UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Sistemas Informáticos y Computación

Sistema de recuperación de información semántica multilingüe

Trabajo Fin de Máster

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

AUTOR/A: Casamayor Segarra, Andreu

Tutor/a: Hurtado Oliver, Lluis Felip

Cotutor/a: Ahuir Esteve, Vicent

CURSO ACADÉMICO: 2022/2023

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

valgrAI
Valencian Graduate School and Research Network of Artificial Intelligence

DSIC DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València

Sistema de recuperació d'informació semàntica multilingüe

TREBALL FI DE MÀSTER

Màster Universitari en Intel·ligència Artificial, Reconeiximent de Formes i Imatge Digital

Autor: Andreu Casamayor Segarra
Tutor: Lluís Felip Hurtado Oliver, Vicent Ahuir Esteve

Curs 2022-2023

Dedicatòria

A les persones més properes i que més m'han ajudat: a ma mare Encarna, a mon pare Juan Carlos, a la meua germana Paula i a la meua parella Anna.

Agraïments

Voldria agrair als meus tutors Lluís i Vicent per la seua inestimable ajuda, consell i guia, que m'han brindat al llarg del desenvolupament, i que sense ella haguera sigut impossible portar a terme aquest projecte.

Resum

En el món modern en el qual vivim, és habitual estar rodejat d'informació innecessària, ja siga per mitjà de les pantalles del mòbil o ordinador. A conseqüència, cada vegada és més complicat trobar la informació que busquem. Aquest problema es pot pal·liar per mitjà dels Sistemes de Recuperació d'Informació (SRI), sistemes que s'encarreguen de processar una consulta i retornar la informació més rellevant. Aquest procés de cerca cada vegada és més pesat i costos, donat que cada volta hi ha més informació, i per això, la necessitat d'utilitzar nous algoritmes, mètodes i models que tinguen la capacitat d'emmagatzemar i tractar amb totes les noves dades.

El projecte desenvolupat consta de dos sistemes: un analitzador de models i un SRI. L'objectius del primer sistema és analitzar i comparar els diferents models de Processament del Llenguatge Natural que trobem al mercat actual i utilitzen les tecnologies més avançades com: SBERT, XLNet, MPNet. Una vegada hem analitzat i comparat els models, podem fer la selecció del model que millor s'ajusta a les nostres dades, i fer ús d'ell. Aquest sistema ha sigut desenvolupat utilitzant ferramentes com *Beir* i utilitzant el corpus *mMarco* i la seua traducció al català. En segon lloc, un sistema SRI: que utilitza el model prèviament seleccionat i entrenat. Tots aquest models es basen en representacions vectorials denses contextuals, que són capaços d'entendre els context que rodeja a la frase i captar el significat semàntic. El desenvolupament del SRI ha sigut possible gràcies a ferramentes com *Haystack* i *Hugging Face*.

Els resultats obtinguts han sigut molt satisfactoris i han demostrat que el model seleccionat és capaç d'agafar el significat contextual i semàntic de les frases i transformar-ho en representacions vectorials. En conclusió, hem creat un SRI que utilitza representacions vectorials denses amb les últimes tecnologies i que té un rendiment alt.

Paraules clau: Processament del Llenguatge Natural, Sistemes de recuperació d'informació, representacions vectorials denses, SBERT, XLNet, MPNet, analitzador de models

Resumen

En el mundo moderno en el cual vivimos, es habitual estar rodeado de información innecesaria, ya sea por medio de las pantallas del móvil u ordenador. En consecuencia, cada vez es más complicado encontrar la información que buscamos. Este problema se puede paliar por medio de los Sistemas de Recuperación de Información (SRI), sistemas que se encargan de procesar una consulta y devolver la información más relevante. Este proceso de búsqueda se está haciendo más lento y costoso, dado que cada vez hay más información que procesar, y por eso, la necesidad de utilizar nuevos algoritmos, métodos, modelos que tengan la capacidad de almacenar y tratar las enormes cantidades de datos.

El proyecto desarrollado consta de dos sistemas: un analizador de modelos y un SRI. El objetivo del primer sistema es analizar y comparar los diferentes modelos de Procesamiento del Lenguaje Natural que encontramos al mercado actual y utilizan las tecnologías más avanzadas como: SBERT, XLNet, MPNet. Una vez hemos analizado y comparado los modelos, podemos hacer la selección del modelo que mejor se ajusta a nuestros datos, y hacer uso de él. Este sistema ha sido desarrollado utilizando herramientas como *Beir* y utilizando el corpus *mMarco* y su traducción al catalán. En segundo lugar, un sistema SRI: que utiliza el modelo previamente seleccionado y entrenado. Todos estos modelos se basan en representaciones vectoriales densas contextuales, que son capaces de entender el contexto que rodea a la frase y captar el significado semántico. El desarrollo del SRI ha sido posible gracias a herramientas como *Haystack* i *Hugging Face*.

Los resultados obtenidos han sido muy satisfactorios i han demostrado que el modelo seleccionado es capaz de coger el significado contextual y semántico de las frases y transformarlo en representaciones vectoriales. En conclusión, hemos creado un SRI que utiliza representaciones vectoriales densas con las últimas tecnologías y que tiene un rendimiento alto.

Palabras clave: Procesamiento del Lenguaje Natural, Sistemas de recuperación de información, representaciones vectoriales densas, SBERT, XLNet, MPNet, analizador de modelos

Abstract

In the modern world in which we live, it is common to be surrounded by unnecessary information, either through mobile or computer screens. As a result, it is increasingly difficult to find the information we are looking for. This problem can be alleviated by means of Information Retrieval Systems (IRS), systems that process a query and return the most relevant information. This search process is becoming slower and more expensive, since there is more and more information to process, and therefore, the need to use new algorithms, methods, models that have the ability to store and treat the huge amounts of data.

The developed project consists of two systems: a model analyzer and an IRS. The objective of the first system is to analyze and compare the different Natural Language Processing models that we find in the current market and use the most advanced technologies such as: SBERT, XLNet, MPNet. Once we have analyzed and compared the models, we can make the selection of the model that best fits our data, and make use of it. This system has been developed using tools such as *Beir* and using the corpus of *mMarco* and its translation into Catalan. Secondly, an IRS system: which uses the previously selected and trained model. All these models are based on contextual dense vector representations, which are able to understand the context surrounding the sentence and capture the semantic meaning. The development of the SRI has been possible thanks to tools such as *Haystack* and *Hugging Face*.

The results obtained have been very satisfactory and have shown that the selected model is able to take the contextual and semantic meaning of the sentences and transform it into vector representations. In conclusion, we have created an IRS that uses dense vector representations with the latest technologies and has a high performance.

Key words: Natural Language Processing, Information retrieval systems, dense vector representations, SBERT, XLNet, MPNet, model analyzer

Índex

Índex de figures

Índex de taules

CAPÍTOL 1
Introducció

Hui en dia, vivim en un món on la càrrega d'informació és cada vegada major, a totes hores estem rebent informació per mitjà de pantalles, com la del mòbil o l'ordinador. Aquesta és la raó per la qual necessitem els Sistemes de Recuperació d'Informació (SRI), per poder filtrar tota la informació innecessària que hi ha a internet i cercar aquella que ens cal. Google és l'exemple més clar del que és un SRI per a la web, donat que per mitjà del seu cercador trobem aquella informació que es desitja d'una forma ràpida i precisa. No obstant això, hi ha d'altres exemples com Bing, Yahoo o DuckDuckGo.

Als últims anys estem vivint el boom de les xarxes neuronals, les quals han afectat el nostre dia a dia significativament, ajudant-nos a ser més eficients o millorar la productivitat en la nostra feina. Aquestes les podem trobar en tota mena de llocs, com per exemple als nostres ordinadors o en internet, també en diferents àmbits de la informàtica, com automatització de certes tasques, ajuda a la presa de decisions, o inclús per millorar o agilitzar el desenvolupament de *software*. Un dels àmbits més importants d'ús de les xarxes neuronals és el del Processament del Llenguatge Natural (PLN) on els models basats en xarxes han tingut un paper fonamental per augmentar significativament el rendiment de les solucions proposades per a certes tasques d'aquest àmbit.

Malgrat aquesta millora, trobem que en el mercat actual moltes de les ferramentes creades o les xarxes desenvolupades estan fetes per a idiomes majoritaris deixant a banda altres més minoritaris, com per exemple el català. Per això, aquest projecte té l'objectiu final d'aportar una nova ferramenta, un SRI en català utilitzant les eines més modernes i desenvolupant xarxes que puguen competir en el mercat actual.

1.1 Motivació

Les causes principals de l'elecció d'aquest TFM (recuperació d'informació basada en vectors densos) són les següents:

- Primer. Per la continuïtat amb el meu TFG, aquest projecte és una millora respecte al meu treball presentat amb anterioritat. El meu TFG consistia en la construcció de 3 SRI: Booleà, Word2Vec, STSB, per comparar les diferències existents entre els models. Donat que, aquest any he sigut estudiant del Màster d'Intel·ligència Artificial, Reconeixement de Formes i Imatge Digital, he pogut aprofundir en els sabers de l'àmbit del PLN i les xarxes neuronals. No sols això sinó que amb les noves experiències i coneixements adquirits sóc capaç de fer autocrítica del meu projecte i veure millores a afegir. Creant en mi una necessitat de millorar el meu projecte i fer-lo competitiu al mercat actual.

A més de crear una necessitat de millora en el meu projecte anterior, també han creat un necessitat d'aprenentatge de les noves tecnologies i avanços que han aparegut en aquest darrer any en els SRI. Convertint aquest projecte en un lloc on puc desenvolupar les meues capacitats, habilitats i coneixements.

- Segon. Sóc beneficiari d'una beca de col·laboració en el Departament de Sistemes Informàtics i Computació (DSIC) de la Universitat Politècnica de València (UPV), més concretament en l'equip de recerca **ELiRF**. L'equip se centra a desenvolupar projectes dins de l'àmbit del PLN. La meua feina dins de l'equip és el meu TFM, desenvolupar un SRI capaç de tractar amb el seu propi corpus de documents (DACSA), que explicarem més detalladament en capítols posteriors. També sóc beneficiari d'una beca de **valgrAI**, *Valencian Graduate School and Research Network of Artificial Intelligence*, una fundació sense ànim de lucre present a la Generalitat Valenciana i les 5 grans universitats públiques de la Comunitat Valenciana. Aquesta fundació s'encarrega de coordinar la formació i investigació sobre l'àmbit de la IA.

- Tercer. L'equip de recerca ELiRF forma part del projecte **AMIC-PoC: DESARROLLO DE UN PROTOTIPO PRECOMPETITIVO PARA EL ANALISIS AFECTIVO DE INFORMACION MULTIMEDIA**. El projecte comprén diverses universitats espanyoles com: la Universitat Politècnica de València, la Universidad de Zaragoza, la Universidad Politécnica de Madrid i la Universidad del País Vasco. El projecte consisteix en el desenvolupament d'un sistema d'anàlisi i catalogació de contingut multimèdia, l'objectiu del sistema és la recuperació de fragment multimèdia, sobretot de xarxes socials, que contesta a una consulta realitzada per l'usuari. El treball de l'equip ELiRF és el de crear el motor de recuperació dels fragments, per tant, d'aquest TFM s'espera extraure el millor model possible per a la feina de la recuperació. Suposant un repte investigador i personal per a la meua persona.

- Per últim. Fent una mirada exhaustiva del mercat actual, podem trobar la falta d'eines desenvolupades per al català. És a dir, els recursos disponibles per desenvolupar projectes en català són escassos i no es poden comparar amb els d'altres llengües majoritàries, ni en prestacions ni en nombre d'ells. En ser una persona catalanoparlant vull proporcionar el meu gra d'arena per donar suport a aquesta comunitat d'informàtics que utilitzem el català.

En conclusió, aquest projecte ha suposat un repte personal el qual ha ajudat a poder satisfer les meues ganes d'aprenentatge, desenvolupar les meues habilitats, i contribuir amb una nova ferramenta a l'àmbit del PLN en català.

1.2 Objectius

El principal objectiu d'aquest TFM és el següent:

Implementació d'un recuperador d'informació basat en representacions vectorials denses

Per poder assolir aquest objectiu final l'hem divit en subobjectius:

- Estudi dels sistemes de recuperació d'informació.

- Estudi del mercat actual, les noves xarxes i les ferramentes més potents.

- Estudi d'un sistema de valoració objectiva depenent de diferents mètriques.

- Creació i entrenament d'un model dens, com el *Sentence to BERT*, amb un gran nombre de dades.
- Valoració i comparació del nostre model amb altres models preentrenats.
- Implementació d'un recuperador d'informació semàntic, utilitzant el model amb les millors prestacions.
- Avaluació del SRI creat.

1.3 Impacte Esperat

Aquest projecte suposarà una nova ferramenta d'ús lliure dins de l'àmbit del PLN en català. No sols que utilitza tecnologies de l'estat de l'art, sinó que serà lliure el model, la xarxa neuronal, perquè qualsevol persona puga utilitzar-la per als seus projectes personals.

1.4 Estructura de la memòria

L'estructura de la memòria està configurada pels següents capítols:

1. **Introducció**: Preàmbul inicial on s'exposarà el problema i el projecte.
2. **Estat de l'art**: Es tractarà de resumir i explicar tota la teoria que envolta al projecte i també un estudi de les últimes tecnologies relacionades.
3. **Anàlisi del problema**: Explicarem el problema a resoldre, quines possibles solucions hem trobat i les especificacions dels projecte.
4. **Disseny de la solució**: Detallarem les parts que componen la solució i com estan estructurades.
5. **Desenvolupament de la solució proposada**: Explicarem com ha sigut el procés d'implementació amb les seues decisions i problemes.
6. **Experimentació**: Detallarem l'experimentació realitzada en aquest projecte.
7. **Resultats**: Avaluació dels resultats del sistema final.
8. **Conclusions**: Farem un breu resum per destacat tot el que em pogut asolir en aquest projecte.
9. **Treballs futurs**: Exposarem possibles funcionalitats a afegir.

CAPÍTOL 2
Estat de l'art

Aquest capítol presenta una explicació detallada de tota la teoria que fonamenta aquest projecte i una revisió de les últimes tecnologies que trobem.

2.1 Definició dels Sistemes de Recuperació d'Informació

En aquest apartat definirem que és un SRI, i per poder fer-ho primer hem de preguntarnos que significa el terme 'recuperar informació'.

2.1.1. Definició de recuperació d'informació

Resulta curiós que el terme "recuperació d'informació" genere tanta controvèrsia a l'hora d'establir una definició dins de l'àmbit de les ciències de la informació. *Rijsbergen* va definir el problema de la forma següent, "es tracta d'un terme que sol ser definit en un sentit molt ample" [10]. Aquesta controvèrsia a l'hora de definir aquest terme, ha generat que la comunitat científica estiga dividida en diferents grups de definicions.

En el primer grup d'autors la tecnologia informàtica ha tingut un paper fonamental a l'hora de formular-les, donat que consideren ambdues definicions (recuperació d'informació i recuperació de dades) sinònims. El principal exemple que trobem és al Glossari de l'Associació de Bibliotecaris Americans on defineix *'Information retrieval'* com a primera acceptació recuperació d'informació i com a segona recuperació de dades. No obstant això, hem de tindre en compte que la recuperació d'informació es pot fer sense recórrer a la tecnologia.

Un segon grup estableixen diferències entre les dues termes. Un dels grans exponents d'aquest grup és Blair, on en el seu llibre *'Language and representation in information retrieval'* exposa les següents diferències [5]:

1. En la recuperació de dades la pregunta és molt clara, està altament formalitzada, i, per tant, la resposta són les dades desitjades. En contraposició, en la recuperació d'informació la pregunta és prou ambigua i costa de traduir a un llenguatge formal, i, per tant, la resposta serà una sèrie de documents que, amb un grau d'incertesa, respondran a la pregunta.

2. Segons la relació anterior, requisits als sistema i satisfacció de l'usuari, podem concloure que la recuperació de dades és determinista, no hi ha incertesa, i la recuperació d'informació és probabilística, per la incertesa present.

3. Finalment, trobem una gran diferència a l'hora de determinar l'èxit en una cerca, en la recuperació de dades el criteri és l'exactitud, i en la recuperació d'informació és la satisfacció final de l'usuari amb la resposta.

Partint de les diferències anteriors, Rijsbergen planteja la següent taula [10]:

	Recuperació de dades	**Recuperació d'informació**
Encert	Exacte	Parcial, el millor
Model	Determinista	Probabilístic
Llenguatge de consulta	Fortament estructurada	Estructurat o natural
Especificació de la consulta	Precisa	Imprecisa
Error en la resposta	Sensible	Insensible

Taula 2.1: Recuperació de dades vs recuperació d'informació

En la Taula 2.1 podem comprovar a simple vista la definició de les diferència que plantejava Blair. On per la part de la recuperació de dades trobem que tots els seus processos tant de recuperació com de valoració del resultat són exactes, és a dir, no té cabuda la incertesa. Com a conseqüència, tenim una recuperació on tot ha de ser exacte, cada pregunta té la seua resposta exacta. Al contrari, en la recuperació d'informació depén de la probabilitat, és a dir, són recuperacions amb incertesa on el resultat s'avalua de forma subjectiva seguint la satisfacció de l'usuari.

Per acabar en aquest grup, Baeza-Yates planteja les següents definicions [11]:

- **Dades i Text**: "Les dades es poden estructurar en taules, arbres, etc. per recuperar exactament el que un desitja. Per contra, el text no posseeix una estructura clara i resulta difícil crear-la"
- **Recuperació d'informació**: "Donada una necessitat d'informació (consulta + perfil d'usuari + ...) i un conjunt de documents, s'han d'ordenar els documents de més a menys rellevants, i presentar un subconjunt d'eixos documents, format pels K primers documents més rellevants".

Donades aquestes dues definicions l'autor planteja una solució, aquesta està dividida en dues parts:

1. Tria del millor model que donada una entrada calcule la rellevància de tots els documents.
2. Disseny d'algorisme i estructura de dades per implementar eficientment el model.

Per últim, l'autor recalca que el problema de caracteritzar les necessitats de l'usuari: quina informació necessita depenent del seu context i la consulta, no és un problema senzill de resoldre.

El tercer grup d'autors pren com a base la definició efectuada per Salton, "la recuperació d'informació té a veure amb la representació, emmagatzematge, organització i accés als ítems d'informació" [1]. Aquests autors recalquen el fet que no és el seu treball diferenciar els dos termes anteriors o que ja estan prou diferenciats. Els majors representants són Feather i Storges que defineixen la recuperació d'informació com "el conjunt d'activitats necessàries per a fer disponible la informació a una comunitat d'usuaris" [6].

El quart i últim grup, es caracteritza per evadir definir el terme recuperació d'informació. En aquest grup tenim com a exponent Chowdhury, el qual sols indica que "el terme

recuperació d'informació va ser encunyat en 1952 i va anar guanyant popularitat en la comunitat científica de 1961 d'ara en avant, després, mostrant els propòsits, funcions i components dels SRI" [7].

2.1.2. Definició d'un SRI

En aquest projecte utilitzarem les definicions efectuades pel segon grup d'autors. Més concretament utilitzarem la definició efectuada per Salton: "qualsevol SRI pot ser descrit com un conjunt d'ítems d'informació (DOCS), un conjunt de peticions (REQS) i algun mecanisme (SIMILAR) que determine quin ítem satisfan les necessitats d'informació expressades per l'usuari en la petició" [1].

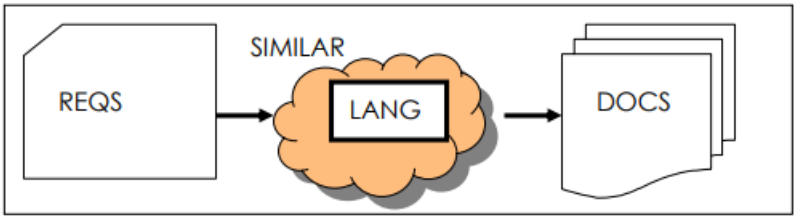


Figura 2.1: Esquema d'un SRI

La Figura 2.1 il·lustra la definició anterior, amb la particularitat d'afegir un LANG: sistema d'indexació o classificació que converteix els documents al format desitjat.

