweets*, una tècnica utilitzada és el *transfer learning* (en parlarem a la següent secció) que bàsicament consisteix en la reutilització de dades per a l'entrenament del model. Un altre mètode o estudi a realitzar seria la combinació de varis corpus per a l'ampliació del volum de dades i l'equilibri en el nombre de mostres entre les diferents emocions. En ambdues opcions, tot i que la realització de l'estudi prendria el seu temps, si els resultats son satisfactoris ajudaria a reduir costos en temps i recursos. La tasca d'obtenció d'un *dataset* de qualitat i el seu etiquetatge és un procés molt lent i ambigu, ja que les emocions no son quelcom racional.

8.1 Competències tècniques

En aquest apartat es detallaran cadascuna de les competències tècniques especificades a l'inici del projecte i la manera en que s'ha assolit cada una d'elles:

- **CES1.1: Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics. [En profunditat]**
  S'ha desenvolupat un sistema que compta amb diversos subsistemes, cada un desenvolupats per separat, però tots units mitjançant APIs RESTful. A part de les pròpiament creades, s'ha fet ús de serveis externs com l'API de Microsoft per a la traducció del text o l'API de ParallelDots per aplicació de *sentiment analysis*. També hem fet ús del Apache Kafka per a la comunicació entre el monitor de Twitter i el Orchestrator.

- **CES1.2: Donar solució a problemes d'integració en funció de les estratègies, dels estàndards i de les tecnologies disponibles. [En profunditat]**
  Tal i com he comentat al punt anterior, tot i que cada subsistema (excepte l'Orchestrator que és el motor principal del sistema) es pot utilitzar de manera independent, hem creat APIs de cada component per a la comunicació entre ells. El monitor de Twitter està connectat amb el motor principal mitjançant el Kafka.

- **CES1.3: Identificar, avaluar i gestionar els riscos potencials associats a la construcció de software que es poguessin presentar. [Bastant]**
  Durant l'inici del projecte, es va realitzar un plantejament i una etapa de planificació del projecte en qüestió. A part de fixar els objectius i acotar l'abast, de realitzar la planificació temporal i estudiar la viabilitat econòmica, es va realitzar un anàlisi de possibles riscos i la solució que podríem aportar en cas de que sorgissin al llarg de l'estudi.

- **CES1.7: Controlar la qualitat i dissenyar proves en la producció de software. [Bastant]**
  S'han realitzat proves per comprovar cada component per separat i proves per a comprovar que la integració del sistema funciona correctament. Per a la realització de les proves s'ha utilitzat el software Postman. Per a la comprovació i millora dels models de *machine learning* s'ha realitzat etapes de validació per a comprovar la qualitat del model i del conjunt de dades obtingut.

- **CES1.8: Desenvolupar, mantenir i avaluar sistemes de control i de temps real. [En profunditat]**
  Disposem del monitor de Twitter que, connectat a l'API pròpia de Twitter, ens permet recollir els *tweets* en temps real amb els filtres establerts. Aquest *tweets* s'envien a Kafka que permet la rebuda i, al mateix temps, l'enviament dels missatges a l'Orchestrator de manera molt eficient.

- **CES1.9: Demostrar comprensió en la gestió i govern dels sistemes software. [Bastant]**
  Per a la realització i desenvolupament del sistema s'ha dissenyat i explicat detalladament cadascun dels components que el formen, incloent imatges que ajudin a la comprensió. També s'han descrit cadascuna de les funcionalitats principals acompanyant-les de diagrames de seqüència per facilitar l'enteniment del funcionament del sistema.

- **CES2.1: Definir i gestionar els requisits d'un sistema software. [Bastant]**
  A l'inici del projecte es van definir els objectius del sistema conjuntament amb els requisits funcionals que complementaven cada un d'ells. Addicionalment, es fan especificar els

requisits no funcionals del sistema que son crucials per al bon desenvolupament del software.

- **CES2.2: Dissenyar solucions apropiades en un o més dominis d'aplicació, utilitzant mètodes d'enginyeria del software que integrin aspectes ètics, socials, legals i econòmics. [Bastant]**
  En l'etapa de gestió del projecte, és va desenvolupar un pla de gestió econòmica i viabilitat del producte software. A més, és va redactar un informe de sostenibilitat per tenir en compte tots els aspectes a nivell econòmic, ambiental i social. Addicionalment, vam remarcar les lleis i regulacions que s'havien de valorar a l'hora de desenvolupar el sistema.

8.2 Estudis i treballs derivats

El sistema desenvolupat per aquest projecte de recerca té molt més potencial del abastat en aquest projecte. Inicialment, s'ha desenvolupat pels objectius fixats pel curs d'aquesta línia d'investigació, però s'ha utilitzat per a dues vessants més. Actualment, dins els grup de recerca del GESSI, aquest sistema s'ha aprofitat pel següent:

- **EmoEvalEs:** Participació en un seminari de *sentiment analysis* per a text en espanyol
- **Transfer learning:** Aplicació de *transfer learning* i realització d'un paper pel CrowdRE.

8.2.1 EmoEvalEs

El EmoEvalEs és un concurs organitzat per la SEPLN (Sociedad Española para el Procesamiento del Lenguaje Natural) i el IberLEF (Iberian Languages Evaluation Forum), basat en l'aplicació d'*emotion analysis* en text generat per usuaris en xarxes socials, Twitter en aquest cas. Des del 2012, la SEPLN ja organitzava aquest concurs per a *sentiment analysis* en espanyol, però no va ser fins el passat 2020 que van introduir la classificació del text en emocions.

El *dataset* aportat pels organitzadors consisteix en un total de 8223 *tweets* en espanyl relacionats amb diferents esdeveniments que van tenir lloc l'abril del 2019 relacionats amb diferents dominis: entreteniment, catàstrofes, política, commemoracions globals i vagues a nivell mundial. Els *tweets* estan distribuïts en tres subconjunts: *dev* (844 *tweets*), *train* (5723 *tweets*) i *test* (1656 *tweets*). Els *datasets* de *dev* i *train* estan etiquetats amb set emocions diferents (veure Taula 8.1) i s'utilitzen pel desenvolupament i entrenament del model. El *dataset* de *test* no està etiquetat ja que s'utilitzarà per avaluar cadascun dels models desenvolupats pels participants del concurs.

Emotion	**Nombre de mostres**		
	dev	*train*	*test*
anger (also includes annoyance and rage)	85	589	168
disgust (also includes disinterest, dislike, and loathing)	16	111	33
fear (also includes apprehension, anxiety, concern, and terror)	9	65	21
joy (also includes serenity and ecstasy)	181	1227	354

sadness (also includes pensiveness and grief)	104	693	199
surprise (also includes distraction and amazement)	35	238	67
others: the emotion expressed in a tweet as 'neutral or no emotion'	414	2800	814

Taula 8.1 Distribució d'emocions pel dataset EmoEvalEs (Elaboració pròpia)

Per a la participació al concurs, hem utilitzat algunes parts del sistema creat. El model escollit per aplicar la classificació ha estat el BETO, ja que és el que millor ha resultat en aquest projecte. El procediment seguit consta de tres etapes (veure Figura 8.1):

- **Pre-processament tweets:** Aplicació de tècniques de processament del llenguatge natural per "netejar" i normalitzar el text.
- **Entrenament del model:** Entrenar el model pel *dataset* del concurs amb el model BETO.
- ***Fine-tuning* model:** Aplicar l'ajustament dels híper-paràmetres (els mateixos que vam ajustar pel projecte) iterativament fins a obtenir bons resultats.