Una vegada definit que és un SRI, passarem a explicar les funcions principals:

1. Identificar les fonts d'informació rellevants a les àrees d'interés dels usuaris.
2. Analitzar els continguts dels documents i representar aquest contingut d'una forma adequada per a la comparació.
3. Analitzar la petició de l'usuari, i transformar-la d'una forma compatible amb la representació del documents.
4. Realitzar la comparativa entre la petició i els documents.
5. Retornar aquells documents més rellevants i ajustar el sistema amb la retroalimentació de l'usuari.

2.2 Història dels Sistemes de recuperació d'informació

Per posar en context aquest projecte, anem a fer un repàs a l'evolució del SRI al llarg del temps.Baez-Yates sintetitza aquesta evolució dividint-la en les següents 4 parts [11]:

1. **Inicis**: Un dels casos més antic que podem trobar són el mètodes de recuperació en les col·leccions de papirs en l'antic Egipte. Tanmateix, el cas més característic d'un mètode de recuperació són les taules de continguts dels llibres, que han anat evolucionant fins a convertir-se en l'índex que hui en dia tots coneixem i constitueixen els nuclis dels SRI.
2. **Recuperació d'informació en biblioteques**: Les biblioteques varen ser els primers organismes a utilitzar els SRI. Primerament desenvolupats per elles mateixes, sent

sistemes molt rústics i poc eficients. Posteriorment amb l'aparició d'un mercat informàtic, on empreses i institucions desenvolupen els seus propis sistemes, més eficients i potents.

3. **La World Wide Web**: L'evolució natural dels SRI els ha conduït a la web, on l'aplicació pràctica és enorme i on el nombre d'usuaris ha crescut exponencialment. En les últimes dècades hem sigut partícips de la consolidació de la web, produida per l'abaratiment de les tecnologies i el seu gran desenvolupament. A més, la web produeix un enorme nombre de documents a aquests sistemes, gràcies al fet que no hi ha cap mena de restriccions a l'hora de publicar un document.

4. **Present/Futur**: Podem afirmar que aquesta evolució no ha arribat a la seua fi, sinó que és un començament. Donat el gran volum de documents actuals, els sistemes de recuperació són cada vegada més necessaris. És a dir, els SRI s'adapten al medi, cada vegada són més eficients, extensos i ràpids. És un món en constant transformació. Com diu *Wang* "sistemes d'informació estan destinats a integrar-se plenament amb altres sistemes convencionals, arribant a ser més estesos i de major influència tant en negocis com en la vida familiar" [2]. Hui en dia, la gran millora que ha experimentat els SRI ha sigut causada per les denominades xarxes neuronals. Aquestes han possibilitat que els SRI guanyen en qualitat i rapidesa, fent possible el seu ús en la nostra vida quotidiana.

2.3 Sistemes de recuperació d'informació

En aquest apartat explicarem en més profunditat els processos i la funcionalitat que fan els SRI. Primer de tot estudiarem com es transforma el text per poder utilitzar-ho, posteriorment classificarem els SRI en diferents tipus i, finalment, estudiarem les possibles mètriques d'avaluació dels SRI.

2.3.1. Preprocessament de Documents

El preprocessament de documents s'utilitza per seleccionar aquells termes més determinants, que millor representen el document, i així obtindre posteriorment una millor discriminació entre documents. El preprocessament es realitza tant sobre les possibles consultes de l'usuari, com per als documents que componen el corpus. En finalitzar, podrem veure la consulta com una expressió la qual avaluarà quins documents del corpus la compleixen i quins no, o com document de referència amb el qual calcularem la distància semàntica per a cada document al corpus.

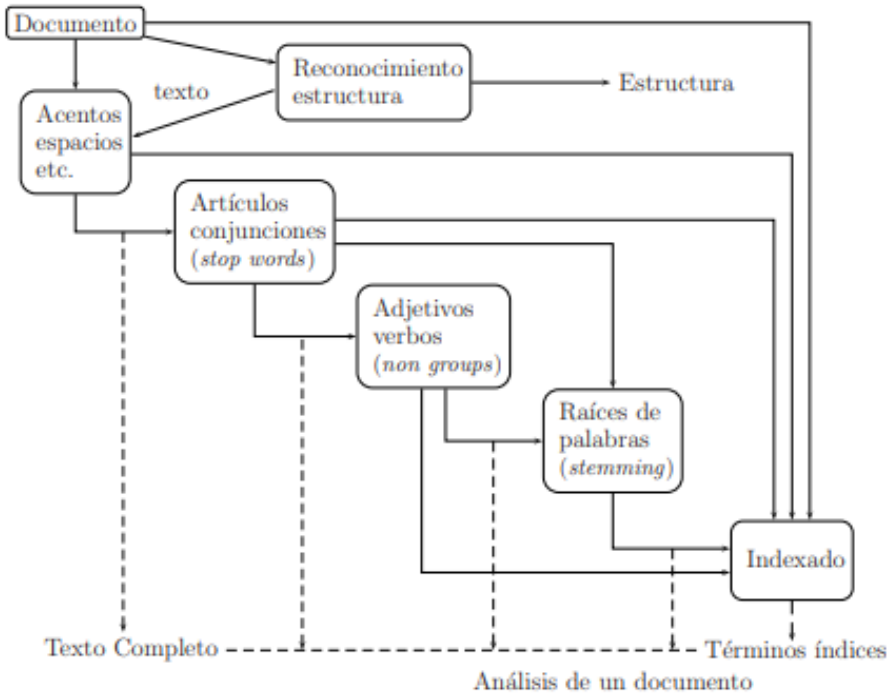


Figura 2.2: Preprocessament de documents. Font: Luis Gabriel Jaimes, Fernando Vega Riveros, Modelos clásicos de recuperación de la información, Revista Integración, Escuela de Matemáticas,Universidad Industrial de Santander, Vol. 23, No. 1, 2005, pág. 17–26

Com podem veure a la Figura 2.2, el procés es compon de les següents fases:

1. **Anàlisi lèxica del text**: L'objectiu d'aquesta fase és determinar quin tipus de tractament realitzarem sobre els números, guions, accents, majúscules i minúscules, etc.

2. **Eliminació de paraules buides**: Per norma general, no totes les paraules que ens trobem en una frase o document tenen el mateix valor discriminatiu; per exemple, paraules que actuen com a nom tenen un valor més gran. Per consegüent, aquesta fase es tracta de reduir el nombre de termes amb un valor discriminatiu xicotet o nul; així aconseguim, també, delimitar el nombre de termes que s'indexen.

3. **Aplicació de lematització**: Consisteix en l'eliminació de variacions morfo-sintàctiques i obtenció de termes lematitzats. És a dir, estem interessats en l'arrel dels verbs, el lexema, i eliminem sufixes i prefixes.

4. **Selecció de termes**: Última fase, que té com a objectiu seleccionar aquells termes que formaran part de l'índex, o la frase, depenent de la unitat base triada. En aquest cas, voldràs triar aquells termes amb major rellevància, és a dir, que millor representen la informació del text.

En resum, el preprocessament consisteix a analitzar i triar les paraules que millor representen el text, per a utilitzar-les en l'índex. Aquest procés explicat és opcional, els nous sistemes permeten no haver de fer un preprocessament del text per utilitzar-lo a posteriori.

2.3.2. Classificació dels SRI

A l'hora de dissenyar un SRI, s'ha de tindre en compte el model a utilitzar. Román Villena, defineix aquest models seguien les següents normes [12]:

- Mètode utilitzat per obtenir les representacions de la consulta i els documents.
- L'estratègia seguida per conéixer la rellevància del documents respecte a una consulta realitzada
- Estratègia d'ordenació per importància dels documents d'eixida.

Seguint aquesta definició, molts autors van proposar diferents classificacions per al models. Una de les més completes la formula Dominich, el qual proposa dividir el models en cinc grans grups [3]:

Models	**Descripció**
Models clàssics	Consisteixen en els models més comuns: Booleà, Espai Vectorial i Probabilístic
Models alternatius	Basats en la lògica difusa
Models lògics	Aquest models tracten a la recuperació d'informació com un procés inferencial. En conseqüència, estan bastats en la lògica formal.
Models basats en la interactivitat	Utilitzen la retroalimentació de l'usuari en la rellevància de documents, i són capaços d'expandir el grossor de la busqueda.
Models basats en Intel·ligència artificial	Estan basat majoritàriament en xarxes neuronals, bases de coneixements, algoritmes genètics i processament del llenguatge natural.

Taula 2.2: Classificació dels Models de RI

Un altre autor que formula una classificació és Baeza-Yates, el qual pren com a base de la divisió la tasca inicial que realitza un usuari en el sistema:

1. **Recuperació**: Ús d'una equació de cerca inserida en un formulari per a la recuperació d'informació
2. **Navegació**: Classificació per hipertext [8], molt utilitzada en la web, inicialitzada per la cerca de documents utilitzant les referències.

Partint d'aquesta primera classificació inicial, l'autor remarca l'existència de dos grans grups de recuperació:

- **Clàssics**: El grup del clàssics està compost pels models booleans, vectorials i probabilístic, posteriorment, l'autor, exposa paradigmes complementaris per cada model (algebraics, teoria de conjunts i probabilístics)
- **Estructurats**: El models que componen el grup d'estructurats, són aquells que usen llistes de termes sense superposició i nodes pròxims.

Una vegada vist els models de recuperació, l'autor, exposa que els models de navegació es poden dividir en tres grans tipus:

- **Estructura Plana**: Característica principal és que tracten cada document aïllat del context. Una lectura sense context.

- **Estructura Guiada**: Utilitza una estructura, un directori, composta per una jerarquia de classes i subclasses, per organitzar el documents, facilitant la cerca als usuaris.

- **Hipertext**: L'ús de nodes i enllaços per facilitar l'obtenció d'informació als usuaris.

A continuació trobem una taula resum de Baeza-Yates [11], la visió lògica dels documents:

Tipus	**Termes Índex**	**Text Complet**	**Text complet i Estructurat**
Recuperació	Clássics Conjunts teórics Algebraics Probabilístics	Clàssics Conjunts teòrics Algebraics Probabilístics	Estructurats
Navegació	Estructura Plana	Estructura Plana Hipertext	Estructura Guiada Hipertext

Taula 2.3: Classificació models segons Baeza-Yates

2.3.3. Conceptes previs dels Models

En aquesta secció farem un repàs als conceptes necessaris per a entendre el Sistemes de Recuperació d'Informació.

2.3.3.1. Distàncies

En aquest subapartat anem a fer un repàs teòric de les diferents distàncies que es poden utilitzar per al càlcul de la similitud entre vectors. Cal mencionar, que alguns models representen els documents en forma de vectors, i aquestes distàncies s'utilitzen per comparar-los.

Producte Escalar

El producte escalar és una mesura de similitud entre dos vectors que es calcula de la següent forma:

$$v \cdot u = v_1u_1 + v_2u_2 + ... + v_nu_n \tag{2.1}$$

On v = $v_1, v_2, ..., v_n$ i u = $u_1, u_2, ..., u_n$ són dos vectors amb dimensions iguals.

El producte escalar relaciona els angles dels dos vectors. Quan el resultat és alt assenyala que els vectors tenen components similars en la mateixa direcció, és a dir, són vectors similars. Al contrari, quan el valor està proper a 0 significa que són diferents o ortogonals, el que indica una baixa similitud.

És important tindre en compte que el producte escalar és sensible a les magnituds dels vectors, provocant que la captura de similituds entre patrons i estructures complexes siga més difícil.

Similitud Cosinus

La similitud cosinus és una de les mesures més utilitzades per a comparar vectors en el nostre àmbit de PLN. La mesura indica la similitud entre dos vectors per mitjà de la relació de l'angle que formen. Si tenim dos vectors u i v, la fórmula per calcular l'angle és la següent:

$$\cos\theta = \frac{u \cdot v}{|u||v|} \tag{2.2}$$

On $u \cdot v$ representa el producte escalar entre el vector i $|u|, |v|$ les normes dels vectors, respectivament.

El resultat estarà compres entre [-1,1], on els valors pròxims a 1 indiquen una similitud major, i valors pròxims a -1 al contrari. Donat que els valors pròxims a 1, l'angle és menor i tenen una direcció similar, i per als valors propers a -1 tenen direccions oposades.

Aquesta mesures és molt útil en el nostre camp, ja que no es veu afectada per les magnituds dels vectors, sinó que sols afecta l'angle i la direcció entre ells. Sent útil per a comparar vectors que representen característiques o propietats d'alguna cosa.

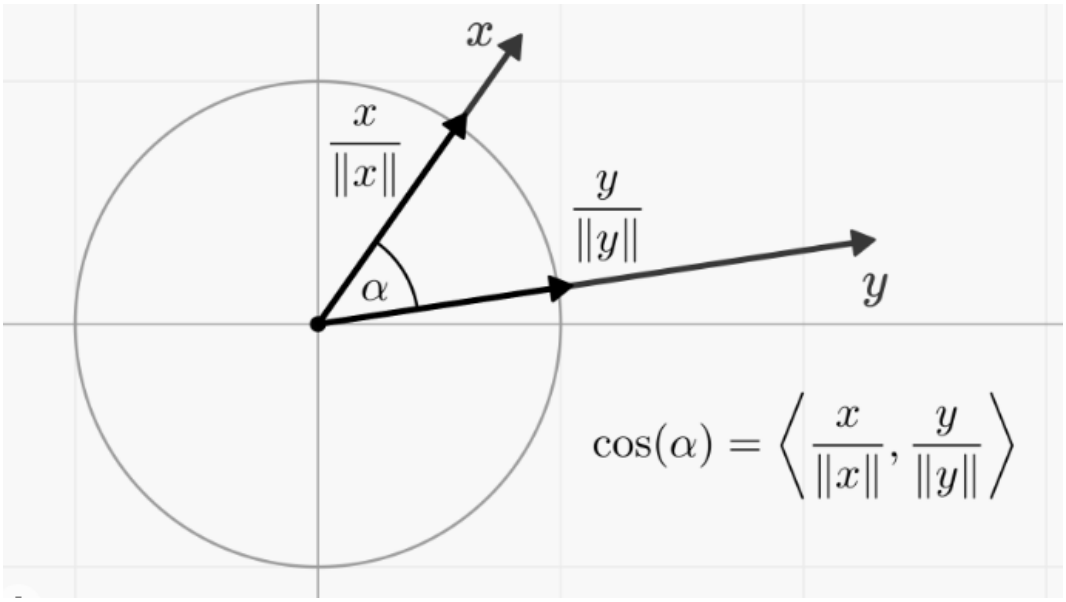


Figura 2.3: Similitud cosinus

Distància euclidiana

La distància euclidiana, com el seu propi nom indica, és una mesura de distància i no de similitud. No obstant això, si agafem la inversa de la distància la podem convertir en una mesura de similitud.

La distància euclidiana és la distància entre dos punts en un espai euclidià de múltiples dimensions. Per poder calcular la distància entre dos vectors, u i v, s'utilitza la següent fórmula:

$$distancia\_euclidiana(u,v) = \sqrt{(v_1-u_1)^2 + (v_2-u_2)^2 + ... + (v_n-u_n)^2} \tag{2.3}$$

Per poder transformar-la a una mesura de similitud hem de fer ús de la inversa, és a dir:

$$similitud\_euclidiana(u,v) = 1/(1 + distancia\_euclidiana(u,v)) \tag{2.4}$$

Aquesta transformació assigna un valor més alt a aquells vectors amb major similitud, és a dir, on la distància euclidiana és menor entre ells, i valors baixos als vectors que estan llunys entre ells.

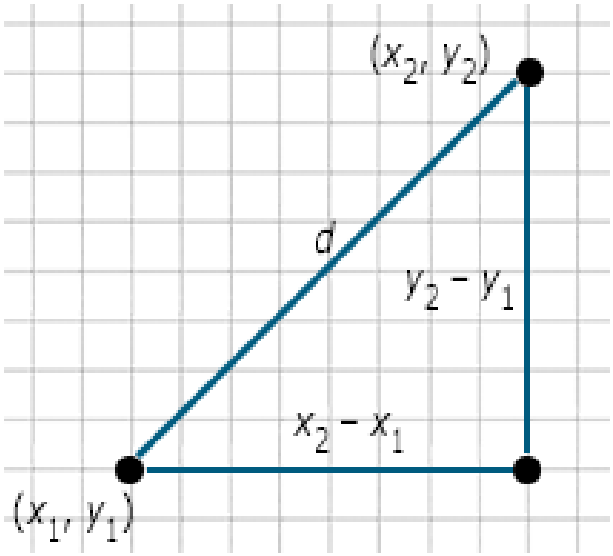


Figura 2.4: Exemple de la distància euclidiana

2.3.4. Models SRI Clàssics

En aquest subapartat anem a fer un repàs dels diferents models clàssics. Es tracten ja de models en desús en alguns àmbits per les seues baixes prestacions en comparació als models de xarxes neuronals, però sempre és bo entendre un poc des d'on partim per poder explicar el camí recorregut.

2.3.4.1. Model Booleà

El primer model clàssic, és un model basat en l'Àlgebra de Bool i la teoria de conjunt, on per cada consulta es retorna una seria de documents que continguen els termes cercats.

En aquest model, l'estructura utilitzada és una matriu de grandària MxN, on:

- M: és el nombre de documents en el corpus.
- N: És el nombre de termes en el vocabulari del corpus.

Cal remarcar, que s'ha fet un preprocessament, explicat en la Secció 2.3.1, del corpus per extraure de cada documents les paraules més rellevants, i aquestes seran les que confeccionen el Vocabulari.

Per cada document s'extraurà una filera d'1s i 0s amb talla N (mida del vocabulari). Cada posició de la llista indica si el terme i-esim del vocabulari apareix (1) o no (0) al document. Donat que es fa per cada document del corpus amb talla M, obtindrem la matriu MxN que representarà el corpus respecte al vocabulari elegit. La Taula 2.4 mostra un exemple d'una matriu per al model booleà on hi ha 5 documents i 5 termes:

	T1	T2	T3	T4	T5
D1	0	1	0	0	1
D2	0	0	1	1	0
D3	1	1	0	0	1
D4	1	0	1	0	0
D5	1	1	1	0	0

Taula 2.4: Exemple d'estructura per al model booleà

En la Taula 2.4 podem observar com per a cada document D tenim un vector de 0 i 1 depenent si apareix el terme T o no.

A l'hora de realitzar una consulta, els document són vists com un conjunt de termes i la consulta com una expressió booleana (cotxe *AND* carretera *AND* accident), obtenint per cada document un parell (document, consulta). A cada parell, li assignarem un valor de similitud. Aquest valor serà binari, 0 o 1, depenent si el document compleix l'expressió booleana o no. Aquesta expressió booleana està constituïda per les regles bàsiques de la lògica de conjunts, com poden ser la Intersecció, la Unió i la negació.

Avantatges:

- Implementació senzilla.
- Rapidesa en la cerca d'informació.

Desavantatges:

- No existeix cap discriminació entre documents de major o menor rellevància.
- Cap comptabilitat d'aparició dels termes, dona igual si un terme apareix una o mil voltes.
- No hi ha *matching* parcials dels termes.

En conclusió, aquest model clàssic sols té en compte si existeix el terme en el document o no, no és possible crear un ranking de rellevància entre el documents. Per tant, per a aquest projecte el descartarem.

2.3.4.2. Model de l'Espai Vectorial (MEV)

L'objectiu d'aquest model és dotar als documents d'un grau de pertinença a una consulta realitzada. Aquest model va ser creat per G. Salton, C.S. Yang i A. Wong en 1975 [54], i era el model més utilitzat per a la tasca de RI fins a l'aparició de les xarxes neuronals.

Utilitzant el preprocès explicat amb anterioritat a la Secció 2.3.1. El MEV selecciona les paraules més representatives de cada document. Per a cada document selecciona aquells termes no buits que aporten un significat substancial al document.

El model està constituït pels següents elements:

- D = $d_1, d_2, d_3, ..., d_N$, el conjunt de documents
- T = $t_1, t_2, t_3, ..., t_K$, el conjunt de termes que constitueixen el vocabulari del documents.
- Podem representar cada document, que són files de la matriu DxT, de la forma següent:

$$d_i = w_{i,1}, w_{i,2}, w_{i,3}, ..., w_{i,K}$$

On $w_{i,k}$ representa el pes del terme k en el document i. Per tant, cada document pot ser vist com un punt en l'espai vectorial amb dimension igual a la quantitat de tinga el vocabulari d'indexació.

- Les preguntes poden ser vistes igual que els document, P = $w.p1, w.p2, w.p3, ..., w.pK$, on k és el nombre de termes distints en la col·lecció.

Per poder captar la informació que aporta cada terme al document, és necessari elegir un mètode de pesat adequat, el més utilitzat el dia de hui és el **TF-IDF**.

TF-IDF

Mètode utilitzat per calcular els pesos dels termes dins del Model de l'Espai Vectorial (MEV), on l'objectiu és crear una matriu que relacione cada document amb els termes del vocabulari. Aquest mètode està compost pels següents elements:

- N = nombre de termes diferents en el vocabulari.
- D = nombre de documents que composen la col·lecció
- Matriu W(DxN).
- $tf_{i,j}$ = nombre d'aparicions del terme t_j en el document d_i.
- df_j = nombre de documents on apareix el terme t_j.
- idf_j = log D/df_j

Cada fila de la matriu és la representació vectorial d'un document, i aquesta està composta per n diferents valors, els quals representen el pes de cada terme del vocabulari en aquest document. Aquest valor ve determinat per la següent fórmula:

$$w_{i,j} = tf_{i,j} \times idf_j \tag{2.5}$$

La funció de la fórmula és determinar el valor discriminatiu que té un terme dins d'un document, si el terme apareix molt en el document d_i i poc en la resta de documents tindrà un valor discriminatiu alt, però, si també està molt present en la resta de documents el seu valor discriminatiu baixara.

En altres paraules, $tf_{i,j}$ indica la importància del terme en el document i idf_j la freqüència del terme en la col·lecció de documents. Per tant, serà significatiu en $tf_{i,j}$ altes i idf_j baixes.

Aclarir, que si un terme no apareix en el document el seu valor final serà 0. Una vegada hem calculat la matriu de termes-documents amb els seus pesos, podem començar el procés de recuperació d'informació. El procés consisteix a transformar la consulta realitzada en un vector de les mateixes característiques que el d'un document, i a posteriori realitzar el càlcul de similitud, aquest càlcul es pot realitzar de diferents formes, com s'ha vist a la Secció 2.3.3.1.

Aquesta operació es realitzarà amb tots els documents, obtenint una llista ordenada de valors que representaran la similitud del document amb la consulta. Així podem tornar un nombre x de documents amb major rellevància.

Un dels principals desavantatges d'aquest model és la nul·la informació contextual que utilitza. No pot tindre en compte sinònims de paraules o paraules amb significats

pareguts depenent del context. Un altre dels problemes, és el poc ús dels significats de les paraules, és a dir, paraules polisèmiques no són tractades correctament. En conclusió, és un model que no captura ni informació contextual ni semàntica.

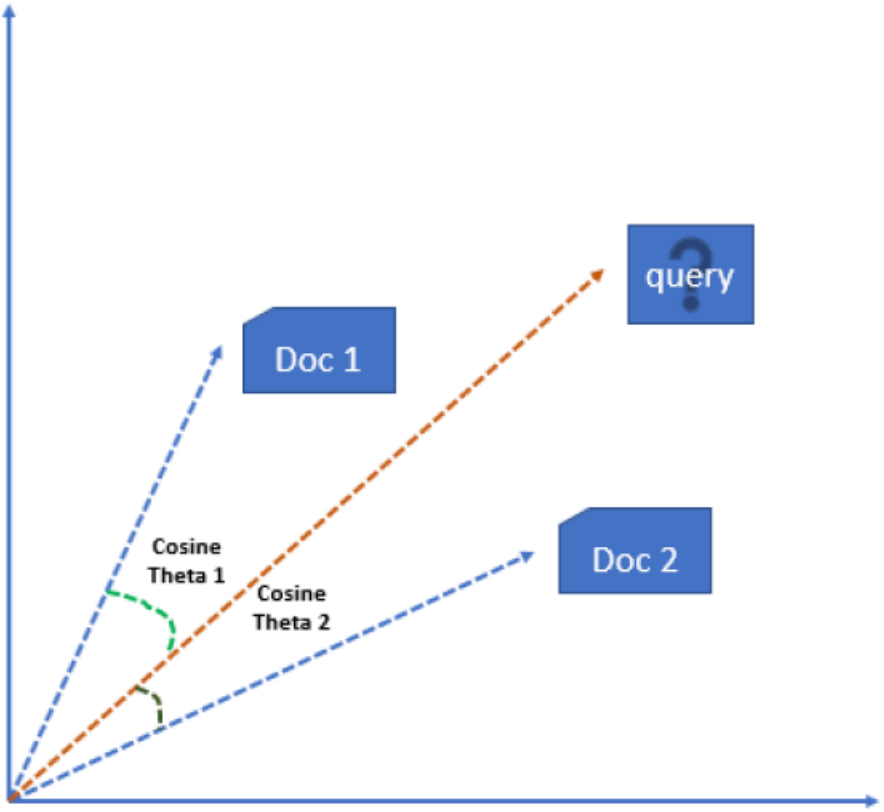


Figura 2.5: Exemple de similitud entre una consulta i dos documents, url: https://2.bp.blogspot.com/-saTZSoc5RAA/WfghS_CMvJI/AAAAAAAAGBg/PcZvT0QNZCcPJq8fAv2v_cSwrnagdm9RgCK4BGAYYCw/s1600/cosine_similarity.PNG

La Figura 2.5 mostra el procés de comparació de similitud entre 2 documents amb la consulta realitzada. Com observar, el documents 2 és el més pròxim a la consulta en l'espai vectorial atenent a la similitud cosinus, per tant, és el document amb informació més rellevant i aquell que volem recuperar en primer lloc..

2.3.5. Models amb Xarxes neuronal

En aquest subapartat anem a mirar en profunditat els models més moderns que ens podem trobar en l'actualitat. Primer anem a fer un repàs dels model més utilitzat fa 10 anys i després estudiàrem el model més modern. Aquest models no constitueixen tot el SRI, si no sols són la forma de crear les representacions vectorials de les paraules, frases, etc... Són components essencials a l'hora de crear els SRI, donat que permeten un millora a l'hora de comparar la consulta i els documents en la recuperació. Posteriorment, s'utilitza algun algorisme o component que s'encarrega de comparar aquestes representacions vectorials creades amb la representació de la consulta, per retornar aquelles més similars.

Però primer de tot hem d'entendre que és un *word embedding*.

2.3.5.1. Word Embedding

Per entendre que és un *word embeddings* anem a fer un desglossament de les seues paraules, primer de tot *word* que vol dir ´´paraula´´, i el segon *embedding* on el seu significat és ´´incrustació´´, La incrustació de paraules és la codificació del significat semàntic de les paraules en un vector dens de dimensions fixes. Un vector dens és aquell on hi ha molt pocs elements amb valor nul.

Els *words embeddings* són representacions vectorials denses de les paraules, aquests poden ser creats per diferents tècniques o elements, com les xarxes neuronals, models probabilístics, etc. Una de les seues característiques més important és el fet que paraules similars tenen representacions vectorials properes entre si.

Els *words embeddings* són molt utilitzats en l'àmbit del PLN, per exemple:

- Anàlisis de sentiments: On permeten capturar la polaritat (negativa, neutral o positiva), o d'altres elements de les paraules, que ajuden a la classificació dels sentiments.
- Traducció Automàtica: El *words embeddings* ajuden a capturar les relacions entre paraula de diferents idiomes, i on també la similitud entre elles.
- La recuperació d'informació: Tema principal del projecte, on els *words embeddings* ens permeten capturar més informació de les paraules, i, per tant, ens possibilita una comparació de similituds per trobar aquella que millor s'adequa a la consulta realitzada.

Els *words embeddings* han demostrat ser un tècnica molt útil en el món del PLN, ja que, aquestes representacions ajuden a millora l'eficiència i la precisió en moltes tasques. Tanmateix, cal criticar aspectes del seu rendiment, com per exemple, no són capaços d'entendre la polisèmia. La paraula "gat" pot tindre diferents significats depenent del context, però el seu *word embeddings* sols tindrà un dels significats, aquell més comú. Altres problemes el tenim a l'hora de representar frases, donat que, fer combinacions de WE és una feina costosa en la pràctica.

2.3.5.2. Word2Vec

El model Word2Vec utilitza el **Word Embedding** 2.3.5.1 per fer la seua representació. Aquest model intenta captar el significat contextual de la paraula o unitat bàsica en les seues representacions. En contraposició del model vectorial que tractava cada paraula independentment del context.

Creat per Tomas Mikolov i un equip d'investigadors en 2013 [13], que treballaven per a Google. Aquest model és una tècnica molt popular per a generar *word embeddings* i es caracteritza per ser una xarxa neuronal de dues capes. La funcionalitat d'aquest model és la d'extraure les representacions denses de les totes paraules de cada document del corpus. Aquestes representacions es poden utilitzar per diferents tasques, en el nostre cas les utilitzem aquests per fer un càlcul de la similitud entre paraules.

Word2Vec presenta 2 arquitectures principals:

- ***Continuous bag-of-words* (CBOW)**: En aquesta arquitectura l'objectiu de l'entrenament és predir una paraula a partir de context que l'envolta. El CBOW funciona de la següent forma:

1. **Generacions de mostres d'entrenament:** S'utilitza una finestra per anar generant mostres d'entrenament. És a dir, del text anem agafant paraules veïnes i creant les mostres, si tenim la 5ª paraula del corpus agafem la 3ª i la 6ª per generar la mostra.

2. **Codificació dels vectors**: Cada paraula es transforma en un vector *one-hot*, on la posició de la paraula en el vocabulari correspon al valor 1 en el vectors, i les altres a 0. És a dir, un vector ple de 0 a excepció de l'1 que representa la paraula

3. **El model i el seu entrenament**: S'utilitza una xarxa neuronal d'una sola capa, on l'objectiu de la xarxa serà predir una paraula objectiu depenent del context. La representació del context s'extrau de la mitjana del vectors de les paraules del context. Utilitzant una funció de pèrdua per minimitzar l'error, els pesos de les capes ocultes s'ajusten per aprendre els vectors de les paraules.

4. **Inferència**: Una vegada tenim la xarxa entrenada, els vectors resultants són representacions de les paraules en un espai vectorials on les paraules similars estan més properes.

- ***Skip-Gram***: Predicció de les paraules del context a partir d'una paraula. El model *Skip-Gram* té el següent procés:

1. **Generacions de les mostres d'entrenament:** De totes les paraules netes de la col·lecció de documents, va agafant el context amb una finestra que es desplaça, és a dir, si l'ample de la finestra és 2, considerarem dues paraules a la dreta i dues a l'esquerra. Aquest procediment es repeteix per a totes les paraules.

2. **Codificació del vectors:** Les paraules es codifiquen en vectors *one-hot*, on totes les paraules tenen un valor de 0, a excepció de la paraula objectiu que té un valor d'1.

3. **El model:** Es basa en una xarxa-neuronal d'una sola capa oculta, on per cada paraula del vocabulari tindrem una representació vectorial de dimensions fixes, que aniran canviant en l'entrenament.

4. **Entrenament:** La xarxa intenta predir el context partint d'una paraula clau, aleshores a cada iteració la xarxa està aprenent a predir millor el context per mitjà de l'aprenentatge supervisat, on es van ajustant els pesos de la xarxa per minimitzar l'error.

5. **Inferència:** Una vegada entrenada, podem obtindre els *word embeddings* des de la capa oculta de la xarxa.

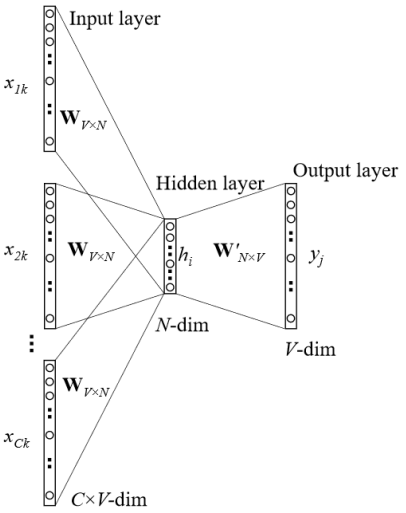


Figure 2: Continuous bag-of-word model

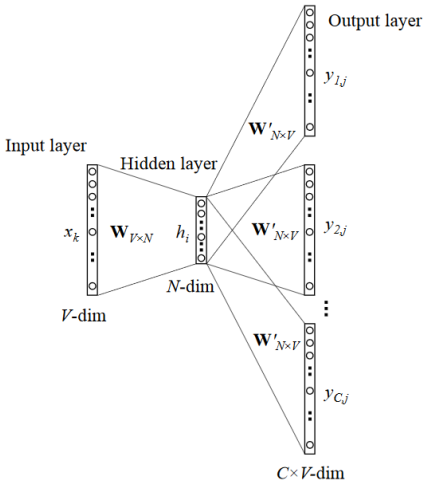


Figure 3: The skip-gram model

Figura 2.6: Vista funcional de les dues arquitectures

Ambdues arquitectures utilitzen una capa oculta per a aprendre les representacions vectorials. Com s'ha explicat en cada iteració de l'entrenament aquest pesos de la capa oculta es modifiquen per a minimitzar l'error. La principal diferència recau en el seu objectiu en l'entrenament i com són les arquitectures.

Gràcies a la seua arquitectura el model Word2Vec permet generar representacions vectorials de paraules que captures informació semàntica i contextual, amb la característica que representacions pròximes tenen les seues paraules tenen significats i contextos similars.

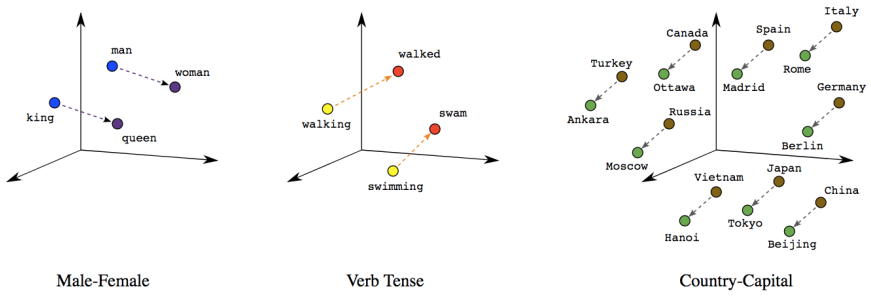


Figura 2.7: Comparació dels *embeddings* de paraules amb significat i contextos similars

La Figura 2.7 mostra com paraules amb significats similars estan en espai vectorial proper. Una altra de les característiques que mostra és com paraules que tenen relació, país amb capital, tenen una distància similar per totes aquelles que ho són. Per exemple, Espanya - Madrid té una distància igual que Itàlia - Roma en el seu espai.

2.3.5.3. Sentence-BERT

El model ***Sentence-BERT*** és, com diuen els autors, "una modificació de la xarxa preentrenada BERT que utilitza estructures de xarxes siameses per derivar els *embeddings* d'oracions semànticament significatives que es poden comparar utilitzant similitud cosinus´´ [14]. Per poder entendre millor aquest model és necessari entendre que són les

xarxes BERT i l'estructura transformer.

2.3.5.3.1 Transformers

Els *Transformers* són una arquitectura, un model d'aprenentatge automàtic, que van ser introduïts per primera volta l'any 2017 per l'investigador Vaswani i el seu grup d'investigadors, en el seu article "Attention is all you need" [15]. Aquest model presenta la innovació de substituir les capes recurrents de les *"Long short-term memory"* (LSTM) per capes d'atenció.

L'arquitectura *Transformer* està basada en els models d'atenció, aquests models permeten transformar, captar, relacions i dependències entre les paraules sense necessitat d'utilitzar capes convolucionals o recurrents. És a dir, podem captar el context que rodegen a les paraules sense la necessitat d'aquestes capes, permeten una millor paral·lelització i eficiència.

A continuació mostrarem una imatge de l'arquitectura d'un *Transformer*.

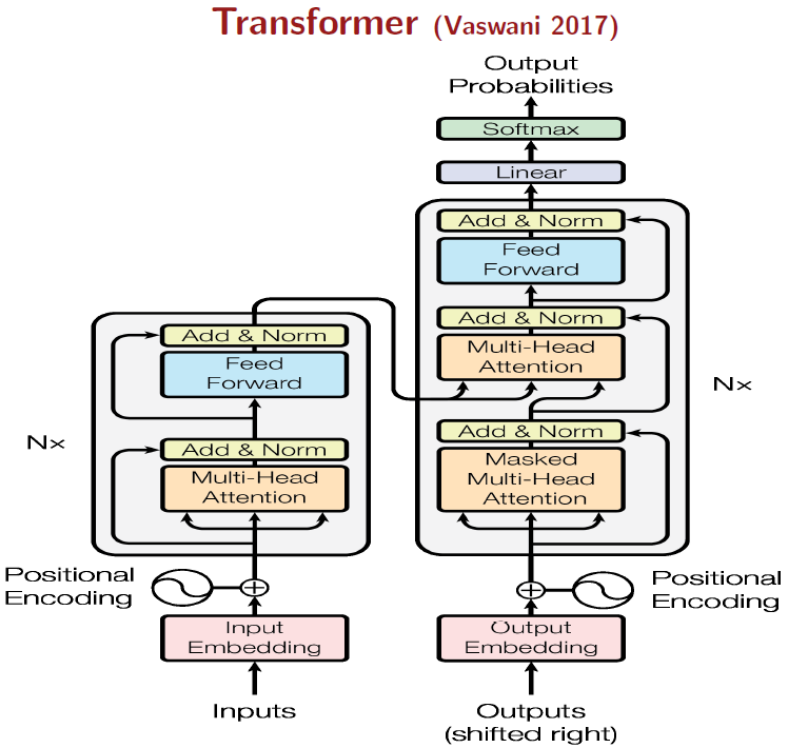


Figura 2.8: Arquitectura d'un Transformer

Anem a destacar els elements principals d'aquesta arquitectura:

- Codificador i descodificador: La funció del codificador (el primer bloc de la figura 2.8) és crear les representacions vectorials, i la funció del descodificador (segon bloc) és crear una seqüència d'eixida utilitzant les representacions.
- Bloc de la xarxa: Com podem comprovar a la figura 2.8, tant el codificador com el descodificador estan formats per diferents blocs, capes. Aquestes estan compostes per 2 subcapes: una capa d'atenció i una "feed forward", aquesta és una xarxa neuronal completament connectada.

- Capes d'atenció: Permeten que el model es puga centrar en diferents parts de la frase. El model utilitza múltiples capes d'atenció, per poder centrar-se en diferents coses i no sols en una entrada. Així, el model és capaç de trobar més tipus de relacions i característiques entre les paraules. En la figura 2.8 les capes d'atenció fan referència a les "Multi-Head attention" 2.8.

- Connexions residuals i normalització: Per solucionar el desvaniment del gradient, els *Transformers* utilitzen xarxes residuals per anar saltant capes. No sols això, sinó que al final de cada capa s'aplica una operació de normalització per millor l'entrenament de la xarxa.