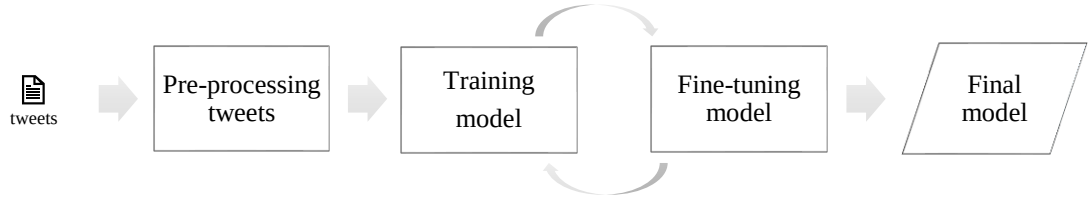

Figura 8.1 Procediment EmoEvalEs (Elaboració pròpia)

Un cop finalitzada l'etapa d'avaluació del model de cadascun dels participants, cada equip ha de redactar un *paper*. Actualment està en fase d'avaluació i pendent de ser publicat. Si tot segueix el curs previst, aquest estiu es faran públics tots els articles redactats pels participants i el realitzat per a l'organització. Al setembre es realitzarà un *workshop* que tancarà la competició.

8.2.2 *Transfer learning*

El *transfer learning* és una tècnica de processament del llenguatge natural utilitzada quan un model entrenat per a una tasca específica és reutilitzat per a la realització d'una tasca relacionada. Per aquest cas particular, reutilitzarem el model entrenat amb un *dataset* en concret per a classificar els missatges en emocions d'un altre *dataset* diferent.

Els *datasets* utilitzats son *datasets* públics extrets de diferents recursos i diferents contextos, i son els següents:

- ***Emotion Dataset for NLP*** [91]**:** Col·lecció de documents que conté 20000 missatges etiquetats amb una de les següents emocions: 'sad', 'angry', 'joy', 'surprise', 'fear', 'love'. Per adaptar-ho al *dataset* elaborat per nosaltres (el mateix que per aquest treball de fi de grau), hem canviat els noms d'algunes emocions ('joy' per 'happy') i eliminat les que no contemplàvem ('love' i 'fear').
- ***SMILE Twitter Emotion Dataset*** [92]**:** Conjunt de *tweets* relacionats amb el Brisith Museum creats amb el propòsit de classificar emocions expressades a Twitter en el context de l'art i les experiències culturals als museus. El corpus original conté 3085 entrades classificades en les següents emocions: 'anger', 'disgust', 'happiness', 'surprise', 'sadness', 'not-relevant'. Novament, pel nostre *target* hem modificat lleugerament alguns noms ('happiness' per 'happy' i 'sadness' per 'sad) i eliminat els *tweets* etiquetats com 'disgust' i 'not-relevant'.
- ***Twitter Reviews Dataset*** [93]**:** Corpus format per 10017 *tweets* originàriament, etiquetats en sis emocions diferents ('happy', 'sad', 'surprise', 'disgust', 'angry', 'fear'). En aquest cas, hem eliminat els *tweets* classificats com a 'disgust' i 'fear'.

El procediment seguit per a l'aplicació de *transfer leaerning* és molt similar al utilitzat per el concurs EmoEvalEs. L'única variant en aquest cas és que hem reutilitzat el *training* de cadascun dels models per testejar-los amb altres conjunts de dades. A diferència del EmoEvalEs, aquí hem provat tots els models desenvolupats amb tots els corpus, realitzant totes les combinacions possibles de cada *dataset* amb cada model.

Actualment, estem treballant en un paper per a participar en el CrowdRE '21, (Crowd-Based Requirements Engineering) un *workshop* basat en la enginyeria de requisits enfocat aquest any a la pandèmia de la Covid-19 i la reforçant la bretxa entre el CrowdRE i el desenvolupament.

8.3 Avaluació personal

Aquest projecte, i el meu pas pel grup de recerca GESSI en general, ha suposat un gran salt en l'àmbit acadèmic i professional. Fa poc més d'un any vaig entrar al grup sense tenir cap coneixement de tècniques de d'aprenentatge automàtic ni processament del llenguatge natural. Per a mi ha estat un gran repte la tasca d'autoaprenentatge per adquirir els coneixement necessaris de *machine learning* i desenvolupar aquest sistema des de zero.

Sé que em queda molt per aprendre en aquest sector i en molts d'altres, sé que em que em queden anys de formació, però també sé que aquest és només el principi. Durant aquest cinc anys en el Grau d'Enginyeria Informàtica he viscut moments molt intensos, molt durs i de molt esforç però, tal i com diuen, tot esforç té la seva recompensa. I la meva recompensa és tancar aquesta etapa amb satisfacció per haver realitzat aquest projecte del que n'estic tan orgullosa.

9 Glossari

- ***API Request:*** sol·licitud d'un recurs realitzat a una API.
- ***Benchmark***: prova per a mesurar el rendiment d'un sistema o d'un dels components que el formen.
- ***Emoji*****:** caràcter que es visualitza com a un icona gràfic, molt comú en xarxes socials per a complementar el text.
- ***Endpoint:*** url utilitzada per a l'obtenció d'un recurs d'una API.
- **Esclat de gradients:** problema ocorregut quan s'acumulen grans errors en el gradient provocant un gran nombre d'actualitzacions en els pesos de la xarxa neuronal del model durant l'etapa de *training*.
- ***Inter-rater agreement:*** grau d'acordança entre dues o més persones que estan avaluant unes dades concretes. En aquest context, correspon al grau d'acordança a l'hora d'etiquetar els *tweets* per classificar-los en emocions.
- ***IPython kernel*****:** *shell* interactiu per a *Jupiter notebooks*.
- ***Jupiter notebooks:*** entorn de treball interactiu que permet desenvolupar codi en Python integrant codi, text i gràfiques.
- ***Kafka topic:*** categoria utilitzada per emmagatzemar un conjunt de *records*. Similar a una carpeta en un sistema de fitxers, on els fitxers serien els *records.*
- **Menció (Twitter):** fet d'anomenar un usuari a Twitter. Va acompanyat d'un @ abans del nom d'usuari.
- **Open-source:** software on el codi font és pública amb una llicència que forma part del domini públic
- ***Record*****:** vector de bytes que pot emmagatzemar qualsevol objecte en qualsevol format. Usualment s'envia en format JSON o text pla.
- **SSL (Secure Socket Layer):** protocol utilitzat per a mantenir una connexió segura a través d'una xarxa (normalment Internet) i protegir la informació.
- ***Stopword:*** paraula que manca de significat com pronoms, articles, preposicions, etc.
- ***Token (authorization):*** codi xifrat per autoritzar una petició, en aquest cas, per a l'autorització de les peticions realitzades a l'API.
- ***Token (machine learning):*** instància d'una seqüència de caràcters que s'uneixen formant una unitat semàntica útil per a processar.
- ***Workshop:*** esdeveniment de formació i/o entreteniment on es reuneixen diferents participants i aprenen noves habilitats o adquireixen nous coneixements amb la finalitat de millorar en un camp específic del seu àmbit.

10 Referències

[1] «Software and Service Engineering Group». https://gessi.upc.edu/en (consulta març 01, 2021).

[2] «Everything There Is to Know about Sentiment Analysis», *MonkeyLearn*. https://monkeylearn.com/sentiment-analysis/ (consulta març 19, 2021).

[3] «Sentiment Analysis: Types, Tools, and Use Cases», *AltexSoft*. https://www.altexsoft.com/blog/business/sentiment-analysis-types-tools-and-use-cases/ (consulta març 22, 2021).

[4] «▷ Sentiment Analysis: ¿Qué es y para qué sirve?», *Comunycarse Network Consultants*, feb. 10, 2020. https://www.comunycarse.com/es/sentiment-analysis-que-es-y-para-que-sirve/ (consulta març 20, 2021).

[5] «What is Sentiment Analysis? Definition, Types, Algorithms». https://theappsolutions.com/blog/development/sentiment-analysis/ (consulta març 20, 2021).

[6] T. Dalgleish i M. Power, *Handbook of Cognition and Emotion*. John Wiley & Sons, 2000.

[7] «Paul Ekman», *Paul Ekman Group*. https://www.paulekman.com/ (consulta feb. 28, 2021).

[8] M. Mohri, A. Rostamizadeh, i A. Talwalkar, *Foundations of Machine Learning, second edition*. MIT Press, 2018.

[9] Y. Zhang, *New Advances in Machine Learning*. BoD – Books on Demand, 2010.

[10] «The Three Types of Machine Learning Algorithms», *Pioneer Labs | Technology Strategists & Delivery Experts*. https://pioneerlabs.io/insights/the-three-types-of-machine-learning-algorithms (consulta feb. 28, 2021).

[11] «Oracle Data Science», *Types of Machine Learning and Top 10 Algorithms Everyone Should Know*. https://blogs.oracle.com/datascience/types-of-machine-learning-and-top-10-algorithms-everyone-should-know-v2 (consulta feb. 27, 2021).

[12] por B. Jassova, «Chatbots y Procesamiento de Lenguaje Natural: La Guía Definitiva - Landbot.io», *https://landbot.io/es/*. https://landbot.io/es/blog/chatbots-procesamiento-lenguaje-natural/ (consulta març 02, 2021).

[13] «Natural Language Processing (NLP) Simplified : A Step-by-step Guide». https://datascience.foundation/sciencewhitepaper/natural-language-processing-nlp-simplified-a-step-by-step-guide (consulta feb. 28, 2021).

[14] «Volere Requirements Specification Template», *Volere Requirements*. https://www.volere.org/templates/volere-requirements-specification-template/ (consulta març 01, 2021).

[15] «Lynguo», *Instituto de Ingeniería del Conocimiento*. https://www.iic.uam.es/soluciones/entorno-digital/lynguo/ (consulta feb. 25, 2021).

[16] «ParallelDots | World Class NLP APIs for Text Analysis». https://www.paralleldots.com/text-analysis-apis (consulta feb. 25, 2021).

[17] «Emotion Analysis | ParallelDots AI APIs». https://www.paralleldots.com/emotion-analysis (consulta feb. 25, 2021).

[18] M. García-Vega *et al.*, «Overview of TASS 2020: Introducing Emotion Detection», set. 2020.

[19] K. Schwaber i J. Sutherland, «The Scrum Guide». nov. 2020. [En línia]. Disponible a: https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100

[20] «What is Scrum?», *Scrum.org*. https://www.scrum.org/resources/what-is-scrum (consulta feb. 27, 2021).

[21] Atlassian, «Jira | Issue & Project Tracking Software», *Atlassian*. https://www.atlassian.com/software/jira (consulta març 01, 2021).

[22] «GitHub», *GitHub*. https://github.com/ (consulta març 01, 2021).

[23] «Google Drive». https://www.google.com/intl/ca/drive/ (consulta març 01, 2021).

[24] «Google Meet». https://meet.google.com/ (consulta març 01, 2021).

[25] «Salarios | Indeed». https://es.indeed.com/career/salaries (consulta març 13, 2021).

[26] «Días laborables 2021 España». https://www.dias-laborables.es/dias_laborables_feriados_2021.htm (consulta març 11, 2021).

[27] «Precio kWh Gas Natural», *preciogas.com*. https://preciogas.com/faq/precio-kwh (consulta març 14, 2021).

[28] «Power Supply Calculator». https://www.newegg.com/tools/power-supply-calculator/ (consulta març 14, 2021).

[29] «Fibra Orange», *Orange*. https://www.orange.es/promociones-ofertas/fibra-internet (consulta març 14, 2021).

[30] «BOE.es - BOE-A-2018-16673 Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.» https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673 (consulta abr. 22, 2021).

[31] «BOE.es - BOE-A-1996-8930 Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia.» https://www.boe.es/buscar/act.php?id=BOE-A-1996-8930 (consulta abr. 22, 2021).

[32] «Apache License, Version 2.0». https://www.apache.org/licenses/LICENSE-2.0 (consulta juny 17, 2021).

[33] «Filtered stream | Twitter API». https://developer.twitter.com/en/docs/twitter-api/tweets/filtered-stream/introduction (consulta abr. 19, 2021).

[34] «FAQ | Covid-19 Stream». https://developer.twitter.com/en/docs/labs/covid19-stream/faq (consulta maig 05, 2021).

[35] «Part 1: Apache Kafka for beginners - What is Apache Kafka? - CloudKarafka, Apache Kafka Message streaming as a Service». https://www.cloudkarafka.com/blog/part1-kafka-for-beginners-what-is-apache-kafka.html (consulta maig 10, 2021).

[36] «Patrones creacionales». https://refactoring.guru/es/design-patterns/creational-patterns (consulta abr. 14, 2021).

[37] «Factory Design Pattern in Java», *JournalDev*, maig 22, 2013. https://www.journaldev.com/1392/factory-design-pattern-in-java (consulta abr. 14, 2021).

[38] «MVC Architecture in 5 minutes: a tutorial for beginners», *Educative: Interactive Courses for Software Developers*. https://www.educative.io/blog/mvc-tutorial (consulta abr. 19, 2021).

[39] N. Basta, «The Differences between Sigmoid and Softmax Activation Function», *Medium*, abr. 05, 2020. https://medium.com/arteos-ai/the-differences-between-sigmoid-and-softmax-activation-function-12adee8cf322 (consulta maig 18, 2021).

[40] P. Hagerty, «Entropic Ghosts», *Medium*, març 01, 2019. https://gab41.lab41.org/entropic-ghosts-35670292bc87 (consulta maig 19, 2021).

[41] A. Vaswani *et al.*, «Attention Is All You Need», *ArXiv170603762 Cs*, des. 2017, Consulta: feb. 12, 2021. [En línia]. Disponible a: http://arxiv.org/abs/1706.03762

[42] J. Devlin, M.-W. Chang, K. Lee, i K. Toutanova, «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding», *ArXiv181004805 Cs*, maig 2019, Consulta: feb. 12, 2021. [En línia]. Disponible a: http://arxiv.org/abs/1810.04805

[43] R. Horev, «BERT Explained: State of the art language model for NLP», *Medium*, nov. 17, 2018. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270 (consulta feb. 12, 2021).