- Codificació posicional: S'afegeixen als *embeddings* de les paraules per dotar-los d'informació posicional relativa a la seqüència.

Aquesta arquitectura és una de les més utilitzades, com a base, per a l'àmbit del PLN. Ha demostrat millorar els resultats de les seues predecessores, ja que, aconsegueix captar més relacions presents entre les paraules.

2.3.5.3.2 BERT

BERT [55], *Bidireccional Encoder Representations from Transformers*, és un Model de Llenguatge que permet codificar i decodificar l'entrada, generar text i fer ajustat per a diferents tasques de classificació. Va ser creat per Devlin i el seu equip de Google l'any 2018. L'aparició d'aquest model va produir una revolució en l'àmbit del PLN, ja que millorava notablement les prestacions dels sistemes. És un model basat en la tecnologia dels Transformers 2.3.5.3, encara que sols utilitza la part del codificador per a generar els *embeddings*.

Aquest model presenta una gran millora respecte als anteriors, donat que és capaç de generar els *embeddings* de les paraules dinàmicament i no estar prefixat. Permet, d'aquesta forma, considerar el context real de les paraules. Per poder entendre-ho millor, utilitzarem un exemple:

1. "El gat menja un peix".

2. "El gat és utilitzat pel mecànic".

Si parlem de significats trobem que gat té dos significats, paraula polisèmica, un de ser animal i l'altre de ser una ferramenta. Ací és on està gran part de l'interessant, aquest model és capaç de reconéixer els dos significats en calcular-se dinàmicament, en l'última capa del model calcula l'*embedding* de la frase tenint en compte tot el seu context. En contraposició, si haguérem utilitzat un model Word2Vec en compte de calcular l'*embedding* dinàmicament, haguera utilitzat el prefixat per l'entrenament de la xarxa, i com la definició de gat com animal és més comú, seria el significat majoritari, produint un possible error.

Una vegada introduït el model, passarem a un explicació del seu funcionament. Primer de tot, BERT funciona amb ***subwords***, aquest son subsegments de paraules, per poder tractar paraules desconegudes com una combinació de *subwords* coneguts, permitent reduir el vocabulari i, per tant la dimensionalitat de l'entrada de milions a milers.

BERT consta de 2 fases en l'entrenament i ús, com es pot vore a la Figura 2.9:

1. **Pre-Training**: Fase més laboriosa on el model compren el llenguatge.

2. **Fine-Tuning**: En aquesta fase el model és entrenat per a treballar en tasques específiques.

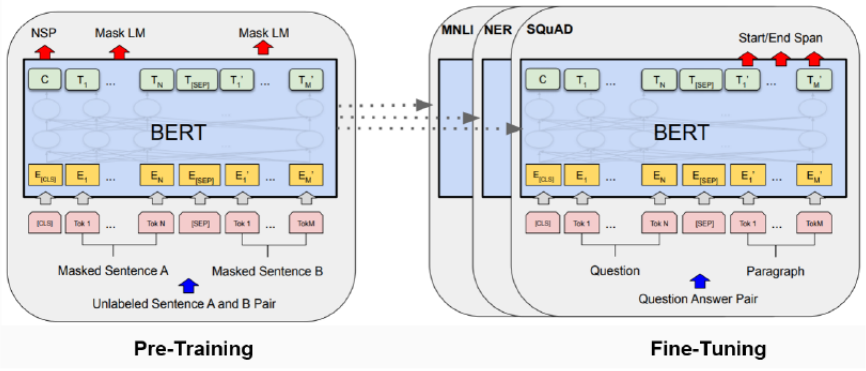


Figura 2.9: Procés d'entrenament de BERT

En aquesta primera fase d'entrenament, *Pre-Training*, trobem dues subtasques que defineixen els objectius de l'entrenament:

1. **Masked Language Model (MLM)**: Aquesta tasca consisteix a emmascarar un 15% del tokens d'entrada. D'aquest 15% es dividirà de la següent forma:

   - 80% seran transformats al token especial **[MASK]**
   - 10% es transformarà a paraules aleatòries.
   - L'últim 10% no es transformarà.

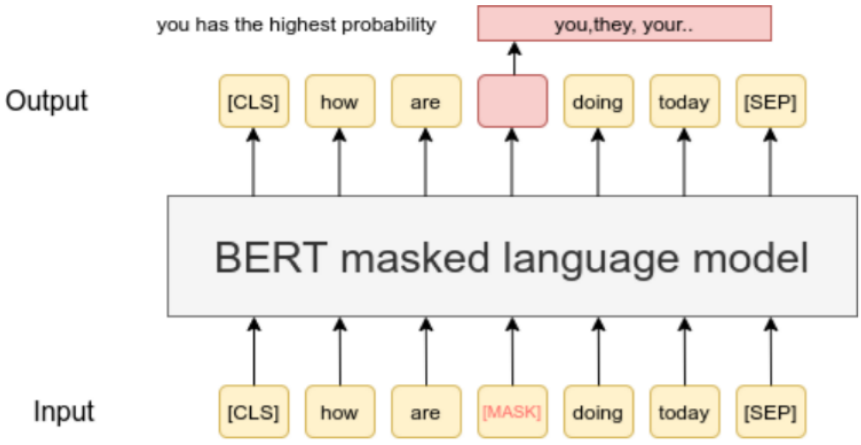


Figura 2.10: Exemple de MLM

L'objectiu és predir els tokens emmascarats d'acord amb el seu context.

2. **Next Sentence Prediction (NSP)**: L'objectiu d'aquesta tasca és fer que el model siga capaç de percebre i entendre les relacions entre les fases, és a dir, que puga estructurar correctament el text. Aquesta tècnica consisteix a predir si la frase objectiu B és la continuació a la frase A, en un 50% del casos ho serà en l'altre 50% no.

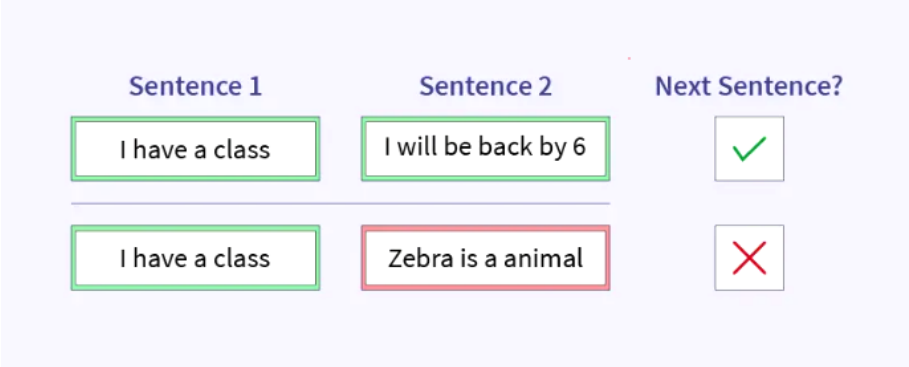


Figura 2.11: Exemple de NSP

En Fine-Tuning s'ajusta un model de llenguatge preentrenat per a resoldre una tasca específica. Donat que el model preentrenat ja té coneixements adquirits, aquests podran ajudar a resoldre millor la tasca en qüestió; és el que es coneix com a *transfer learning*. Aquest procés consisteix en l'ús de dades etiquetades de la tasca requerida, per ajustar el pesos de la xarxa. En el nostre cas l'ajustarem per poder fer similitud de frases.

En conclusió, BERT és un model que ha demostrat ser excel·lent en molts àmbits del PLN, establint un nou estàndard de referència en el camp. Gràcies a la seua capacitat de crear representacions vectorials gràcies a la capacitat per a capturar tant el significat contextual com el semàntic dinàmicament, el model és capaç d'entendre el llenguatge d'una manera que fins ara no s'havia aconseguit, millorant el rendiment en moltes tasques.

BERT resol la cerca semàntica per parells, utilitza un codificador creuat: dos oracions entren al model BERT i calcula la puntuació de similitud. Com mostra la Figura 2.12 on tenim un model BERT on entren dos oracions i per mitjà d'una capa d'avanç es calcula la puntuació de similitud. Aquest tipus de codificador té un problema, a l'hora de comparar milers d'oracions donaria com a resultats una enorme quantitat de càlculs a fer, suposant hores i hores d'entrenament.

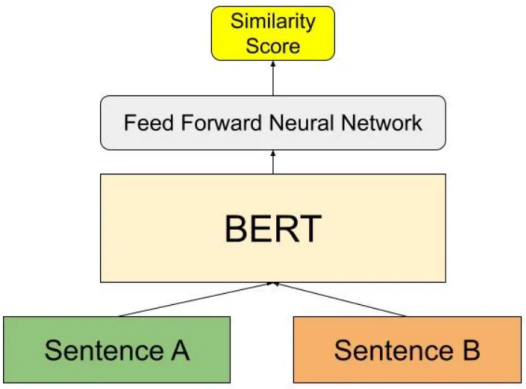


Figura 2.12: Xarxa de codificadors creuats de BERT. Url: https://towardsdatascience.com/an-intuitive-explanation-of-sentence-bert-1984d144a868

BERT tracta de solucionar el problema creant *embeddings* d'oracions, però BERT crea *embeddings* per paraules. Aleshores utilitzaren la mitjana de tots els *embeddings* de les paraules per crear les oracions, com a resultat obtingueren *embeddings* incorrectes amb un rendiment molt baix.

El model SBERT va ser creat per Nils Reimers and Iryna Gurevych l'any 2019. SBERT és un model que soluciona un dels principals desavantatges dels BERT, la comparació de

similituds entre dues frases. Com diuen els autors: [12]"Trobar la parella més semblant en una col·lecció de 10.000 frases requereix uns 50 milions de càlculs d'inferència (65 hores) amb BERT. La construcció de BERT el fa inadequat per a la similitud semàntica".

SBERT és similar al codificador creuat 2.12 però eliminant el capçal final. SBERT utilitza una arquitectura siamesa que contenen 2 models BERT idèntics i que comparteixen pesos. Aquesta arquitectura és capaç d'obtindre dos nivells de codificació, un a nivell de paraula i l'altra com a frase.

L'ús de xarxes siameses és causat pel fet que són entrenades calculant similituds entre dues coses. En conseqüència, resulta fàcil agregar noves classes o comparar dos coses on un element no ha sigut vist. SBERT utilitza un concepte denominat *triplet loss* per entrenar la xarxa siamesa.

2.3.5.3.3 Triplet loss

Per explicar aquest concepte utilitzarem la tasca de trobar imatges similars. Tenim una imatge, la xarxa ha de trobar una imatge negativa (pertany a una classe diferent) i una positiva (pertany a la mateixa classe). Una vegada trobades, la xarxa calcula una puntuació de similitud entre la imatge elegida i les altres 2. La xarxa agafa aquest dos *scores* per calcular un *score* de pèrdua, que l'utilitza per actualitzar els pesos de la xarxa.

Totes les imatges del corpus són assignats aquest grups, és a dir, tota imatge té el seu par positiu i negatiu. Per això la xarxa es pot entrenar amb cada imatge, sense la necessitat d'utilitzar la força bruta, donat que no necessita entrenar-se en tota combinació d'imatge.

SBERT utilitza aquesta tècnica, però amb oracions, fent que un par positiu siga aquell que té una distància xicoteta i un par negatiu siga aquell que té una distància gran.

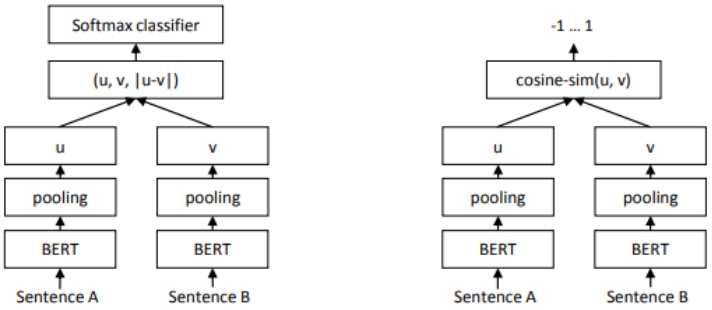


Figura 2.13: Entrenament i inferència del model SBERT

El procés d'entrenament de SBERT el poden visualitzar en la primera part de la Figura 2.13. SBERT utilitza dues oracions a l'hora de l'entrenament. L'oració A anirà al 1ª BERT i l'oració B al 2ª BERT, cada BERT generarà una representació vectorial de l'oració per mitjà de la tècnica *pooling*, tècnica que generalitza característiques d'una xarxa, en aquest cas, funciona agafant les representacions del model BERT i fent la mitjana.

Una vegada hem acabat la fase del *pooling*, tenim dos representacions: 1 per a la frase A i altra per a la frase B. Aquestes representacions seran concatenades, i per mitjà de tècniques d'aprenentatges supervisades entrenarem un classificador softmax.

En aquest entrenament el model utilitzarà el resultat de la funció de pèrdua per anar ajustant els paràmetres, per maximitzar similituds de les representacions de frases similars i minimitzar aquelles de frases poc similars. Aquest procés ajudarà al model a

entendre millor la semàntica, permetin crear representacions vectorials que reflecteixen la similitud en l'espai semàntic.

En la fase d'inferència, el model calcula la similitud de les frase per mitjà de la similitud cosinus 2.3.3.1, que generarà un puntuació de similitud. Procés representat en el segon bloc de la imatge 2.13

En conclusió, el model SBERT permet utilitzar tota la potència semàntica i contextual que teníem en els models BERT per ser capaços de comparar la similitud entre frases. Aquest model ha permés millorar en tasques on la comprensió de l'oració completa és crucial.

2.3.5.4. Model XLNeT

El model XLNet és un model de llenguatge pre-entrenat basat en l'arquitectura de *Transformer-XL*, va ser creat per Zhilin Yang i el seu equip d'investigadors al gener de l'any 2020 [16], aquest model va eixir per resoldre els problemes que tenia BERT. Per poder entendre millor aquest model hem de presentar d'algunes de les característiques dels *Transformers-XL*.

El *Transformer-XL* és una extensió del *Transformer* original, en la que es resolen les limitacions de les dependències de llarg termini, donat que no tenen límit en la grandària de l'entrada. Gràcies a la seua atenció desenvolupada (dona la capacitat de recordar informació de paraules anteriors en la seqüència, encara que estiguen molt separades) i memòria contextual recurrent (emmagatzematge d'estats anteriors, permeten conservar molta més informació del text i millorant l'atenció i la comprensió d'aquest) és capaç d'entendre millor les dependències contextuals i manejar seqüències més llargues, permetent una millor representació vectorial.

XLNet agafa les següents tres característiques dels *Transformer-XL*:

- Codificació posicional: Especifica la posició de cada token en la seqüència
- Recurrència del segment: Emmagatzema l'estat ocult del primer segment en la memòria de cada capa i actualitza l'atenció en conseqüència.
- Modificació atenció: En el XLNet sols observen la representació oculta del tokens que precedeixen al token processat. És a dir, si processem el token 3, sols podrem agafar la informació dels tokens anteriors i la seua posició.

Aprofundint en el XLNet, trobem que és un model autoregressiu generalitzat on el token a processar sols depén dels anteriors tokens. Es diu un model generalitzat perquè captura el context bidireccional (tant esquerre com dret) per mitjà de la tècnica **Modelatge del llenguatge de permutació (PLM)**, aconseguint integrar en un mateix model la idea de l'autoregressió i el context bidireccional.

La tècnica PLM consisteix a capturar el context de la paraula bidireccionalment per mitjà de totes les permutacions possibles de la paraula en una frase. És a dir, en comptes d'agafar el context d'esquerra a dreta o viceversa, utilitzem totes les possibles permutacions d'una mateixa frase. Amb l'objectiu, que tota posició aprén la informació contextual de les altres posicions. Amb aquesta tècnica no hi ha cap necessitat d'utilitzar **[MASK]** i ni de danyar les dades d'entrada.

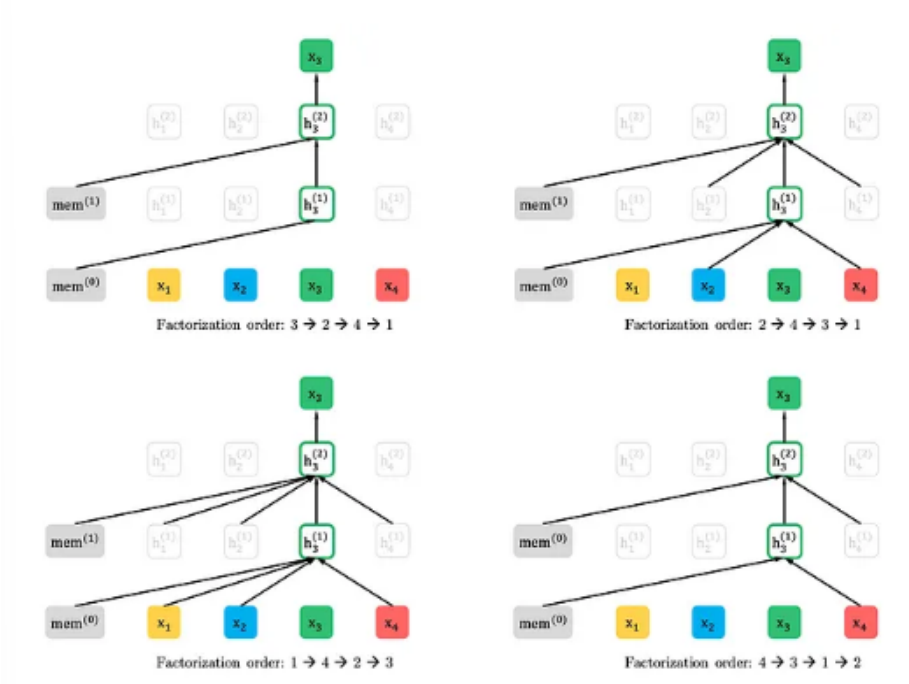


Figura 2.14: Exemple de la tècnica de PLM

En la Figura 2.14 l'objectiu és predir $x3$. Primer de tot el model crea totes les possibles permutacions de la seqüència, després per cada seqüència sols té en compte aquells token que precedeixen a x3 en les capes ocultes, per assolir l'autoregressió. Al final, després de processar totes les permutacions, hem aconseguit aprendre $x3$ partint de les altres paraules de la seqüència.

A continuació mostrarem un exemple de les diferències entre BERT i XLNet.

L'oració exemple és [New York is a city] i volem predir ´´New York", i la permutació actual és la següent:

Figura 2.15: Permutació actual de "New York is a city"

Aleshores per cada model intentaria predir els tokens de la següent forma:

- BERT no permutaria l'oració (´´[[M] [M] is a city]´´) on deixaria fixe els token:

$$\log P(\text{New} \mid [M]\ [M] \text{ is a city}) + \log P(\text{York} \mid [M]\ [M] \text{ is a city})$$

  Per tant, podria predir ´´New Francisco is a city".

- XLNet, al ser autoregressiu, prediu la seqüència en ordre, és a dir, primer 4 i després 5, obtenint:

$$\log P(\text{New} \mid \text{is a city}) + \log P(\text{York} \mid \text{New, is a city})$$

En conclusió, el model XLNet pot millorar els resultats en moltes de les tasques on anteriorment s'utilitzava BERT, gràcies al seu nou enfocament que li permet captar millor les dependències a llarg termini i millorar la comprensió del context global. En canvi, a l'hora dels resultats, no mostra una millora respecte a BERT.

2.3.5.5. MPNet

El model MPNent (Multilingual Pre-trained Model with a Cross-lingual Encoder) és un nou model de llenguatge pre-entrenat, va ser creat per un equip d'investigadors de Microsoft, més concretament per Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu i Tie-Yan Liu [17]. La principal característica d'aquest model és que hereta els avantatges tant de la tècnica MLM (Masked Language Model 2.3.5.3) com del PLM (Modelatge del llenguatge de permutació) 2.3.5.4, evitant les limitacions de cadascun.