[44] «google-research/bert», *GitHub*. https://github.com/google-research/bert (consulta maig 19, 2021).

[45] *dccuchile/beto*. DCC UChile, 2021. Consulta: maig 19, 2021. [En línia]. Disponible a: https://github.com/dccuchile/beto

[46] «SVC Documentation», feb. 19, 2021. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC (consulta feb. 19, 2021).

[47] C. Chih-Chung i L. Chih-Jen, «LIBSVM: A Library for Support Vector Machines», *2001*, p. 39, gen. 2021, [En línia]. Disponible a: https://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf

[48] «LIBSVM: A Library for Support Vector Machines», feb. 19, 2021. https://www.csie.ntu.edu.tw/~cjlin/libsvm/ (consulta feb. 19, 2021).

[49] «Información sobre Java 8». https://www.java.com/es/download/help/java8_es.html (consulta maig 27, 2021).

[50] «IntelliJ IDEA: el IDE de Java eficaz y ergonómico de JetBrains», *JetBrains*. https://www.jetbrains.com/es-es/idea/ (consulta maig 27, 2021).

[51] «Spring Boot». https://spring.io/projects/spring-boot#overview (consulta maig 27, 2021).

[52] «Maven – Introduction». https://maven.apache.org/what-is-maven.html (consulta maig 27, 2021).

[53] «javadoc». https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html (consulta maig 27, 2021).

[54] «Visual Studio Code - Code Editing. Redefined». https://code.visualstudio.com/ (consulta maig 27, 2021).

[55] «Welcome to Flask — Flask Documentation (2.0.x)». https://flask.palletsprojects.com/en/2.0.x/ (consulta maig 27, 2021).

[56] «Postman | The Collaboration Platform for API Development», *Postman*. https://www.postman.com/ (consulta maig 27, 2021).

[57] «API Documentation & Design Tools for Teams | Swagger». https://swagger.io/ (consulta maig 27, 2021).

[58] «Springfox Reference Documentation». http://springfox.github.io/springfox/docs/current/#introduction (consulta maig 29, 2021).

[59] «OkHttp». https://square.github.io/okhttp/ (consulta maig 29, 2021).

[60] «Google Code Archive - Long-term storage for Google Code Project Hosting.» https://code.google.com/archive/p/json-simple/ (consulta maig 29, 2021).

[61] «Jakarta Activation». https://eclipse-ee4j.github.io/jaf/ (consulta maig 29, 2021).

[62] «javax.net.ssl (Java Platform SE 8)». https://docs.oracle.com/javase/8/docs/api/javax/net/ssl/package-summary.html (consulta maig 29, 2021).

[63] «Kafka Java Client | Confluent Documentation». https://docs.confluent.io/clients-kafka-java/current/overview.html (consulta maig 29, 2021).

[64] «google/gson», *GitHub*. https://github.com/google/gson (consulta maig 29, 2021).

[65] «opencsv –». http://opencsv.sourceforge.net/ (consulta maig 29, 2021).

[66] «SLF4J». http://www.slf4j.org/ (consulta maig 29, 2021).

[67] «com.fasterxml.jackson.core (Jackson-core 2.8.0 API)». https://fasterxml.github.io/jackson-core/javadoc/2.8/com/fasterxml/jackson/core/package-summary.html (consulta maig 29, 2021).

[68] V. Durmont, *vdurmont/emoji-java*. 2021. Consulta: maig 29, 2021. [En línia]. Disponible a: https://github.com/vdurmont/emoji-java

[69] «Apache HttpComponents – HttpClient Overview». https://hc.apache.org/httpcomponents-client-5.1.x/ (consulta maig 29, 2021).

[70] A. Vicenzi, *emojis: Emojis for Python*. Consulta: maig 28, 2021. [En línia]. Disponible a: https://github.com/alexandrevicenzi/emojis

[71] *github/gemoji*. GitHub, 2021. Consulta: maig 28, 2021. [En línia]. Disponible a: https://github.com/github/gemoji

[72] «ipywidgets — Jupyter Widgets 8.0.0a4 documentation». https://ipywidgets.readthedocs.io/en/latest/ (consulta maig 28, 2021).

[73] «Matplotlib: Python plotting — Matplotlib 3.4.2 documentation». https://matplotlib.org/ (consulta maig 28, 2021).

[74] «Natural Language Toolkit — NLTK 3.6.2 documentation». https://www.nltk.org/ (consulta maig 28, 2021).

[75] «What is NumPy? — NumPy v1.20 Manual». https://numpy.org/doc/stable/user/whatisnumpy.html (consulta maig 28, 2021).

[76] «pandas documentation — pandas 1.2.4 documentation». https://pandas.pydata.org/docs/ (consulta maig 28, 2021).

[77] «PyEnchant — PyEnchant 3.2.0 documentation». http://pyenchant.github.io/pyenchant/ (consulta maig 28, 2021).

[78] «PyTorch documentation — PyTorch 1.8.1 documentation». https://pytorch.org/docs/stable/index.html (consulta maig 28, 2021).

[79] «Torch | Scientific computing for LuaJIT.» http://torch.ch/ (consulta maig 28, 2021).

[80] «scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation». https://scikit-learn.org/stable/ (consulta maig 28, 2021).

[81] «SciPy library — SciPy.org». https://www.scipy.org/scipylib/index.html (consulta maig 28, 2021).

[82] «seaborn: statistical data visualization — seaborn 0.11.1 documentation». https://seaborn.pydata.org/ (consulta maig 28, 2021).

[83] «Overview», *Stanza*. https://stanfordnlp.github.io/stanza/ (consulta maig 28, 2021).

[84] «Transformers». index.html (consulta maig 28, 2021).

[85] J. Brownlee, «How to Configure the Learning Rate When Training Deep Learning Neural Networks», *Machine Learning Mastery*, gen. 22, 2019. https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/ (consulta maig 31, 2021).

[86] «Setting the learning rate of your neural network.», *Jeremy Jordan*, març 02, 2018. https://www.jeremyjordan.me/nn-learning-rate/ (consulta maig 31, 2021).

[87] J. Brownlee, «A Gentle Introduction to Transfer Learning for Deep Learning», *Machine Learning Mastery*, des. 19, 2017. https://machinelearningmastery.com/transfer-learning-for-deep-learning/ (consulta abr. 16, 2021).

[88] B. Shmueli, «Multi-Class Metrics Made Simple, Part I: Precision and Recall», *Medium*, juny 04, 2020. https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bddc2 (consulta maig 30, 2021).

[89] «Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures», *Exsilio Blog*, set. 09, 2016. https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/ (consulta maig 30, 2021).

[90] «Descending into ML: Training and Loss | Machine Learning Crash Course», *Google Developers*. https://developers.google.com/machine-learning/crash-course/descending-into-ml/training-and-loss (consulta maig 30, 2021).

[91] «Emotions dataset for NLP», feb. 03, 2021. https://kaggle.com/praveengovi/emotions-dataset-for-nlp (consulta feb. 03, 2021).

[92] «SMILE Twitter Emotion Dataset». https://kaggle.com/ashkhagan/smile-twitter-emotion-dataset

[93] «Twitter Reviews for Emotion Analysis», feb. 10, 2021. https://kaggle.com/shainy/twitter-reviews-for-emotion-analysis (consulta feb. 10, 2021).

Annex

A Documentació APIs

A1 API Sentiment Analysis

`POST /api/emotion`

Parameters		
Name	**Description**	**Example**
Authorization *required string *(header)*	Authorization	ce3adcbe-1cdb-40f6-b107-8dd45ef246f4
Texto *required *(body)*	Text a analitzar	`{ "text": "Este año ha sido muy duro... A ver cuándo volvemos a la normalidad de una vez 😭" }`
tool *required string *(query)*	*Available values* : ParallelDots, BETO, BERT, SVC	BERT

`POST /api/translator`

Parameters		
Name	**Description**	**Example**
Authorization *required string *(header)*	Authorization	ce3adcbe-1cdb-40f6-b107-8dd45ef246f4
Texto *required *(body)*	Text a traduir	`{ "language": { "from": "en", "to": "es" }, "text": "Covid19 sucks but during this year I've lived moments I'll never forget, it's helped me a lot in my personal growth" }`
translator *required string *(query)*	*Available values* : Microsoft	Microsoft

A2 API Tweets Preprocessing

`POST /tweets-preprocessing/v1/preprocessing`

Parameters		
Name	**Description**	**Example**
Authorization *required string *(header)*	Authorization (x-api-key)	ce3adcbe-1cdb-40f6-b107-8dd45ef246f4

Texto *required
(body)
	Text a processar	{
		"text": " Este año ha sido muy duro... A ver cuándo volvemos a la normalidad de una vez 😭"
		}

A3 API BERT

`POST /bert/v1/emotion`

Parameters		
Name	**Description**	**Example**
Authorization *required string *(header)*	Authorization (x-api-key)	ce3adcbe-1cdb-40f6-b107-8dd45ef246f4

Texto *required
(body)
	Text a analitzar	{
		"text": " Este año ha sido muy duro... A ver cuándo volvemos a la normalidad de una vez 😭",
		"bert": true
		}

BERT *required Boolean *(body)*	Indica si el model a escollir correspon al BERT Multilingual (true) o al BETO (false)	

A4 API SVC

`POST /svc/v1/emotion`

Parameters		
Name	**Description**	**Example**
Authorization *required string *(header)*	Authorization (x-api-key)	ce3adcbe-1cdb-40f6-b107-8dd45ef246f4

Texto *required
(body)
	Text a analitzar	{
		"text": " Este año ha sido muy duro... A ver cuándo volvemos a la normalidad de una vez 😭"
		}

B Testing results

B1 BERT i BETO

Test 1

Configuració hyperparàmetres

- **Learning rate:** 10^{-4}
- **Epsilon:** 10^{-4}
- **Epochs:** 10

Resultats

BERT	BETO
Epoch 1 Training loss: 1.4190936471734727 Validation loss: 1.272063902446202 F1 Score (Weighted): 0.3772607914933663 Accuracy: 0.47575057736720555	Epoch 1 Training loss: 1.3592871129512787 Validation loss: 1.1873394761766707 F1 Score (Weighted): 0.4180746493027739 Accuracy: 0.5219399538106235
Epoch 2 Training loss: 1.232545908008303 Validation loss: 1.0867045266287667 F1 Score (Weighted): 0.5492450678651604 Accuracy: 0.5981524249422633	Epoch 2 Training loss: 1.117236077785492 Validation loss: 0.9930353931018284 F1 Score (Weighted): 0.6049974750725771 Accuracy: 0.628175519630485
Epoch 3 Training loss: 1.0318500058991569 Validation loss: 1.0470541545322962 F1 Score (Weighted): 0.5671333621966883 Accuracy: 0.5958429561200924	Epoch 3 Training loss: 0.8929755304540906 Validation loss: 0.9276351162365505 F1 Score (Weighted): 0.6484380265329022 Accuracy: 0.6605080831408776
Epoch 4 Training loss: 0.8837058948619025 Validation loss: 0.9722376550946917 F1 Score (Weighted): 0.6254013272587085 Accuracy: 0.6443418013856813	Epoch 4 Training loss: 0.7216845090900149 Validation loss: 0.9085568104471479 F1 Score (Weighted): 0.6874001213472563 Accuracy: 0.6951501154734411
Epoch 5 Training loss: 0.7786130506013121 Validation loss: 0.9478264961923871 F1 Score (Weighted): 0.6421177750435741 Accuracy: 0.6558891454965358	Epoch 5 Training loss: 0.5742231904129896 Validation loss: 0.8966374397277832 F1 Score (Weighted): 0.6806337739000112 Accuracy: 0.6859122401847575
Epoch 6 Training loss: 0.6871629740510669 Validation loss: 0.9425788010869708 F1 Score (Weighted): 0.6380173543691507 Accuracy: 0.6466512702078522	Epoch 6 Training loss: 0.4606810265353748 Validation loss: 0.9809404781886509 F1 Score (Weighted): 0.6611693450608653 Accuracy: 0.6651270207852193
Epoch 7 Training loss: 0.600846857896873 Validation loss: 0.9546443564551217 F1 Score (Weighted): 0.6460346915344719 Accuracy: 0.651270207852194	Epoch 7 Training loss: 0.35291927946465357 Validation loss: 0.9901713984353202 F1 Score (Weighted): 0.6715228285088506 Accuracy: 0.6766743648960739
Epoch 8 Training loss: 0.5108701639941761 Validation loss: 0.9723458119801113 F1 Score (Weighted): 0.6352252216628854 Accuracy: 0.6420323325635104	Epoch 8 Training loss: 0.29721553650285515 Validation loss: 1.0381126063210624 F1 Score (Weighted): 0.6645059916600142 Accuracy: 0.6697459584295612
Epoch 9 Training loss: 0.45926921176058905 Validation loss: 0.9954503434044975 F1 Score (Weighted): 0.6396431657740442 Accuracy: 0.6443418013856813	Epoch 9 Training loss: 0.2394789909677846 Validation loss: 1.0818981613431657 F1 Score (Weighted): 0.6518232032186704 Accuracy: 0.6558891454965358

Epoch 10 Training loss: 0.40277119034102987 Validation loss: 1.0022327048437936 F1 Score (Weighted): 0.6349724986219559 Accuracy: 0.6420323325635104	Epoch 10 Training loss: 0.2212754748761654 Validation loss: 1.056197430406298 F1 Score (Weighted): 0.6712647950305031 Accuracy: 0.6766743648960739

Taula 10.1 Resultats test 1 per a BERT i BETO (Elaboració pròpia)