Analitzant les tècniques MLM i PLM trobem els seus avantatges i desavantatges:

- MLM: Té al seu abast la informació posicional completa de la seqüència, però no pot comprendre les dependència a llarg termini dels termes, per tant, no aprén bé les relacions semàntiques.
- PLM: Pot modelar la dependència a llarg termini i comprendre les relacions semàntiques, però no té al seu abast la informació posicional que provoca discrepàncies entre el pre-entrenament i el fine-tuning.

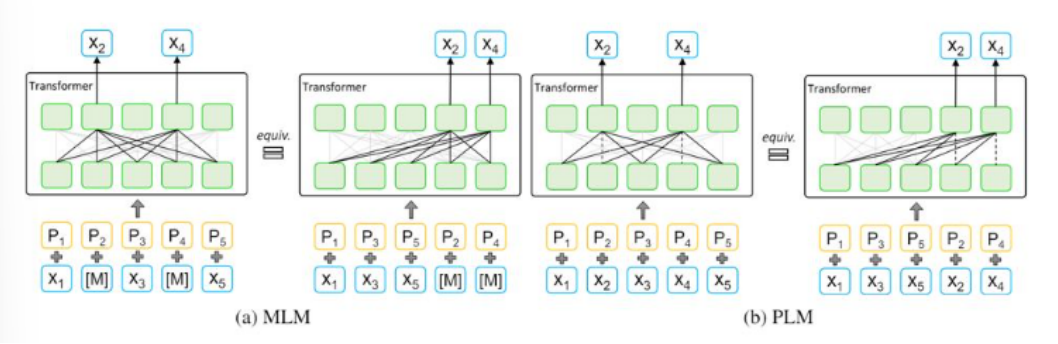


Figura 2.16: (a) *Masked Language Model* (MLM) i (b) *Permuted Language Model* (PLM), la figura dreta tant en (a) com (b) és la vista unificada del dos

Per fer-ho més fàcil d'entendre numerarem cada figura. figura 1, la de l'esquerra del tot que té com entrada $(x1, M, x3, M, x5)$, la figura 2, la de la seua dreta que com entrada té $(x1, x3, x5, M, M)$; la figura 3 que com entrada té $(x1, x2, x3, x4, x5)$ i l'última, la de la dreat del tot, que com entrada té $(x1, x3, x6, x2, x4)$.

Per poder heretar els avantatges, hem d'analitzar les tècniques amb una vista unificada. Cada model utilitza com estructura base els *Transformers*, aquest son insensible a l'ordre del tokens si agreguem informació posicional. D'aquesta forma, si afegim informació posicional, afirmem que les dos figures de (a) (figura 1 i figura 2) de la Figura 2.16 són equivalents, igual que les dos figueres de (b) (figura 3 i figura 4), és així, que passem a considerar la figura 2 i figura 4 com la vista unificada. Per tant, podem dividir l'entrada de MLM i PLM en una part no predita i una part predita respectivament. Com a conseqüència, podem veure com la part no predita són iguals tant en MLM i PLM, però la part predita diferent.

Una vegada hem creat la vista unificada el procés del MPNet és el següent:

1. Primer fusionem la part no predita de MLM i PLM. Exemple, tenim la cadena $(x_1, x_2, x_3, x_4, x_5, x_6)$ i la permutem a la següent $(x_1, x_5, x_3, x_4, x_6, x_2)$. Dividim aquesta cadena en dos parts: la part predita (x_4, x_6, x_2) i la part no predita $(x_1, x_5, x_3, [M], [M], [M])$ a la qual s'afegís la informació posicional $(p_1, p_5, p_3, p_4, p_6, p_2)$. Com es mostra en la figura següent:

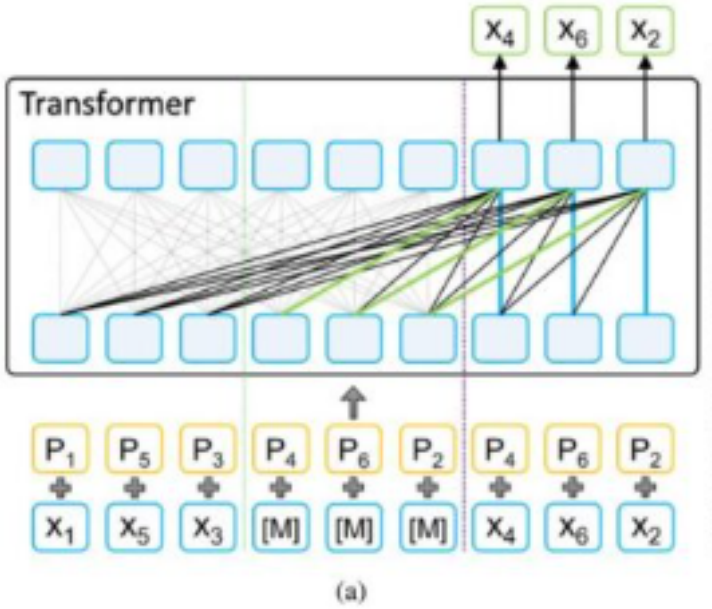


Figura 2.17: estructura del model MPNet

2. Modelar la dependència d'eixida: Adoptem dos fluxos d'autoatenció que estan en PLM per la generació autoregressiva. Exemple: volem preedir x_6, el model pot vore $(x_1 + p_1, x_5 + p_5, x_3 + p_3)$ de la part no predita i, també, el pronosticat amb anterioritat $(x_4 + p_4)$ de la part predita. Així aconseguim evitat la falta d'informació de dependències en MLM.

3. Modelar la coherència d'entrada: Per ser coherent l'entrenament amb les tasques posteriors, utilitzarem l'emmascarament i la informació posicional. Exemple: si volem predir x_6, no sols veu l'anterior $(x_1 + p_1, x_5 + p_5, x_3 + p_3)$ i $(x_4 + p_4)$, sinó que també veu $([M] + p6, [M] + p2)$. Així el model pot agafar la informació posicional completa cada volta que prediu un token. Aconseguint evitar la falta d'informació posicional de PLM.

Com a conseqüència, el procés del MPNet uneix les dues millors característiques dels models anteriors evitant així els desavantatges. MPNet és un model que ha demostrat un fort rendiment en gran varietat de tasques PLN en múltiples idiomes. La seua capacitat d'entendre el context i la semàntica el fan realment útil en gran varietat d'escenaris.

2.3.5.6. Comparació entre BERT, XLNet, MPNet

En aquest subapartat anem a fer un repàs de les principals diferències dels tres models i de les seues prestacions.

La primera comparació que realitzarem és la diferència entre la informació utilitzada per aconseguir els objectius en el preentrenament. És a dir, com el MLM, PLM i MPNet

utilitza la informació per aconseguir el objectius. En la Taula 2.5 es pressuposa que cada objectiu prediu un 15% de tokens. Podem veure com el MPNet com agafa el millor de cada món per obtindre els millors resultats.

Objectius	Tokens	Posicions
MLM (BERT)	85%	100%
PLM (XLNet)	92.5%	92.5%
MPNet	92.5%	100%

Taula 2.5: Informació usada dels diferents objectius de preentrenament

La Taula 2.5 mostra com MPNnet utilitza un 92,5% de la informació que aporten els tokens i un 100% de la informació que aporta el posicionament. Els percentatges fan referència al total d'informació que utilitza.

La següent comparació és la diferència a l'hora d'analitzar una seqüència. Suposem "[The, task, is, sentence, classification]" [17] i necessitem predir "[sentence, classification]", per cada model obtindrem una factorització diferent 2.6.

Models	Factorització
BERT	log P (sentence | the task is [M] [M]) + log P (classification | the task is [M] [M])
XLNet	log P (sentence | the task is) + log P (classification | the task is sentence)
MPNet	log P (sentence | the task is [M] [M] + log P (classification | the task is sentence [M])

Taula 2.6: Factorització dels diferents models

On, per exemple, MLM (BERT) fa la predicció [sentence, classification] de forma independent, per tant, pot portar a l'error de predir el segon token com "answering" i fer una predicció final com "[sentence answering]". PLM(XLNet) no té la informació posicional, per tant, podria predir tres tokens [oració, parell, classificació]. MPNet en tindre tant la informació posicional, saber que hi ha 2 tokens a predir, com les dependències entre el tokens, pot conseguix un millor resultat.

Experimentació

Finalment, l'equip d'investigador que ha creat el model MPNet, ha realitzat diferents experiments [17] utilitzant un corpus pensant per a l'avaluació la capacitat dels models de llenguatge en tasques de comprensió del llenguatge, es tracta de *The General Language Understanding Evaluation* (GLUE) *Benchmark* [18]. Aquest està compost per 9 tasques diferents en l'àmbit de la comprensió del llenguatge natural, 2 tasques d'una sola frase (CoLA [19], SST-2 [20]), 3 tasques de similitud i parafrasejar (MRPC [21], STS-B [22], QQP), 4 tasques d'inferència (MNLI [23], QNLI [24], RTE [4], WNLI [25]). La Taula 2.7 mostra els resultats dels tres models sobre les 9 tasques:

Models	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLa	STS	Avg
BERT	84.5%	91.7%	91.3%	68.6%	93.2%	87.3%	58.9%	89.5%	83.1%
XLNet	86.8%	91.7%	91.4%	74.0%	94.7%	88.2%	60.2%	89.5%	84.5%
MPNet	**88.5%**	**93.3%**	**91.9%**	**85.2%**	**95.4%**	**91.5%**	**65.0%**	**90.9%**	**87.7%**

Taula 2.7: El resultats de cada madel en el *Benchmark GLUE*

Com poder comprovar, el millor model en totes les tasques ha sigut el MPNet, és per això que, serà el model utilitzat per a crear les representacions vectorials en aquest projecte.

2.3.6. Exemples de SRI

En aquest subapartat mostrarem exemples d'implementacions de SRI al món actual. Aquests són:

- Motors de cerca web: Aquest motors de cerca són els SRI més utilitzats hui en dia, permeten als usuaris realitzar cerques d'informació en la web per mitjà d'una consulta, retornant aquelles pàgines webs més rellevants. El exemples més importants són Google, Bing, entre d'altres Yahoo.

Figura 2.18: Motors de cerca web

- Sistemes de recuperació de documents: Utilitzats per cercar i recuperar documents d'una col·lecció específica, d'un corpus privat, on l'usuari per mitjà de paraules clau fa la consulta i obté els resultats més rellevants. Com a exemples tenim les biblioteques digitals, base de dades d'investigació i repositoris acadèmics.

Figura 2.19: repositoris de documents d'investigació

- Sistemes de recomanació: Aquests sistemes utilitzen les preferències i comportaments dels usuaris com a consulta, per, posteriorment, retornar o recomanar elements rellevants o similars a les preferències. El exemples més característics són plataformes de *streaming* tant de música com de pel·lícules.

Figura 2.20: Plataformes de *streaming*

- Sistemes de cerca interna: Moltes pàgines web o xarxes socials, implementen sistemes de recuperació d'informació dins del seu àmbit, aconseguint tindre un SRI específic a les seues característiques. Una altra característica que comporten és la possibilitat de cercar per *hashtag*, perfils d'usuari o documents sencers. Els exemple que trobem són Twitter, Instagram o llocs web. La mateixa UPV té un SRI de cerca interna en la web.

Figura 2.21: Xarxes sócials

2.4 Avaluació del Sistemes de recuperació d'informació

Els SRI com qualsevol altre sistema és susceptible de ser avaluat, com diu Blair "és la mateixa naturalesa dels SRI la que propicia la seua necessitat crítica d'avaluació, just com qualsevol altre camp de treball que aspire a ser classificat com camp científic"[5].

Molts autors exposen que dins de l'avaluació existeixen dos factors determinants, el "accés físic", que se centra en la forma de recuperar la informació i la presentació a l'usuari, i "l'accés lògic", la localització de la informació demandada. Blair il·lustra aquestes reflexions amb aquest exemple "considerem una biblioteca: descobrir on es troba un llibre amb un tòpic determinat és un problema relacionat amb l'accés físic de l'objecte (el llibre); descobrir que llibre pot informar-nos sobre la determinada matèria és un proble-

ma d'accés lògic" [5]. On tots els autors estan d'acord és en el fet que el segon terme, "accés lògic", és el més important, ja que, tracta sobre la rellevància de l'objecte.

Seguint aquesta lògica, Baeza-Yates afirma que existeixen dos tipus d'avaluacions:

- "Quan s'analitza el temps de resposta i l'espai necessari per a la gestió, s'estudia el rendiment de les estructures de dades utilitzades en la indexació dels documents, la interacció amb el sistema, els retards de les xarxes de comunicació i qualsevol altre retard addicional introduït pel *software* del sistema. Aquesta Avaluació es podria denominar simplement com avaluació del funcionament del sistema"[11]

- "En els SRI els documents recuperats no seran respostes exactes a la petició, la pregunta pot estar vagament formulada. Els documents recuperats es classifiquen segons la rellevància cap a la pregunta, com de relacionat està amb la temàtica de la pregunta el conjunt de documents retornats. Aquest tipus d'avaluació es coneix com a avaluació del funcionament de la recuperació". [11]

2.4.1. Rellevància

Com hem vist en l'apartat anterior hem de mesurar la recuperació de documents amb la rellevància d'aquests, però que significa rellevància. La definició del terme rellevància és un dels grans problemes en l'avaluació del SRI, donat que aquest concepte és la base de moltes de les mesures que s'apliquen en els SRI.

El concepte de rellevància ha sigut estudiat des de molt punts de vista [41]: lògic, filosòfic, psicològic, etc. Concloent en dos enfocaments principals:

- Rellevància objectiva: Enfocada en els sistemes, la qual indica com la informació recuperada coincideix amb la consulta realitzada.

- Rellevància subjectiva: Aquesta se centra en l'usuari, Boyce diu "la rellevància subjectiva s'estudia segons la informació nova que aconsegueix l'usuari, per tant, la informació ja coneguda no és important [49]. Schamber la defineix com "la utilitat o potencial ús dels materials recuperats amb relació a la satisfacció dels objectius, interessos, treball o problemes de l'usuari."

Existeixen altres autors que determinen que la rellevància es troba en un punt mitjà, tenint tant components subjectius com objectius, és així, com Barry determina set criteris de rellevància per a documents [42]:

1. Informació que conté el document.
2. Experiència prèvia de l'usuari.
3. Creences i preferències de l'usuari.
4. Altres informacions i fonts.
5. Fonts del document.
6. Document com entitat física.
7. Situació de l'usuari.

Obtenint 2 criteris objectius (1,5) i 5 subjectius (1,2,4,6,7).

Molt lligat al concepte de rellevància trobem el de pertinència, moltes voltes arribant a mesclar-se i confondre's. Korfhage exposa que "la rellevància és la mesura de com una pregunta s'ajusta a un document i pertinència és la mesura de com un document s'ajusta a la necessitat informativa"[43]. Segons aquest autor, la diferència entre els conceptes radica en com realitzem la consulta d'informació. Per tant, la consulta presenta dues dificultats: ha de ser el reflex de la necessitat d'informació i adequada per poder trobar els documents necessaris.

Tanmateix, trobem que la valoració de la pertinència és més complicat pel fet que és l'usuari qui ha de valorar si un document s'ajusta a la seua necessitat.

2.4.1.1. Càlcul de la rellevància

La mètrica més utilitzada pel càlcul de la rellevància és utilitzar valors binaris, és a dir, un 1 si és rellevant o un 0 si no ho és. Algun autors han proposat una escala de rellevància, com Keen [44] que utilitza una escala de 4 valors, o Saracevic [45] que utilitza una de tres valors (rellevant, parcialment rellevant, no rellevant). Totes aquesta escales tenen el mateix problema: poder fer la distinció entre rellevant i parcialment rellevant és molt complicat.

Actualment, el càlcul de la rellevància es fa per mitjà de dos mètodes:

- **Manual**: Consisteix en l'avaluació manual dels documents retornats, per poder saber si s'ajusten a la consulta o no. Presenta diversos problemes, com la necessitat de ser més d'una persona a la valoració dels resultats, col·leccions molt grans requereixen invertir una gran quantitat de temps i diners. Per poder resoldre'ls existeixen col·leccions de documents especialitzades, que contenen una quantitat mitjana de documents i tots relacionats en la mateixa àrea. Un exemple és Crandfield [46] on els autors d'articles fan preguntes on la resposta són els seus propis articles.
- **Polling**: Utilitzat per a grans col·leccions de documents, consisteix en el fet que un grup d'experts analitzen els primers x (nombre gran) documents retornats per diferents sistemes, que indiquen si són rellevants o no. Assumint que la gran majoria de documents rellevants són retornats, per tots o almenys per alguns sistemes, i els no recuperats per cap sistema són considerats com irrellevants.

  Aquest sistema és el que s'utilitza en *Text Retrieval Conference* (TREC) des de 1994 [47]

2.4.2. Principals mesures d'avaluació

Una vegada definit el concepte de rellevància relacionant-ho amb la recuperació d'informació, podem passar a establir una seria de mesures que ens serviran per a avaluar els documents.

2.4.2.1. La precisió

La precisió és definida per Salton com "la proporció de material recuperat rellevant del total dels documents recuperats"[1]. Aquesta definició va ser completada per Frakes, el qual exposa "el resultat de l'operació està compres entre 0 i 1"[48]. La recuperació perfecta obté un valor d'1.

Aquesta mesura, avalua directament la correlació entre la consulta i la col·lecció de documents i indirectament l'algoritme d'indexació. Si aquest algoritme generalitza molt a l'hora de la indexació obtindrem una precisió baixa (dona igual la similitud).

La fórmula de la precisió és la següent:

$$\text{Precisió} = \frac{\text{Documents Rellevants Recuperats}}{\text{Total Documents Recuperats}} \tag{2.6}$$

Si analitzem l'equació, trobem que aquesta mesura està relacionada amb el soroll, com més pròxim al 0 tenim més documents inservibles, per tant, més soroll. Tanmateix, com els documents estan ordenats per rellevància, generalment com més documents recuperem pitjor serà la precisió, no obstant depén del funcionament del rànquing. Aquest fenomen el podem observar en la següent figura.

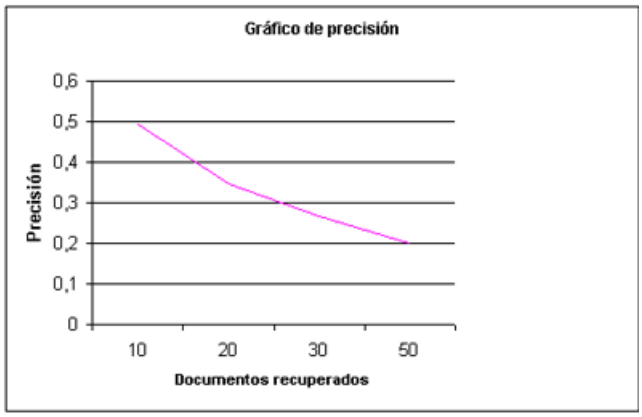


Figura 2.22: Precisió en variar el nombre de documents recuperats. Raquel Gómez Díaz. L'avaluació en recuperació d'informació. "Hipertext.net", núm. 1, 2003. <https://arxiu-web.upf.edu/hipertextnet/numero-1/evaluacion$_r i.html$ >

L'eix de les x marca en nombre de documents recuperats i el de les y l'exhaustivitat obtinguda. Com es pot observar, a major nombre de documents recuperats menor és la precisió obtinguda.

2.4.2.2. L'exhaustivitat

Aquesta mesura, en anglés denominada "recall", és la relació entre els documents rellevants recuperats i el total de documents rellevants. A continuació, mostrarem la fórmula:

$$Exhaustivitat = \frac{\text{Documents Rellevants Recuperats}}{\text{Total Documents Rellevants}} \tag{2.7}$$

En aquesta mesura, en obtindre un valor proper a 1 significa que hem recuperat la gran majoria de documents rellevants.