Test 2

Configuració hyperparàmetres

- **Learning rate:** 10^{-4}
- **Epsilon:** 10^{-3}
- **Epochs:** 10

Resultats

BERT	BETO
Epoch 1 Training loss: 1.4008269309997559 Validation loss: 1.6952241488865443 F1 Score (Weighted): 0.27507710765584276 Accuracy: 0.37644341801385683	Epoch 1 Training loss: 1.3506198227405548 Validation loss: 1.1982551642826624 F1 Score (Weighted): 0.3945889449159417 Accuracy: 0.49653579676674364
Epoch 2 Training loss: 1.3722265788487025 Validation loss: 1.3659139020102364 F1 Score (Weighted): 0.340194990446315 Accuracy: 0.44110854503464203	Epoch 2 Training loss: 1.1444105867828642 Validation loss: 1.0688342792647225 F1 Score (Weighted): 0.5846603299645989 Accuracy: 0.6004618937644342
Epoch 3 Training loss: 1.215350559779576 Validation loss: 1.295008829661778 F1 Score (Weighted): 0.4017597736304433 Accuracy: 0.49191685912240185	Epoch 3 Training loss: 0.9368031940289906 Validation loss: 0.9598419666290283 F1 Score (Weighted): 0.6176869053997606 Accuracy: 0.6304849884526559
Epoch 4 Training loss: 1.1844096290213721 Validation loss: 1.2868468250547136 F1 Score (Weighted): 0.4163583236070757 Accuracy: 0.5219399538106235	Epoch 4 Training loss: 0.7504362000950745 Validation loss: 0.9382847973278591 F1 Score (Weighted): 0.6493984553280938 Accuracy: 0.6581986143187067
Epoch 5 Training loss: 1.0706241450139455 Validation loss: 1.2334997483662196 F1 Score (Weighted): 0.42337711567138653 Accuracy: 0.49653579676674364	Epoch 5 Training loss: 0.6113789062947035 Validation loss: 0.9723068475723267 F1 Score (Weighted): 0.64650985052587 Accuracy: 0.6558891454965358
Epoch 6 Training loss: 0.9812999495438167 Validation loss: 1.2429982423782349 F1 Score (Weighted): 0.4208430123203384 Accuracy: 0.5150115473441108	Epoch 6 Training loss: 0.5189246764140469 Validation loss: 0.9937594958714077 F1 Score (Weighted): 0.6338793077939017 Accuracy: 0.6374133949191686
Epoch 7 Training loss: 0.8942682679210391 Validation loss: 1.3634675741195679 F1 Score (Weighted): 0.506768324942068 Accuracy: 0.5057736720554272	Epoch 7 Training loss: 0.4082446140902383 Validation loss: 1.0292005283491952 F1 Score (Weighted): 0.6382183212798638 Accuracy: 0.6420323325635104
Epoch 8 Training loss: 0.7984382105725152 Validation loss: 1.3276012454714095 F1 Score (Weighted): 0.5546742680456282 Accuracy: 0.5542725173210161	Epoch 8 Training loss: 0.32798914917345556 Validation loss: 1.0560204897608076 F1 Score (Weighted): 0.642915195308584 Accuracy: 0.6466512702078522

Epoch 9 Training loss: 0.6605075065578733 Validation loss: 1.367737659386226 F1 Score (Weighted): 0.5736551563364691 Accuracy: 0.5727482678983834	Epoch 9 Training loss: 0.27094427549413275 Validation loss: 1.086911141872406 F1 Score (Weighted): 0.6234483356406456 Accuracy: 0.625866050808314
Epoch 10 Training loss: 0.5214616166693824 Validation loss: 1.4131737181118555 F1 Score (Weighted): 0.58080238546019 Accuracy: 0.5796766743648961	Epoch 10 Training loss: 0.23057868917073523 Validation loss: 1.0870214870997839 F1 Score (Weighted): 0.6352112426260528 Accuracy: 0.6397228637413395

Taula 10.2Resultats test 2 per a BERT i BETO (Elaboració pròpia)

Test 3

Configuració hyperparàmetres

- **Learning rate:** 10^{-4}
- **Epsilon:** 10^{-5}
- **Epochs:** 10

Resultats

BERT	BETO
Epoch 1 Training loss: 1.6081472337245941 Validation loss: 1.561235444886344 F1 Score (Weighted): 0.26809884789664074 Accuracy: 0.3371824480369515	Epoch 1 Training loss: 1.525798065321786 Validation loss: 1.4340324401855469 F1 Score (Weighted): 0.23679831565109627 Accuracy: 0.3464203233256351
Epoch 2 Training loss: 1.5361188650131226 Validation loss: 1.4829593215669905 F1 Score (Weighted): 0.19329744311609684 Accuracy: 0.3625866050808314	Epoch 2 Training loss: 1.4002525167805808 Validation loss: 1.341052566255842 F1 Score (Weighted): 0.33906238387981347 Accuracy: 0.43648960739030024
Epoch 3 Training loss: 1.4628789126873016 Validation loss: 1.425196579524449 F1 Score (Weighted): 0.2489084525835204 Accuracy: 0.3787528868360277	Epoch 3 Training loss: 1.319686872618539 Validation loss: 1.281186546598162 F1 Score (Weighted): 0.41171246253172195 Accuracy: 0.5196304849884527
Epoch 4 Training loss: 1.3995424338749476 Validation loss: 1.376261489731925 F1 Score (Weighted): 0.3815157422829059 Accuracy: 0.48498845265588914	Epoch 4 Training loss: 1.2489646673202515 Validation loss: 1.2310119015829903 F1 Score (Weighted): 0.41914610710589834 Accuracy: 0.5242494226327945
Epoch 5 Training loss: 1.3517859620707375 Validation loss: 1.3363205705370222 F1 Score (Weighted): 0.382918405830879 Accuracy: 0.48498845265588914	Epoch 5 Training loss: 1.1923323039497649 Validation loss: 1.1993252549852644 F1 Score (Weighted): 0.41423676835369977 Accuracy: 0.5127020785219399
Epoch 6 Training loss: 1.323236154658454 Validation loss: 1.2991755178996496 F1 Score (Weighted): 0.3845954778227095 Accuracy: 0.48729792147806006	Epoch 6 Training loss: 1.164062227521624 Validation loss: 1.1778877462659563 F1 Score (Weighted): 0.4272120161083944 Accuracy: 0.5173210161662818
Epoch 7 Training loss: 1.2906488180160522 Validation loss: 1.2726737260818481 F1 Score (Weighted): 0.38727932302745194	Epoch 7 Training loss: 1.125227187361036 Validation loss: 1.1625463451657976 F1 Score (Weighted): 0.4401740468616592

Accuracy: 0.4896073903002309	Accuracy: 0.5265588914549654
Epoch 8 Training loss: 1.2699986015047346 Validation loss: 1.255010826247079 F1 Score (Weighted): 0.3928678053872481 Accuracy: 0.49653579676674364	Epoch 8 Training loss: 1.1071733215025492 Validation loss: 1.1533348730632238 F1 Score (Weighted): 0.4381894900022052 Accuracy: 0.5242494226327945
Epoch 9 Training loss: 1.2575464291231973 Validation loss: 1.2456533738545008 F1 Score (Weighted): 0.3872937076626275 Accuracy: 0.4896073903002309	Epoch 9 Training loss: 1.1088071316480637 Validation loss: 1.1458356380462646 F1 Score (Weighted): 0.44632646750593236 Accuracy: 0.5242494226327945
Epoch 10 Training loss: 1.2571118559156145 Validation loss: 1.243036116872515 F1 Score (Weighted): 0.3908940807573858 Accuracy: 0.4942263279445728	Epoch 10 Training loss: 1.1023264889206206 Validation loss: 1.1428513186318534 F1 Score (Weighted): 0.4519021499422367 Accuracy: 0.5311778290993071