Com hem vist amb la precisió, l'evolució de l'exhaustivitat també es pot representar gràficament.

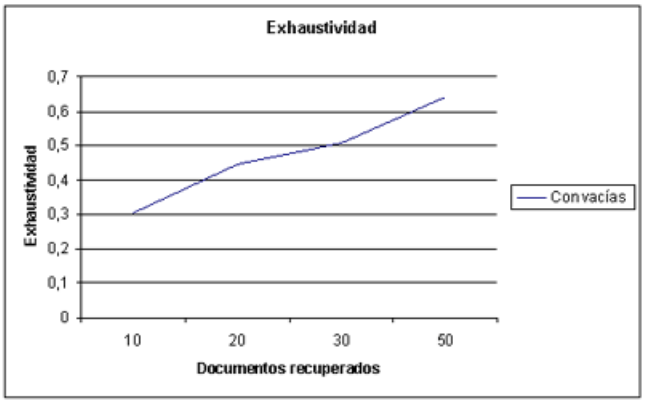


Figura 2.23: Exhaustivitat en variar els nombres de documents recuperats. Raquel Gómez Díaz. L'avaluació en recuperació d'informació. "Hipertext.net", núm. 1, 2003. <https://arxiu-web.upf.edu/hipertextnet/numero-1/evaluacion$_r$*i.html* >

L'eix de les x marca en nombre de documents recuperats i el de les y l'exhaustivitat obtinguda. Com es pot observar, a major nombre de documents recuperats major és l'exhaustivitat obtinguda. No podem utilitzar-la com única mesura, donat que, si retornem tots els documents en cada consulta obtindrem un 1. Com a conseqüència, és necessari l'ús de mesures que calculen la quantitat de documents no rellevants retornats, com és la precisió.

2.4.2.3. Relació entre precisió i exhaustivitat

Per tindre un SRI adequat és necessari que la precisió i l'exhaustivitat estiguen compensades, per poder comprovar-ho podem fer-ho de diferents formes:

- Calculant la precisió exhaustiva puntual: Aquesta tècnica consisteix en cada iteració calcular la precisió i exhaustivitat actual. Per exemple, recuperem 10 documents, 8 d'ells rellevants, aleshores en la primera iteració si el document és rellevant, tindrem 1/1 de precisió i 1/8 d'exhaustivitat.

- Càlcul de l'exhaustivitat i la precisió per trams: Tècnica que calcula l'exhaustivitat en trams, si tenim 20 documents, calcula l'exhaustivitat en el document 5, en el 10, 15 i 20.

Una vegada has obtingut els punts pots recrear un gràfica per fer la comparació de sistemes, com mostrem a continuació. On l'eix x marca els valors de l'exhaustivitat i l'eix y els valors de la precisió, per tant, per cada punt que hem obtingut (punt 1 (0,65 precisió, 0,1 exhaustivitat)) anem marcant-ho en la gràfica. Al final obtenim una corba de com es relacionen els dos conceptes en cada sistema, aconseguint una comparació directa entre els sistemes.

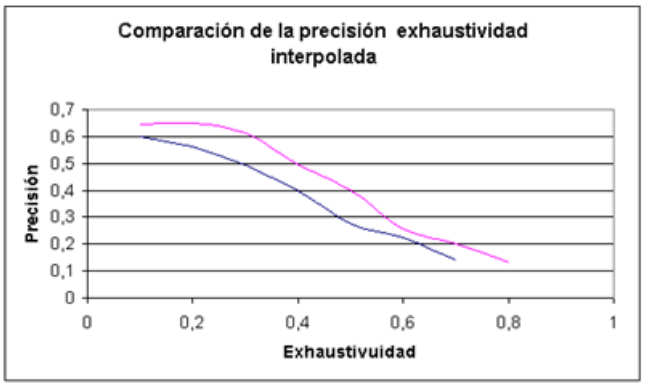


Figura 2.24: Comparació de la precisió exhaustiva interpolada

Per finalitzar, Blair va establir el criteri de "punt de futilitat", on l'usuari marca un punt on els documents deixen d'importar, per tant, les dues mesures es calcularan fins aquest punt [40].

2.4.2.4. Proposició de Fallada

La proposició de fallada o *fall-out*, és una mesura que relaciona el documents no rellevants recuperats i tots els documents no rellevants. La relació és la següent:

$$Fall - out = \frac{\text{Documents No Rellevants Recuperats}}{\text{Total Documents No Rellevants}} \tag{2.8}$$

Aquesta mesura comprén valors entre 0 i 1. Aquells sistemes amb un *Fall-out* igual a 0 o proper, són sistemes que no recuperen cap document irrellevant, és a dir, sistemes bons a priori, encara que no necessàriament, faltaria obtindre més mesures. Tanmateix, aquells sistemes propers a l'1, són sistemes roïns, a priori, donat que recuperen una quantitat gran de documents irrellevants.

2.4.2.5. Mesura F

Aquesta mesura és la mitjana harmònica o el balanç entre precisió i exhaustivitat:

$$F = \frac{(1+\beta^2) \cdot \text{Precisió} \cdot \text{Exhaustivitat}}{\beta^2 \cdot \text{Precisió} + \text{Exhaustivitat}} \tag{2.9}$$

El factor β s'utilitza per a ponderar les mesures. Si el factor β és major que 1 fem èmfasis en l'exhaustivitat, i per donar major importància a la precisió utilitzem valors menors que 1.

L'equació de baix fa referència a la mesura F_1, on precisió i exhaustivitat tenen pesos compensats.

$$F = \frac{2 \cdot \text{Precisió} \cdot \text{Exhaustivitat}}{\text{Precisió} + \text{Exhaustivitat}} \tag{2.10}$$

2.4.2.6. Precisió en K

En els sistemes moderns on la quantitat de documents rellevants és tan gran, l'usuari perd interés a llegir-los, és per això, que existeix aquesta mesura. La Precisió en k, $P(k)$,

calcula el nombre de documents rellevants dintre dels k primers documents recuperats. És a dir, dins del top k de documents recuperats quants són rellevants. Exemple:

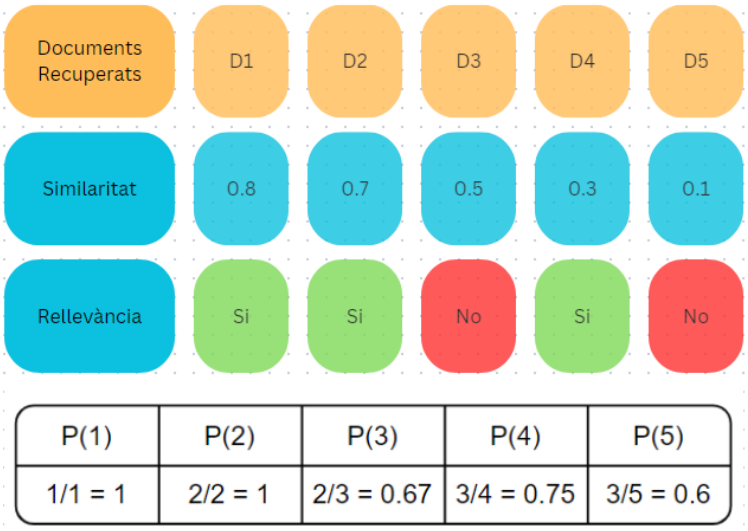


Figura 2.25: Exemple de càlcul de P(k)

La Figura 2.25 mostra el càlcul de la mesura $P(5)$, on hem recuperat 5 documents, per cada documents calculem la similitud amb la consulta i, per acabar si són rellevants o no. Finalment, amb aquestes dades podem fer un càlcul de $P(5) = 0.6$.

2.4.2.7. Precisió Mitjana

Aquesta mesura, anomenada en anglés ***Average Precision (AP)***, és una forma de resumir la corba precisió-exhaustivitat, vista en la Secció 2.4.2.3, que relaciona la precisió amb l'exhaustivitat d'un sistema, en un únic valor. Aquest valor és l'àrea baix la corba.

Aquest valor es calcula per mitjà de la següent fórmula:

$$AP = \frac{\sum_{k=1}^{K=n} \text{Rel}(k) \cdot \text{P}(k)}{\text{Total Documents Rellevants}} \tag{2.11}$$

On n el nombre de documents recuperats, Precisió(k) és la precisió en k, i Rel(k) ens indica la rellevància del document k (0 si no ho és, 1 si ho és). A continuació mostrem un exemple de càlcul:

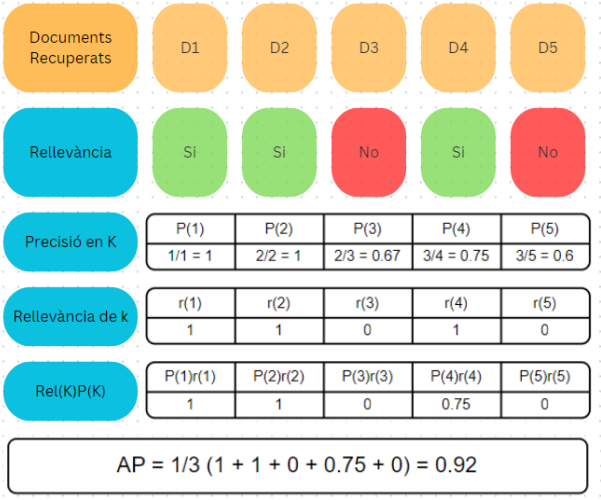


Figura 2.26: Exemple de càlcul de la Precisió Mitjana

La Figura 2.26 mostra el càlcul de la mesura Precisió Mitjana, on hem recuperat 5 documents, on sabem quins són rellevants i quins no. A partir d'aquesta informació anem calculant les diferents variables per poder extraure el resultat final.

2.4.2.8. Mitjana de la Precisió Mitjana

Mesura coneguda com a MAP (*Mean Average Precision*), que calcula la precisió mitjana de cada consulta per a un conjunt de consultes. La fórmula és la següent:

$$MAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q} \tag{2.12}$$

On Q és el nombre de consultes realitzades.

2.4.2.9. Guany Acumulat Descomptat Normalitzat

El guany acumulat descomptat normalitzat, en anglés ***Normalized Discounted Cumulative Gain*** (**NDCG**), és una mesura de qualitat de la classificació en un rànquing. El càlcul de NDGG segueix el següent esquema, tenim un SRI que retorna una llista ordenada d'elements a una consulta realitzada, per altra banda, tenim altre SRI, però en aquest cas "ideal", que retorna la llista ordenada ideal d'elements. L'últim pas és comparar les dues llistes ordenades comparant l'ordre seguit, com més paregut a la llista ideal millor.

Per entendre millor la idea darrere d'aquesta mesura, la desglossarem:

Guany Acumulat

El guany acumulat, o *cumulative Gain* (CG) en anglés, és la suma dels guanys associats als elements dins d'una consulta, els guanys són valors de rellevància que s'associa a cada element de la llista retornada. La fórmula:

$$CG = \sum_{i=1}^{n} Guany(i) \tag{2.13}$$

Guany Acumulat Descomptat

Anomenat en anglés com *Discounted cumulative Gain* (DCG), té el mateix concepte que CG, però aplica un pas addicional, el qual és descomptar els guanys del elements depenent de la posició on apareix en la llista ordenada. La fórmula:

$$DCG = \sum_{i=1}^{n} \frac{\text{Guany}(i)}{\log_2(i+1)} \tag{2.14}$$

El principal problema que trobem en aquesta mesura, és el fet que és una suma acumulada, per tant, és possible que llistes, on l'element més important siga el primer i la resta no siguen rellevants, tinguen un valor menor que llistes on els elements rellevants estiguen al final. Per tant, és necessari la normalització de les llistes.

Guany Acumulat Descomptat Ideal

En anglés IDCG (*Ideal Discounted cumulative Gain*), que és la resposta com a llista ordenada "ideal" de la consulta realitzada. És a dir, ordenar les llistes anteriors per mitjà dels guanys i a posteriori calcular el DCG.

Una vegada ja hem vist totes les parts involucrades en el càlcul del NDCG, formulem l'equació:

$$NDCG = \frac{DCG}{IDCG} \tag{2.15}$$

Tot seguit il·lustrarem el càlcul de la mesura amb un exemple. Tenim 3 consultes amb les seues corresponents llistes ordenades de resposta:

- consulta 1 = llista ordenada 1 (a, b, c, d, e).
- consulta 2 = llista ordenada 2 (f, g, h, i, j).
- consulta 3 = llista ordenada 3 (k, l, m, n, o).

Amb els seus respectius guanys (rellevància) per element, on 1 significarà document rellevant i 0 irrellevant:

- llista ordenada 1 = (0, 0, 1, 1, 1)
- llista ordenada 2 = (1, 0, 1, 0, 1)
- llista ordenada 3 = (1, 0, 0, 0, 0)

Ara calculem el CG de cada llista:

- $CG_1 = 0 + 0 + 1 + 1 + 1 = 3$
- $CG_2 = 1 + 0 + 1 + 0 + 1 = 3$
- $CG_3 = 1 + 0 + 0 + 0 + 0 = 1$

Com podem observar, trobem 2 llistes amb un valor igual i altra amb un valor menor. Pel que encara no podem decidir quina és la millor de totes. Hem de tindre en compte l'ordre de la llista, per això, necessitem calcular el DCG:

- $DCG_1 = \frac{0}{\log_2(1+1)} + \frac{0}{\log_2(2+1)} + \frac{1}{\log_2(3+1)} + \frac{1}{\log_2(4+1)} + \frac{1}{\log_2(5+1)} = 1.31752$
- $DCG_2 = \frac{1}{\log_2(1+1)} + \frac{0}{\log_2(2+1)} + \frac{1}{\log_2(3+1)} + \frac{0}{\log_2(4+1)} + \frac{1}{\log_2(5+1)} = 1.88685$
- $DCG_3 = \frac{1}{\log_2(1+1)} + \frac{0}{\log_2(2+1)} + \frac{0}{\log_2(3+1)} + \frac{0}{\log_2(4+1)} + \frac{0}{\log_2(5+1)} = 1$

En aquest cas podem veure com és necessari la normalització. La llista 3 obté un valor menor que la llista 1, la qual no té cap mena de lògica si volem comparar-les, donat que la llista 1 no ha prioritzat els elements més importants, cosa que si ha fet la llista 3. Açò ve donat perquè estem parlant d'una suma acumulativa, per tant, llistes amb més elements sumen un valor major.

Una vegada vist el problema donem pas a calcular l'IDCG de cada llista:

- $IDCG_1 = \frac{1}{\log_2(1+1)} + \frac{1}{\log_2(2+1)} + \frac{1}{\log_2(3+1)} + \frac{0}{\log_2(4+1)} + \frac{0}{\log_2(5+1)} = 2.13093$
- $IDCG_2 = \frac{1}{\log_2(1+1)} + \frac{1}{\log_2(2+1)} + \frac{1}{\log_2(3+1)} + \frac{0}{\log_2(4+1)} + \frac{0}{\log_2(5+1)} = 2.13093$
- $IDCG_3 = \frac{1}{\log_2(1+1)} + \frac{0}{\log_2(2+1)} + \frac{0}{\log_2(3+1)} + \frac{0}{\log_2(4+1)} + \frac{0}{\log_2(5+1)} = 1$

Una vegada hem calculat el DGC i l'IDGC, passem a calcular el NDCG:

- $NDCG_1 = \frac{DCG_1}{IDCG_1} = \frac{1.31752}{2.13093} = 0.61828$
- $NDCG_2 = \frac{DCG_2}{IDCG_2} = \frac{1.88685}{2.13093} = 0.88546$
- $NDCG_3 = \frac{DCG_3}{IDCG_3} = \frac{1}{1} = 1$

En aquest exemple, podem concloure que la millor llista ha sigut la 3. Si utilitzem la lògica, observem que té sentit, ja que els elements més importants estan en les primeres posicions.

2.4.2.10. Rang Recíproc Mig

El rang recíproc mig, o *Mean reciprocal rank* (MRR) en anglés, és una mesura utilitzada per avaluar el rànquing retornat com resposta a una consulta. Aquesta mesura és la mitjana dels rangs recíprocs de diferents consultes. El rang recíproc és la inversa del rang on apareix una primera resposta rellevant i es calcula utilitzant la següent fórmula:

$$RR = \frac{1}{\text{rang}} \tag{2.16}$$

On el rang representa en quina posició de la llista està la primera resposta rellevant. Exemple: tenim un rànquing on el primer element rellevant està en al posició tercera, per tant, el RR $= 1/3$.

El rang recíproc mig es calcula per mitjà de la següent fórmula:

$$RMM = \frac{1}{Q} \cdot \sum_{i=1}^{Q} \frac{1}{rang_i} \tag{2.17}$$

On Q és el nombre total de consultes realitzades. A continuació mostrarem un exemple:

Consulta	Rànquing	Resposta correcta	Rang	Rang Recíproc
Gos	1: humà, 2: animal, **3: gos**	gos	3	1/3
taula	1: cadira, **2: taula**, 3: estufa	taula	2	1/2
pilota	**1: pilota** 2: ordinador 3: raqueta	pilota	1	1

Taula 2.8: Càlcul de Rang recíproc per a 3 consultes

La Taula 2.8 mostre 3 consultes realitzades i el càlcul del seu rang recíproc. Una vegada calculats el rang recíproc de les consultes, podem passar a calcular el rang recíproc mig. $RRM = \frac{(\frac{1}{3}+\frac{1}{2}+1)}{3} = \frac{11}{18} = 0.61$

2.4.2.11. Mesures relacionades amb l'usuari

Les mesures orientades a usuaris són la raó de l'existència del sistema. L'efectivitat d'un sistema és una mesura que relaciona la satisfacció de l'usuari amb l'eixida que proporciona el sistema. Aquesta mesura és molt complicada de mesurar, ja que, la satisfacció de l'usuari pot variar d'un moment a altra i és molt subjectiva.

Baeza-Yates defineix les següents mesures:

- **Ràtio de cobertura**: Proporció de documents rellevants coneguts per l'usuari que són recuperats.
- **Ràtio de novetat**: Proporció de documents rellevants recuperats no coneguts per l'usuari.
- **Exhaustivitat relativa**: Documents rellevants recuperats que han sigut examinats per l'usuari partit el nombre de documents que l'usuari vol examinar.

El ràtio de cobertura el considerem alt en superar un 30%, aquesta mesura dona confiança a l'usuari, perquè pot comprovar com una part del documents que han sigut recuperats són rellevants. Cal parar atenció per si puja molt, ja que l'usuari es cansarà. Per això, és important tindre un ràtio de novetat alt, així l'usuari tindrà nous documents rellevants a llegir.

Una altra mesura podria ser l'esforç exhaustiu, quants documents necessita examinar l'usuari per trobar aquells que ell desitja. És a dir, si tenim 20 documents retornats, 5 d'ells rellevants, l'usuari vol llegir també 5 rellevants, si aquest documents estan al principi, tindrem un esforç proper a l'1, és a dir, l'usuari ha fet poc esforç, però, si al contrari, el documents estan al final, l'usuari ha hagut de fer un esforç molt alt.

2.5 Eines per al desenvolupament de SRI

En aquest apartat anem a fer un anàlisi de les eines més utilitzades a l'hora de desenvolupar SRI. Farem un repàs a llenguatges de programació, llibreries, Frameworks, etc.

2.5.1. Llenguatge de Programació

Per poder desenvolupar un SRI en necessari utilitzar un llenguatge de programació que s'ajuste a les necessitats de l'àmbit, a continuació presentarem el llenguatges més utilitzats en àrea del PLN.

2.5.1.1. Python

Considerat com uns dels millors llenguatges de programació actuals per la seua capacitat d'adaptació a qualsevol àmbit. Escriptura natural i eficient que facilita l'aprenentatge i la velocitat d'ús. Un llenguatge multiplataforma, usat tant en Windows com en Linux o macOS. Python és adaptable amb altres llenguatges de programació, com Java i C++. Una de les característiques més important és que compta amb una gran biblioteca estàndard de codi obert, que facilita l'escriptura de codi. A més, compta amb una gran comunitat de programadors que desenvolupen noves aplicacions o ajuden a resoldre problemes en fòrums. Python és un dels llenguatges amb més llibreries i paquets, molts d'aquests paquets estan enfocats al *Machine Learning* i a la creació de SRI.

2.5.1.2. Java

Java és un dels llenguatges de programació més utilitzat en tot el món. Java és un llenguatge multiplataforma i orientat a objectes, es considera un del llenguatges més segurs i confiables del mercat. Respecte a l'àmbit del PLN, Java és un llenguatge que s'utilitza molt degut a les nombroses llibreries que ajuden al desenvolupament de sistemes PLN, encara això, molts desenvolupadors prefereixen Python. Tanmateix, Java continua sent una de les opcions més fiables, a causa de la seua estabilitat i velocitat.

2.5.1.3. R

Llenguatge de programació específic per treballar en l'anàlisi de dades i l'aprenentatge automàtic. En aquest àmbits és on millor es pot extraure les seues capacitats. És un llenguatge que compta amb molts paquets preparats per al seu ús dins del PLN; tanmateix, és un llenguatge ràpid i preparat per a compartir informació fàcilment.

En aquest projecte anem a fer ús de **Python** degut a la seua facilitat d'ús, les innumerables llibreries per desenvolupar sistemes PLN i per la meua experiència passada.

2.5.2. Llibreries

En aquest subapartat sols explicarem paquets especialitzats en Python, encara que existeixen nombrosos paquets per altres llenguatges igual d'importants.

Figura 2.27: llibreries PLN per a Python

2.5.2.1. NLTK

Natual Language Toolkit marc de desenvolupament de programes per administrar i analitzar dades de llenguatge humà. NLTK ofereix envoltoris per a potents llibreries de PLN, més de 50 corpus diferents per als teus projectes i molts d'altres recursos per als teus programes. Entre les principals característiques trobem un conjunt de llibreries de processament de text per a categorització, tokenització, etiquetat, etc.

Considerada una de les llibreries més completes i útils per al desenvolupament de sistemes PLN, encara que la seua eficiència és reduïda, és una llibreria lenta.

2.5.2.2. Stanford Core NLP

El marc CoreNLP és una de les llibreries més apreciades pels programadors en l'àmbit del PLN. Va ser desenvolupada en Java, però compta en una API per poder ser usada en Python. Aquesta llibreria compta amb totes les ferramentes necessàries per al desenvolupament de SRI, com tokenització, "part of speech tagger", analitzador de dependències, anàlisi de sentiments, etc. A més, admet sis idiomes per poder utilitzar els seus elements, i és considerada una ferramenta flexible i eficient.

2.5.2.3. SpaCy

Ferramenta fonamental a l'hora de desenvolupar projecte en l'àmbit PLN. És l'evolució de NLTK que incorpora vectors de paraula, word embeddings, i models preentrenats. Aquesta llibreria suporta més de 49 idiomes, i el seu punt fort està en la tokenització. Considerada com la ferramenta més útil en el mercat actual, a causa de la seua capacitat d'adaptació i compatibilitat en qualsevol projecte.

Hi ha d'altres llibreries molt importants com **PyTorch** o **Tensorflow**, que s'utilitzen per a desenvolupament de xarxes neuronals a més baix nivell.

2.5.3. FrameWorks

En aquest apartat en centrarem en els dos frameworks que utilitzarem per al desenvolupament d'aquest projecte.

Figura 2.28: Frameworks utilitzats al projecte

2.5.3.1. Sentence Transformer

ST és un *framework* [39] que treballa amb *embeddings* d'oracions, paràgrafs i imatges. Aquest *framework* treballa amb l'estat de l'art relacionat amb els *embeddings*, tant amb BERT, XLNet, RoBERTa i MPNet. La compatibilitat amb més de 100 idiomes és una de les característiques principals, és a dir, et permet calcular els *embeddings* d'oracions per

un gran nombre d'idiomes. Aquestes representacions vectorials es poden usar en moltes tasques diferents.

Aquest framework és molt útil tant per a la comparació de similitud com per a parafrasejar oracions. És un *framework* que et permet desenvolupar aplicacions PLN amb molta facilitat, ja que tens al teu abast moltes ferramentes diferents. No sols això, el *framework* té permet utilitzar models guardats en Huggign Face.

En conclusió, ST és un *framework* increïble per desenvolupar SRI, donat que et proporciona totes les ferramentes necessàries.

2.5.3.2. Haystack

Haystack és un *framework* de còdi obert per crear sistemes de cerca que funcionen d'una forma intel·ligent en grans col·leccions de documents. Haystack està plantejat per fer de pont entre la investigació i la indústria, donat que molts dels últims avanços en el món del PLN no estan del tot implementats. Per això Haystack proporciona:

- **PLN per a la cerca**: Et dona la capacitat d'elegir entre diferents models i tècniques, per realitzar la recuperació, la resposta de preguntes, classificació de textos.
- **Últims models**: Utilització dels últims models que podem trobar al mercat (BERT, RoBERTa, MiniLM, etc.), i capacitat de canviar de models en un mateix projecte.
- **Bases de dades flexibles**: Gran varietat de diferents tipus de bases de dades a utilitzar com Elasticsearch, Milvus, FAISS, i més.
- **Escalabilitat**: Capacitat d'escalar el sistema per poder gestionar milions de documents al mateix temps.
- **Adaptació del domini**: Dotació de ferramentes per poder anotar exemples, recopilar comentaris dels usuaris, avaluar components i més.

En resum, Haystack és un *framework* especialitzat a crear recuperadors d'informació, on gràcies a les seues *pipelines* i bases de dades podem manejar milions de documents, d'una manera ràpida i eficient. No sols això, sinó que permet l'ús dels últims models creats i també dels propis, sempre mantenint-se actualitzada. Per tant, és la ferramenta idònia per desenvolupar un SRI.

2.6 Crítica a l'Estat de l'Art

Una de les carències principals de l'actualitat, és la manca de corpus utilitzables per desenvolupar projecte en idiomes minoritaris, com pot ser el català, a l'àmbit del PLN. Molts dels corpus que t'ofereixen les ferramentes actuals i que utilitzen els grans projectes, estan en anglés o en castellà. Com a conseqüència, la gran majoria de models estan creats per ser utilitzat en aquest idiomes i on l'adaptació d'aquestes a idiomes minoritaris és costosa. Comportant que els models per idiomes majoritaris solen tindre un rendiment major que els models per llengües minoritàries, donat que hi han moltíssimes més dades d'entrenament.

Com hem comentat l'adaptació és costosa, donat que, com moltes de les ferramentes estan per idiomes majoritaris, moltes de les dades que s'utilitzen per a l'entrenament del models també. Produint un cost de traducció manual (poques dades) o automàtica (moltes dades) i de neteja dels errors, cosa que produeix soroll a les dades.

Com a conseqüència, es crea una barrera d'entrada per a nous desenvolupadors alta, la qual impedeix un creixement còmode i pot desencoratjar a moltes persones de continuar amb el seu projecte.

En conclusió, la crítica principal va encaminada als pocs recursos actuals que pateixen les llengües minoritàries a l'hora de poder desenvolupar projectes, donat que no hi ha grans recopilacions de documents ni models d'alt rendiment.

2.7 Proposta

Una volta analitzades les carències del mercat actual, hem proposat el següent projecte, crear un sistema de recuperació d'informació en català. On les tecnologies utilitzades, encara que no puga competir amb Google, siguen les més punteres, és a dir, utilitzar *frameworks* i models que ens permeten tindre un sistema amb unes prestacions i eficiència altes.

Aquest projecte consistirà, en una primera instància, en la comparació de diferents models punters, com STSB, XLNet i MPNet, en la feina de recuperar informació utilitzant diferents mesures, per elegir el millor model que s'ajusta als nostres requisits, amb un entrenament posterior d'aquell model amb millors prestacions. Posteriorment, en una segona instància, la creació de l'estructura d'un SRI utilitzant *Haystack*, on una de les principals característiques serà la seua escalabilitat a milions de dades. Com a resultat, obtindrem un SRI que siga capaç de tractar les nostres dades, recopilacions de diaris o transcripcions de vídeos, amb unes prestacions altes, comparables als de les tecnologies més actuals del mercat.

En conclusió, presentem un projecte de creació d'un SRI que utilitza les últimes tecnologies i els models més avançats per omplir la manca de projectes d'aquest tipus en l'àmbit del PLN en català.

CAPÍTOL 3
Anàlisi del problema

Una vegada finalitzat l'estudi de l'estat de l'art, on hem vist la teoria que suporta aquest projecte i les tecnologies més punteres en el nostre àmbit. Fem pas, al anàlisi del problema on exposarem el problema principal a resoldre i quines oportunitats d'innovació se'ns presenten a partir d'aquest problema.

3.1 El Problema

El principal problema que trobem, explicat en major profunditat a l'apartat 2.6, és la manca de ferramentes i models[1] desenvolupats per a llengües minoritaris. No obstant això, com enginyers que som tot problema pot convertir-se en un projecte d'innovació, i és el que intentem aconseguir des del grup d'investigació. El nostre projecte intenta agafar aquesta manca de SRI en català i crear-ne un per fer-lo de lliure accés a la comunitat PLN i alliberar el codi.

Per poder crear un SRI que puga donar prestacions semblants als SRI punters al mercat actual, és necessari la fer un estudi dels requisits que ha de complir. Aquest requisits són els següents:

- SRI escalable amb una gran capacitat per indexar un gran nombre de documents.
- Ús de tècniques eficients i tecnologies punteres.
- Avaluació dels models per tal d'assegurar-ne la qualitat.
- Comprovació dels resultats finals del projecte.
- Escriptura estructurada i neta del codi.
- Escriptura de la documentació del projecte.

Una vegada analitzat els requisits que hem de complir per crear un projecte d'acord amb les expectatives, hem enfocat els nostres esforços en els següents apartats: eficiència algorítmica i escalabilitat del sistema, i l'anàlisis de riscos.

3.1.1. Anàlisi d'eficiència algorítmica i escalabilitat

Un dels principals reptes que hi ha en l'àmbit de la informàtica, i més en el PLN, és la necessitat de l'eficiència. Necessitem que els processos cada volta siguen més ràpids

[1] Són les xarxes neuronals utilitzades per extraure els words embeddings

i necessiten menys recursos, per fer front a l'alta demanda de tasques requerides pels usuaris i per l'enorme quantitat de dades que es tracten hui en dia.

Enfocant aquest problema als SRI, trobem que estem treballant en quantitats de dades monstruoses, no sols per a la tasca de recuperació sinó també per a l'entrenament dels models que s'utilitzen. Aleshores, si s'utilitzen algoritmes ineficients que alenteixen l'execució i consumeixen molts recursos, obtindrem sistemes que no satisfaran els requisits mínims, obtindrem sistemes que crearan insatisfacció a l'usuari per no obtindre respostes ràpidament o aquestes respostes no seran correctes. I si les tècniques utilitzades per a l'entrenament de models no són eficients el temps requerit per finalitzar l'entrenament serà inassumible, i pot ser, que el model obtingut no tinga les mateixes capacitats.

Com podem observar, tot va relacionat amb el nombre de dades que tractarem, perquè si parlem de poques dades no es noten tant les diferències plausibles d'ineficiència, però a l'hora de tractar amb moltes dades és on la ineficiència es deixa notar. En aquest projecte estem parlant de tractar al voltant d'1 milió de documents per castellà i entre 600.000 - 800.000 documents en català, és a dir, a quantitat de dades considerable, el corpus utilitzat en aquest projecte es detalla en la Secció 4.2.2.1. Per tant, tenim una necessitat d'algoritmes eficients, per poder reduir el temps d'execució, d'indexació, d'entrenament, etc. al mínim i que la despesa de recursos, emfatitzant en els de memòria, siguen mínims també.

Com a conseqüència, aquest projecte té la necessitat d'utilitzar aquelles ferramentes i llibreries que ens proporcionen algoritmes eficients i avançats per crear un SRI concorde al requisits, que siga eficient i que aquesta eficiència escale amb el nombre de dades.

3.1.2. Anàlisi de riscos

L'objectiu final d'un SRI és donar serveis de recuperació d'informació, aquest serveis poden comportar riscos relacionats amb l'usuari. El riscos són els següents:

Risc d'acceptació

Aquest projecte té la finalitat de crear un sistema on els usuaris puguen cercar informació sobre diferents notícies al llarg del temps. Per tant, és un projecte orientat a l'usuari on la seua acceptació és important. El risc té les següents característiques:

- **Tipus**: Risc d'acceptació de l'usuari
- **Impacte esperat**: Si tenim una mala acceptació per parts de l'usuari, podem deduir, que el nostre sistema no ha complit amb els requisits que esperem, pel fet que o és lent, no recupera la informació en un temps acceptable, o la informació recuperada no és rellevant.
- **Mesures**: Per poder tindre una millor acceptació, i en conseqüència, complir els requisits, utilitzarem les últimes tecnologies al nostre abast per crear un sistema eficient. Una altra mesura és fer diferents proves d'estil per tornar una lectura clara dels resultats a l'usuari.

Risc de satisfacció

Paregut al risc anterior, en ser un projecte orientat a ser utilitzat per usuaris, si els resultats obtinguts per la cerca no són satisfactoris, és a dir, no tenen relació amb la consulta, existeix el risc que els usuaris no tornen a utilitzar el sistema. Per tant, és necessari tindre

un sistema que complisca amb els requisits de satisfacció de l'usuari. Les característiques del risc són les següents:

- **Tipus**: Risc de satisfacció
- **Impacte esperat**: Com hem comentat abans, si el grau de satisfacció en l'usuari és baix pot ser que no tornen a utilitzar aquest sistema. Açò pot ser causat per diferents elements, com l'ús d'un model antic o poc potent, que no té la capacitat de calcular adequadament a la similitud entre frases, o, també, pel fet de retornar els documents en una estructura inadequada.
- **Mesures**: Per poder fer front a aquest risc, hem conclòs, que la millor solució és fer avaluacions i anàlisis dels models a utilitzar, per trobar aquell que millor s'ajusta a aquest projecte. Obtenint un model potent que siga capaç de retornar els documents més semblants a la consulta realitzada.

Risc d'Integració

Comentat amb anterioritat, aquest projecte formara part d'un projecte més gran. Alesho-res, existeix el risc d'una mala integració amb la resta, les seues característiques són:

- **Tipus**: Risc d'integració.
- **Impacte Esperat**: El projecte ha de desenvolupar-se seguint les directrius establer-tes en el projecte majoritari, per a la seua posterior integració, si no pot causar pro-blemes estructurals i endarrerir el desenvolupament.
- **Mesures**: Seguir les directrius establertes i anar contrastant els avanços amb el di-rector del projecte, per poder solucionar problemes futurs.

3.2 Identificació i anàlisi de les possibles solucions

Una vegada analitzats els principals requisits i riscos del projecte a desenvolupar, podem concloure que no hi ha una única solució possible a aplicar. A continuació mostrarem una sèrie de possibles solucions amb els seus avantatges i desavantatges:

3.2.1. Solució amb diferents implementacions de SRI i avaluació manual

Aquesta solució consisteix en la creació de diferents sistemes de recuperació d'informa-ció, on en cada sistema utilitzarem un model de representació vectorial diferent. Cada SRI serà avaluat seguint un criteris d'avaluació manual, el qual consistirà a realitzar diver-ses cerques i analitzar els resultats comparant-los amb els dels altres SRI. A continuació mostrarem els avantatges i desavantatges:

- **Avantatges**:
  - Tres implementacions diferents d'un SRI.
  - Comparació entre els diferents models utilitzant el nostre corpus [2].

[2] La col·lecció de documents

- **Desavantatges**:
  - Lenta implementació, més feina a fer.
  - Imprecisió de treure resultats concloents, no hi ha una valoració objectiva.
  - Molta memòria utilitzada.
  - No hi ha una valoració dels models a utilitzar.

En conclusió, en aquest projecte volen focalitzar-nos en l'avaluació objectiva dels models (xarxes neuronals) a utilitzar, aconseguint una taula comparativa entre ells de diferents mesures d'avaluació. Per tant, aquesta solució no ens permet fer una comparació objectiva, hauríem de ser nosaltres qui a posteriori valorarem els resultats manualment.

3.2.2. Solució amb anàlisi objectiva de models

Solució posterior on utilitzarem la llibreria Beir 4.3.3.5, per fer l'avaluació dels diferents models disponibles. En aquesta avaluació utilitzarem diferents mesures per veure quin model és el que millor s'ajusta a les nostres dades. Al final, amb el millor model crearem el sistema de recuperació d'informació. Aquesta solució té les següents característiques:

- **Avantatges**:
  - Comparació i anàlisi dels diferents models.
  - Una única implementació d'un SRI.
  - Avaluació objectiva dels resultats.
  - Compliment dels objectius.
- **Desavantatges**:
  - No hi ha un treball d'investigació profund, sols un estudi dels models.
  - Poc ambiciós.
  - Temps consumit fent la comparació.

Aquesta solució no presenta cap desavantatge en termes de compliment dels objectius, però, a mi personalment, m'agradaria aprofundir en al creació i entrenament d'un model. Donat que la comparació de models no satisfà la meua voluntat d'investigar.

3.2.3. Solució amb un model preentrenat

Aquesta solució consisteix a agafar la solució anterior i afegir-li l'entrenament d'un model preentrenat, és a dir, fer-lo més específic per a la feina requerida. Per tant, en aquesta solució tindrem 3 etapes, primera la de la comparació de models per triar el millor, la segona l'entrenament d'aquest model i la tercera i última la creació del SRI. A continuació les característiques:

- **Avantatges**:
  - Comparació i anàlisi dels diferents models.
  - Una única implementació d'un SRI.
  - Avaluació objectiva dels resultats.
  - Compliment dels objectius.

  - Entrenament de model, per satisfer interessos investigadors

- **Desavantatges**:
  - Temps consumit amb comparació i entrenament molt alt.
  - Nul·la avaluació del model entrenat.

En conclusió, és una solució que arreplega prou factors positius de l'anterior solució proposada, i a més, podré satisfer les meues ambicions investigadores en entrenar un model i aprofundir en aquest món. Encara que és una solució prou més lenta i costosa en tots els aspectes, donat que hem de fer l'anàlisi de models, entrenar el model seleccionat i crear l'estructura del SRI.

3.3 Solució proposada

Una vegada hem aclarit i estudiat totes les possibles soluciones, fent un estudi dels seus avantatges i desavantatges, l'equip d'investigadors hem acordat que la millor solució per complir els objectius i les ambicions del projecte és la tercera opció.

3.3.1. Descripció de la solució proposada

Aquesta solució consisteix en la implementació d'un SRI utilitzant un model de representacions vectorials denses que millor s'ajusta a les nostres dades. Per poder assolir aquesta finalitat, hem dividit la solució en 4 etapes principals:

1. **Estudi de l'art i preparació de les dades**: Primer de tot cal investigar i estudiar tota la teoria relacionada amb el nostre projecte, per poder elegir no sols les millors ferramentes a utilitzar sinó també quin models volem comparar. En aquesta mateixa fase, després d'elegir les ferramentes a utilitzar i els models, hem de fer compatibles les dades amb els nostres requisits, en el nostre cas, cal traduir un corpus d'entrenament en castellà al català.

2. **Anàlisi comparativa entre models**: Una vegada tenim les dades preparades i sabem quins models volem comparar, amb l'ajuda de llibreries especialitzades en l'avaluació de models, crearem un *script* [3] per mesurar cada model individualment i, posteriorment, crearem una taula comparativa amb els resultats, per elegir el millor model.

3. **Entrenament del model**: Fase que consisteix a entrenar el model amb unes dades més properes al requisits del projecte. Aquest entrenament es fa per mitjà de l'ús d'un *script* especialitzat en l'entrenament de models.