Taula 10.3 Resultats test 3 per a BERT i BETO (Elaboració pròpia)

Test 4

Configuració hyperparàmetres

- **Learning rate:** 10^{-4}
- **Epsilon:** 10^{-4}
- **Epochs:** 15

Resultats

BERT	BETO
Epoch 1 Training loss: 1.4255589842796326 Validation loss: 1.32196991784232 F1 Score (Weighted): 0.37150904845763305 Accuracy: 0.47113163972286376	Epoch 1 Training loss: 1.322514295578003 Validation loss: 1.1493377344948905 F1 Score (Weighted): 0.4275909686358822 Accuracy: 0.5334872979214781
Epoch 2 Training loss: 1.2912725593362535 Validation loss: 1.1408530984606062 F1 Score (Weighted): 0.5051522977701214 Accuracy: 0.5612009237875288	Epoch 2 Training loss: 1.0844129834856306 Validation loss: 0.9786755868366787 F1 Score (Weighted): 0.6258244543991728 Accuracy: 0.6327944572748267
Epoch 3 Training loss: 1.071731571640287 Validation loss: 1.0995866230555944 F1 Score (Weighted): 0.5324567544586697 Accuracy: 0.5750577367205543	Epoch 3 Training loss: 0.8640229595558984 Validation loss: 0.9685097677367074 F1 Score (Weighted): 0.605394452145077 Accuracy: 0.6235565819861432
Epoch 4 Training loss: 0.932821803859302 Validation loss: 1.020452082157135 F1 Score (Weighted): 0.5847817240489621 Accuracy: 0.6027713625866051	Epoch 4 Training loss: 0.6853134036064148 Validation loss: 0.9151780349867684 F1 Score (Weighted): 0.6658002496160578 Accuracy: 0.674364896073903
Epoch 5 Training loss: 0.8186768782990319 Validation loss: 0.9923136489731925 F1 Score (Weighted): 0.6201513378603175 Accuracy: 0.6420323325635104	Epoch 5 Training loss: 0.5389825085710201 Validation loss: 0.9496343731880188 F1 Score (Weighted): 0.6661737416827549 Accuracy: 0.674364896073903
Epoch 6	Epoch 6

Training loss: 0.7156286516359874	Training loss: 0.4191281870007515
Validation loss: 0.9969464029584613	Validation loss: 1.0787475279399328
F1 Score (Weighted): 0.622935152592794	F1 Score (Weighted): 0.6578261343618726
Accuracy: 0.628175519630485	Accuracy: 0.6628175519630485
Epoch 7	Epoch 7
Training loss: 0.6217729481203216	Training loss: 0.2919134610731687
Validation loss: 1.0247136354446411	Validation loss: 1.1177141496113367
F1 Score (Weighted): 0.5997022743686277	F1 Score (Weighted): 0.6489283417832038
Accuracy: 0.6073903002309469	Accuracy: 0.651270207852194
Epoch 8	Epoch 8
Training loss: 0.5415339491197041	Training loss: 0.21778008441573807
Validation loss: 1.0011879801750183	Validation loss: 1.1619591202054704
F1 Score (Weighted): 0.6311917100798798	F1 Score (Weighted): 0.6393191389568498
Accuracy: 0.6374133949191686	Accuracy: 0.6420323325635104
Epoch 9	Epoch 9
Training loss: 0.44926713726350237	Training loss: 0.1519012161131416
Validation loss: 1.1096808484622411	Validation loss: 1.2444519996643066
F1 Score (Weighted): 0.6087849771414655	F1 Score (Weighted): 0.6674670964008301
Accuracy: 0.6166281755196305	Accuracy: 0.6697459584295612
Epoch 10	Epoch 10
Training loss: 0.36304019098835333	Training loss: 0.1321425415309412
Validation loss: 1.0725390315055847	Validation loss: 1.2976978506360735
F1 Score (Weighted): 0.6276771076399902	F1 Score (Weighted): 0.6679505183856838
Accuracy: 0.6304849884526559	Accuracy: 0.674364896073903
Epoch 11	Epoch 11
Training loss: 0.29119216171758516	Training loss: 0.1114759272230523
Validation loss: 1.1098191567829676	Validation loss: 1.3902158055986678
F1 Score (Weighted): 0.6477688954143281	F1 Score (Weighted): 0.6271750829372115
Accuracy: 0.651270207852194	Accuracy: 0.6304849884526559
Epoch 12	Epoch 12
Training loss: 0.2517892228705542	Training loss: 0.10761001746037177
Validation loss: 1.1675380212920052	Validation loss: 1.3429910966328211
F1 Score (Weighted): 0.6305123826044897	F1 Score (Weighted): 0.6700274651665483
Accuracy: 0.6351039260969977	Accuracy: 0.6766743648960739
Epoch 13	Epoch 13
Training loss: 0.19082642692540372	Training loss: 0.06130941790927734
Validation loss: 1.2223365136555262	Validation loss: 1.3896550621305193
F1 Score (Weighted): 0.643548798219111	F1 Score (Weighted): 0.6643406010642203
Accuracy: 0.6466512702078522	Accuracy: 0.6697459584295612
Epoch 14	Epoch 14
Training loss: 0.17037610922540938	Training loss: 0.05846372155273067
Validation loss: 1.2497090782438005	Validation loss: 1.4450451476233346
F1 Score (Weighted): 0.6477955567170617	F1 Score (Weighted): 0.6538538114112961
Accuracy: 0.6466512702078522	Accuracy: 0.6581986143187067
Epoch 15	Epoch 15
Training loss: 0.14931849044348514	Training loss: 0.04839315145675625
Validation loss: 1.217825276511056	Validation loss: 1.4329426458903722
F1 Score (Weighted): 0.6431011698901808	F1 Score (Weighted): 0.664254721564541
Accuracy: 0.6443418013856813	Accuracy: 0.6697459584295612

Taula 10.4 Resultats test 4 per a BERT i BETO (Elaboració pròpia)

Test 5

Configuració hyperparàmetres

- **Learning rate:** 10^{-4}
- **Epsilon:** 10^{-5}
- **Epochs:** 10

Resultats

BERT	BETO
Epoch 1 Training loss: 1.3642922746283668 Validation loss: 1.3293842928750175 F1 Score (Weighted): 0.4491187498339402 Accuracy: 0.5011547344110855	Epoch 1 Training loss: 1.192524750317846 Validation loss: 0.9991317306246076 F1 Score (Weighted): 0.5777111574115114 Accuracy: 0.605080831408776
Epoch 2 Training loss: 1.2102563828229904 Validation loss: 1.0053770116397314 F1 Score (Weighted): 0.6019215418826213 Accuracy: 0.6166281755196305	Epoch 2 Training loss: 0.9279307148286274 Validation loss: 0.9254064049039569 F1 Score (Weighted): 0.6452004916748235 Accuracy: 0.648960739030023
Epoch 3 Training loss: 0.8950024396181107 Validation loss: 0.9786469680922372 F1 Score (Weighted): 0.6218495381467192 Accuracy: 0.6351039260969977	Epoch 3 Training loss: 0.6203018778136798 Validation loss: 1.1419265610831124 F1 Score (Weighted): 0.5888666748582554 Accuracy: 0.6073903002309469
Epoch 4 Training loss: 0.6641924870865685 Validation loss: 0.9832038964544024 F1 Score (Weighted): 0.6374526431363129 Accuracy: 0.6443418013856813	Epoch 4 Training loss: 0.3443211585815464 Validation loss: 1.100507880960192 F1 Score (Weighted): 0.6652300696210405 Accuracy: 0.674364896073903
Epoch 5 Training loss: 0.4547652703310762 Validation loss: 1.1252786091395788 F1 Score (Weighted): 0.6256216168416378 Accuracy: 0.6397228637413395	Epoch 5 Training loss: 0.21469321634088243 Validation loss: 1.3260575532913208 F1 Score (Weighted): 0.6452795303178072 Accuracy: 0.6581986143187067
Epoch 6 Training loss: 0.316621017243181 Validation loss: 1.1262099572590418 F1 Score (Weighted): 0.6531680446823146 Accuracy: 0.6605080831408776	Epoch 6 Training loss: 0.1553690658350076 Validation loss: 1.1863428950309753 F1 Score (Weighted): 0.6490283208942862 Accuracy: 0.6535796766743649
Epoch 7 Training loss: 0.18993309619171278 Validation loss: 1.2145474978855677 F1 Score (Weighted): 0.6495591278139485 Accuracy: 0.6535796766743649	Epoch 7 Training loss: 0.07545493684509504 Validation loss: 1.5159059933253698 F1 Score (Weighted): 0.658160718838519 Accuracy: 0.6605080831408776
Epoch 8 Training loss: 0.12986971264971153 Validation loss: 1.3685834748404366 F1 Score (Weighted): 0.6536812197151295 Accuracy: 0.6605080831408776	Epoch 8 Training loss: 0.059706759289838374 Validation loss: 1.5364989723478044 F1 Score (Weighted): 0.6596022545613538 Accuracy: 0.6651270207852193
Epoch 9 Training loss: 0.08801514648699335 Validation loss: 1.3465901953833443 F1 Score (Weighted): 0.6576436791054959 Accuracy: 0.6605080831408776	Epoch 9 Training loss: 0.0273971309652552 Validation loss: 1.6398147344589233 F1 Score (Weighted): 0.6641802726818427 Accuracy: 0.6651270207852193
Epoch 10 Training loss: 0.06231486278453043 Validation loss: 1.3516345535005843 F1 Score (Weighted): 0.6517579106887879 Accuracy: 0.6535796766743649	Epoch 10 Training loss: 0.017264565153579627 Validation loss: 1.612288304737636 F1 Score (Weighted): 0.6703009472612952 Accuracy: 0.6720554272517321

Taula 10.5 Resultats test 5 per a BERT i BETO (Elaboració pròpia)

Test 6

Configuració hyperparàmetres

- **Learning rate:** 10^{-4}
- **Epsilon:** 10^{-3}
- **Epochs:** 10

Resultats

BERT	BETO
Epoch 1 Training loss: 1.5130892097949982 Validation loss: 1.417715379170009 F1 Score (Weighted): 0.1929698203311543 Accuracy: 0.3625866050808314	Epoch 1 Training loss: 1.4546517857483454 Validation loss: 1.362950427191598 F1 Score (Weighted): 0.2902474647698499 Accuracy: 0.40877598152424943
Epoch 2 Training loss: 1.4236695340701513 Validation loss: 1.3708589587892805 F1 Score (Weighted): 0.1929698203311543 Accuracy: 0.3625866050808314	Epoch 2 Training loss: 1.3563616829259055 Validation loss: 1.2726406029292516 F1 Score (Weighted): 0.4006041284372602 Accuracy: 0.5011547344110855
Epoch 3 Training loss: 1.357023115668978 Validation loss: 1.3119322402136666 F1 Score (Weighted): 0.37776570101784157 Accuracy: 0.4780600461893764	Epoch 3 Training loss: 1.250683912209102 Validation loss: 1.1945089953286308 F1 Score (Weighted): 0.4309047710269314 Accuracy: 0.5265588914549654
Epoch 4 Training loss: 1.2639868876763753 Validation loss: 1.2684750046048845 F1 Score (Weighted): 0.39101744079699685 Accuracy: 0.4942263279445728	Epoch 4 Training loss: 1.1606410443782806 Validation loss: 1.135825821331569 F1 Score (Weighted): 0.48333357775103414 Accuracy: 0.5496535796766744
Epoch 5 Training loss: 1.216241574713162 Validation loss: 1.2469552414757865 F1 Score (Weighted): 0.39132699938215904 Accuracy: 0.4942263279445728	Epoch 5 Training loss: 1.081484671149935 Validation loss: 1.093581429549626 F1 Score (Weighted): 0.5110082214435662 Accuracy: 0.5635103926096998
Epoch 6 Training loss: 1.2011747317654746 Validation loss: 1.2283053738730294 F1 Score (Weighted): 0.4026086221998682 Accuracy: 0.5080831408775982	Epoch 6 Training loss: 1.0565650399242128 Validation loss: 1.0612123267991203 F1 Score (Weighted): 0.5501126135041929 Accuracy: 0.5958429561200924
Epoch 7 Training loss: 1.1697134524583817 Validation loss: 1.221071686063494 F1 Score (Weighted): 0.4077039662142467 Accuracy: 0.5150115473441108	Epoch 7 Training loss: 0.9889470487833023 Validation loss: 1.032767449106488 9 F1 Score (Weighted): 0.5634971704916544 Accuracy: 0.6004618937644342
Epoch 8 Training loss: 1.1515632569789886 Validation loss: 1.2130014555794852 F1 Score (Weighted): 0.40512348606298076 Accuracy: 0.5103926096997691	Epoch 8 Training loss: 0.9661029087645667 Validation loss: 1.0134769167218889 F1 Score (Weighted): 0.5956359017056247 Accuracy: 0.6235565819861432
Epoch 9 Training loss: 1.1479096412658691 Validation loss: 1.2107775381633215	Epoch 9 Training loss: 0.9568475719009127 Validation loss: 1.0040347916739327

F1 Score (Weighted): 0.4077792508661108 Accuracy: 0.5127020785219399	F1 Score (Weighted): 0.6038285910272173 Accuracy: 0.628175519630485
Epoch 10 Training loss: 1.1462856914315904 Validation loss: 1.2022040401186262 F1 Score (Weighted): 0.40883412199809427 Accuracy: 0.5150115473441108	Epoch 10 Training loss: 0.9483434770788465 Validation loss: 1.0001442006656103 F1 Score (Weighted): 0.6086462915726538 Accuracy: 0.6327944572748267

Taula 10.6 Resultats test 6 per a BERT i BETO (Elaboració pròpia)

C Repositoris GitHub

Twitter Monitor
https://github.com/twittersentimentanalysis/Twitter-Monitor

Orchestrator
https://github.com/twittersentimentanalysis/Orchestrator

API Sentiment Analysis
https://github.com/twittersentimentanalysis/API-Sentiment-Analysis

API Tweets Preprocessing
https://github.com/twittersentimentanalysis/TSA-Tweets-preprocessing

APIs Models Machine Learning
API BERT
https://github.com/twittersentimentanalysis/TSA-BERT
API SVC
https://github.com/twittersentimentanalysis/TSA-SVC