4. **Implementació estructura SRI**: Una vegada ja hem entrenat el model a utilitzar, podem passar a crear l'estructura del nostre SRI per mitjà de llibreries i eines triades amb anterioritat. Finalment, posarem en marxa el nostre model i farem una sèrie de consulta per veure que tot funciona correctament.

Els models a comparar seran aquells que estan basats en els últims avanços, com per exemple, BERT, XLNet o MPNet, ja que, triar models anteriors a aquest suposarà un despesa de temps i recursos, i està fora de l'objectiu del projecte.

[3] Arxius que contenen el codi font de Python, destinats a iniciar-se des de la línia d'ordres

S'ha de tindre en compte que l'avaluació final del SRI, ha de ser per mitjanament subjectiva, ja que nosaltres no comptem amb un corpus etiquetat per fer avaluacions. Per tant, en un primer moment utilitzarem un corpus etiquetat per avaluar el model, i confiarem que aquests resultats es transmeten al nostre propi model, no obstant sí que durem a terme una xicoteta avaluació subjectiva final del SRI complet..

3.3.2. Model Conceptual

Un model conceptual és una representació d'un sistema, fet de la composició de conceptes que s'utilitzen per comprendre millor el funcionament del sistema a mostrar, on s'inclouen les relacions entre les parts més importants. El model conceptual d'aquest projecte cal dividir-ho en fases, ja que, conceptualment cada fase és diferent de l'anterior i necessitem representar-ho de forma diferent. Les fases són les següents:

- Primera Fase: Preprocessament del text i comparació de rendiments dels models.

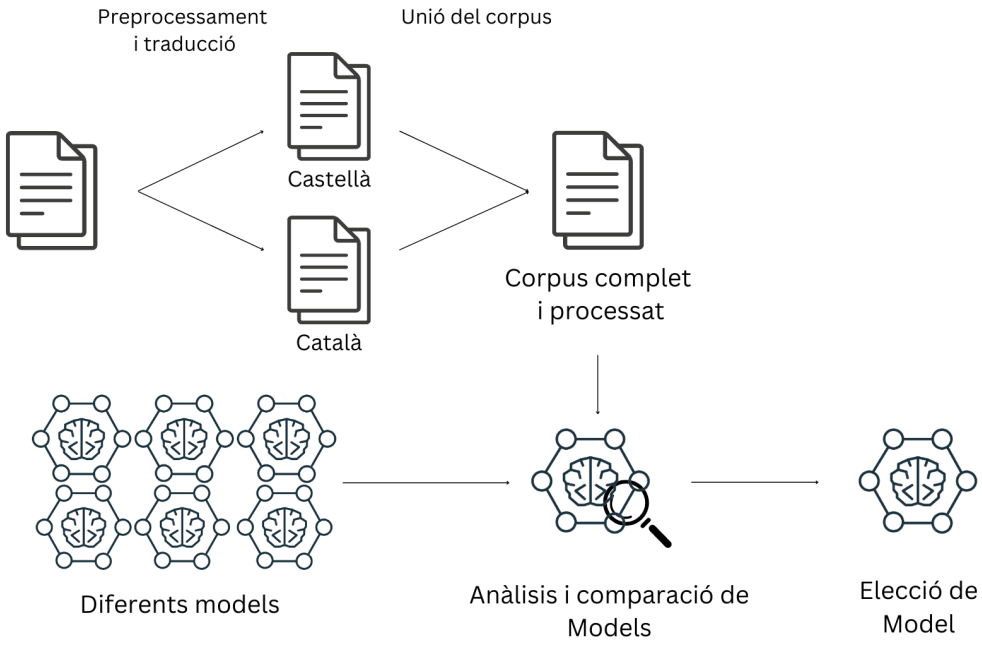


Figura 3.1: Model Conceptual de la Primera Fase

El Model conceptual explica les característiques principals d'aquesta fase, com primer tenim un etapa de preprocessament del corpus i traducció de castellà a català. Finalment, una etapa de comparació de models per trobar aquell amb millor rendiment.

- Segona Fase: Estructura d'un SRI

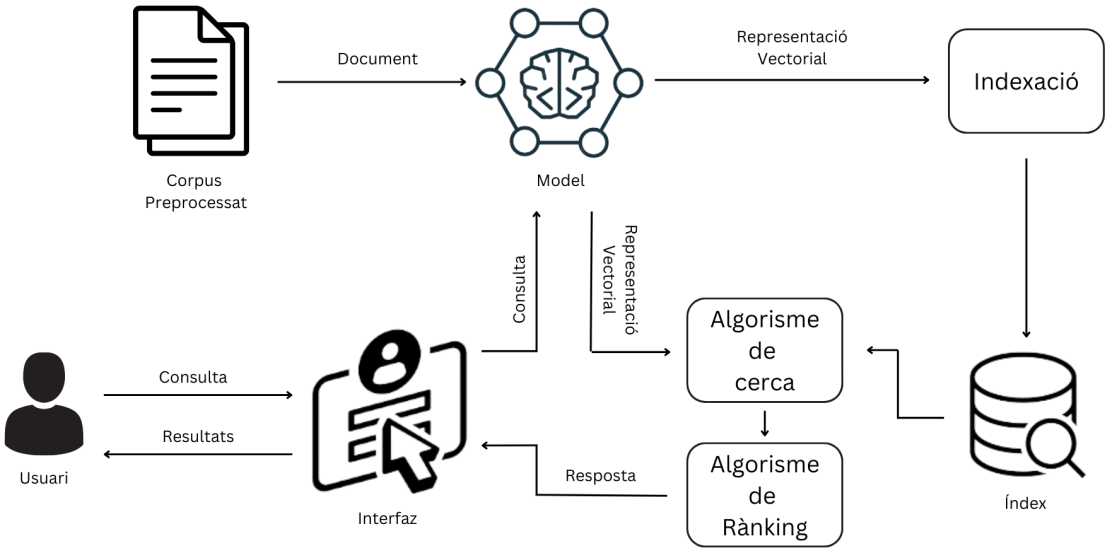


Figura 3.2: Model Conceptual de la Segona Fase

Com es pot observar en el model conceptual, tenim una representació visual del procediment que fa un SRI: Primer la creació de l'índex on cada document ha sigut representat per mitjà de diferents representacions vectorials de les seues frases. Finalment, la fase d'inferència: on arribada una consulta, seguidament la transforma en un vector per fer la comparació amb les representacions vectorials de l'índex. Posteriorment, extreu els documents amb informació més rellevant i els retorna en un rànquing.

3.3.3. Pla de Treball

El pla de treball consisteix a planificar i estimar l'esforç requerit en desenvolupar el projecte. En particular, aquesta planificació s'ha fet abans de començar el desenvolupament, donat que volem mostrar al públic quines variacions ha anat sofrint al llarg del temps. L'important d'aquest apartat és l'avaluació crítica de la planificació.

3.3.3.1. Planificació per Etapes

Aquest subapartat mostra la planificació del projecte. Aquesta planificació la podem dividir en diferents etapes:

- **Primera etapa**: Aquesta etapa consisteix en l'estudi de l'estat de l'art. És a dir, un estudi teòric dels SRI i una avaluació del projectes més punters. Desglossant aquesta etapa tindrem les següents feines: Primer una lectura dels articles científics més important i moderns, continuem en un estudi del mercat actual, on estudiarem diferents SRI creats per les empreses més punteres, per comprendre quines tecnologies utilitzen. Finalment, tindrem una reunió amb els tutors de TFM per concretar quines tecnologies ens beneficien i proposar una solució.
- **Segona etapa**: Una vegada hem estudiat tot el marc teòric i la competència, passem a fer un estudi de les ferramentes a utilitzar. Aquesta elecció consisteix a seleccionar

tant les llibreries i *frameworks* a usar com els models PLN a utilitzar. En aquesta etapa, cal tindre especial èmfasi en cercar i entendre bé les diferències entre els models i les seues prestacions.

- **Tercera etapa**: Posteriorment, a la selecció de models hem de fer l'anàlisi comparativa. Aquesta anàlisi consisteix a testejar els diferents models amb un corpus etiquetat, obtenint com a resultat l'avaluació de diferents mesures per a cada model. Finalment, obtinguts els resultats elegirem quin model serà l'utilitzat per a entrenar-lo.

- **Quarta etapa**: L'objectiu d'aquesta etapa és entrenar el model seleccionat amb dades més específiques de la feina a fer. Per aquest entrenament s'utilitzarà un conjunt de dades etiquetades tant en català com en castellà, que representen diferents consultes amb un document a recuperar. Així, el model aprendrà a saber quin és el document més rellevant per cada consulta.

- **Cinquena etapa**: En aquesta etapa ja tenim tots els elements necessaris (Model, *framework*, llibreries, etc.) per crear el SRI. Per tant, l'objectiu és crear l'estructura del SRI i fer diferents proves. Una vegada ja hem comprovat el seu correcte funcionament, el projecte ha finalitzat amb èxit.

3.3.3.2. Planificació Inicial-Real

L'objectiu d'aquest subapartat és mostrar les diferències entre la planificació inicial efectuada, on es pressuposaven les setmanes de treball que costaria cada etapa, amb el treball Real.

La planificació inicial va ser la següent:

- **Primera etapa**: 2 setmanes de Treball.
- **Segona etapa**: 1 setmana de Treball.
- **Tercera etapa**: 3 setmanes de Treball.
- **Quarta etapa**: 4 setmanes de Treball.
- **Cinquena etapa**: 3 setmanes de Treball.
- **Total**: 13 setmanes de Treball

A causa de l'experiència anterior treballant amb models que necessitem moltes dades, sabem que les etapes més costoses són aquelles relacionades amb l'estudi i entrenament dels models. Una altra dificultat que podem trobar és l'ús d'una nova ferramenta per desenvolupar SRI, donat que la meua inexperiència pot endarrerir tot el desenvolupament.

Una vegada acabat el projecte, la planificació real del treball va ser la següent:

- **Primera etapa**: 2 setmanes de Treball

  En aquesta etapa sí que en vàrem ajustar bé al temps pressupostat, donat que com esperàvem en un primer moment era fer un estudi teòric de les noves tecnologies i models disponibles.

- **Segona etapa**: 2 setmanes de Treball

  En aquesta etapa trobem la primera variació de temps de treball. Perquè l'estudi dels diferents models de representacions vectorials ha sigut més dur de l'esperat, ja que la teoria on se sustenten és cada vegada més complicada. No sols això, sinó que la cerca de models exemples o preentrenats que podem utilitzar ha resultat ser més costosa. Per altra banda, aquesta etapa l'hem realitzat en paral·lel a l'etapa anterior, donat que totes dues són cerques d'informació molt relacionades.

- **Tercera etapa**: 4 setmanes de Treball

  Com ja esperàvem, el preprocessament dels documents i l'anàlisis de models ha consumit prou de temps. No sols hem hagut d'esperar 2-3 setmanes a què el corpus fora traduït al català i netejat, sinó que l'anàlisi de models era prou costos, estem parlant de 10 hores d'execució en GPU per aconseguir els resultats. Tota aquesta demora ve produïda per utilitzar un corpus amb al voltant de 9 milions de documents.

- **Quarta etapa**: 5 setmanes de Treball

  Molt pareguda a l'etapa anterior. L'ús d'un corpus gran, ha produït que l'entrenament del model a utilitzar costarà vora a 3-4 setmanes de treball. Tanmateix, amb les diferents proves realitzades per veure que el funcionament és correcte, ha generat un endarreriment del pla de Treball.

- **Cinquena etapa**: 2-3 setmanes de Treball

  Aquesta etapa ha sigut més ràpida que el pressupostat, gràcies al fet que les ferramentes elegides han facilitat molt la construcció del SRI. A més a més, l'ús d'indexador d'última generació permet un procés d'indexació ràpid. La part més laboriosa és la comprovació subjectiva dels resultats obtinguts pel SRI, però no consumeix un temps excessiu.

- **Total**: 15 setmanes de Treball.

La Figura 3.3 mostra la diferència entre la planificació inicial i la real. On el taronja és la planificació inicial i la seua evolució en les etapes, i el blau mostra com la planificació real ha evolucionat amb relació d'aquesta planificació inicial.

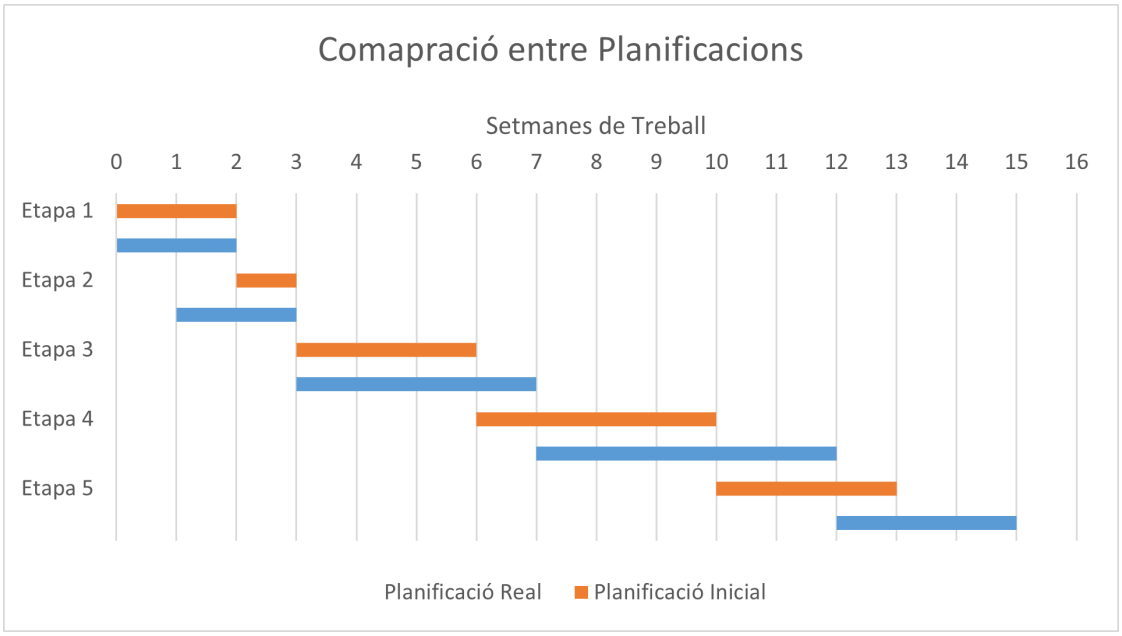


Figura 3.3: Comparació de planificacions

Com s'observa en la figura 3.3, la planificació inicial està prou acorde al temps utilitzat per desenvolupar aquest projecte. Per tant, és una planificació bona del temps de treball, on hem tingut en compte tots els possibles colls de botella, els costosos processos d'entrenament i avaluació de models, etc.

CAPÍTOL 4
Disseny de la solució

Una vegada hem vist la solució proposada i quins poden ser els principals problemes, passarem a detallar quin serà el disseny a utilitzar i quines tecnologies utilitzarem.

4.1 Arquitectura del Sistema

Com hem vist en l'apartat del Model conceptual 3.3.2, tenim dues arquitectures principals en aquest projecte:

- **Comparadors de models**: Arquitectura centrada en l'anàlisi i comparació de models, per trobar aquell que utilitzarem en el nostre projecte.
- **Arquitectura SRI**: La segona arquitectura que crearem estarà basada en un arquitectura d'un SRI. És a dir, per mitjà de diferents ferramentes crearem un SRI complet que siga eficient i eficaç.

4.1.1. Creador de Model

Aquesta arquitectura tindrà com a objectiu principal suportar un sistema que siga capaç de trobar el millor model que s'ajuste a les prestacions que busquem. L'arquitectura està composta per diferents elements:

- **Col·lecció de documents (Corpus)**: El corpus és un conjunt de documents o textos tancat que, generalment, s'utilitzen per a la investigació científica. Aquest corpus s'utilitzarà per a l'avaluació dels models i el seu posterior entrenament.
- **Preprocessador del Corpus**: Aquest element s'encarregarà de preprocessar i netejar el corpus en un primer moment. Posteriorment, la seua feina més important és la de traduir el corpus de castellà a català, ja que, volem un model que siga capaç de treballar en els dos idiomes.
- **Anàlisis i comparació de Models**: Sistema centrat en l'avaluació dels diferents models seleccionats i la tria posterior. Aquesta avaluació utilitzarà les mesures vistes en l'apartat 2.4.2, les quals ens mostraran els diferents valors que han obtingut els models, amb els quals podem fer una decisió final de tria.
- **Entrenament del model**: Element que per mitjà del corpus sencer, entrenarà al model triat per millorar-lo en la tasca de recuperació d'informació.

El funcionament del sistema està representat en el model conceptual en la Figura 3.1. Primer lloc el preprocessador del Corpus netejarà i traduirà el corpus en castellà per crear un en català, posteriorment analitzarem els diferents models més actuals disponibles amb el corpus creat (una part castellà altra part català), per analitzar el seu rendiment i poder realitzar una comparació entre ells. Amb la tria ja realitzada del model amb millors prestacions passarem al seu entrenament, aquesta volta utilitzarem tot el corpus sencer.

4.1.2. Arquitectura SRI

Per al SRI, utilitzarem una arquitectura de 5 components:

- **Col·lecció de documents (Corpus)**: Explicat en el comparadors de models, és un conjunt de documents d'on s'extrauran les respostes a les consultes realitzades per l'usuari.
- **Indexador**: Component encarregat de processar el corpus i indexar-ho. És a dir, aquest component representa les entrades del corpus en un índex, per a la seua posterior recuperació. Aquest component s'encarrega de crear les representacions vectorials del corpus i guardar-les en un índex.
- **Índex**: Representacions dels documents guardades per a la seua posterior recuperació. En un apartat posterior explicarem la tria de l'índex a utilitzar.
- **Interfície**: Aquest component es connecta directament amb l'usuari per rebre les seues consultes i les passa a l'algoritme de cerca i rànquing. Posteriorment, retorna els resultats a l'usuari.
- **Algoritme de cerca i rànquing**: Encarregat de transformar la consulta, i amb la representació trobar aquells documents més pareguts per a la seua posterior ordenació per rellevància. Crea una llista documents ordenada per rellevància o que millor contesta la consulta.

L'arquitectura del sistema està representada en el model conceptual de la Secció 3.2. Sense entrar molt en detalls, la funcionalitat serà la següent: primer es crearà un índex de representacions vectorials a partir de tots el documents que compre el corpus. Posteriorment, l'usuari connectarà amb la interfície per realitzar la seua consulta, la qual es passarà a l'algoritme de cerca i rànquing, on es transformarà en una representació vectorial. Aquesta representació s'utilitzarà per executar una cerca en l'índex de les representacions amb una similitud gran, on es realitzarà una ordenació per rellevància d'aquestes representacions cercades. Finalment, aquesta llista creada es retornarà a l'usuari.

4.2 Disseny Detallat

L'objectiu d'aquest apartat és el de realitzar una descripció detallada del components de les dues arquitectures del nostre projecte. Per tant, dividirem aquest apartat en dues parts: primera on explicarem els elements que han creat el model a utilitzar; segona on detallarem els components que han fet possibles la creació d'un SRI.

4.2.1. Creador del Model

Com hem vist en l'apartat anterior aquesta arquitectura es divideix en 4 parts essencials. Anirem una a una explicant-les i analitzant-les amb profunditat.

Abans d'entrar en l'explicació de les parts, aquesta arquitectura està basada en els scripts d'exemple que proporciona la llibreria **BEIR** [26], una llibreria especialitzada en l'anàlisis de models per a la recuperació d'informació i el seu entrenament. Aquesta llibreries proporciona totes les ferramentes necessàries per a aconseguir el nostre objectiu d'utilitzar el millor model que s'ajuste a la nostra tasca.

4.2.1.1. Col·lecció de Documents (Corpus)

La col·lecció de documents inicial que hem utilitzat per l'anàlisis i entrenament del model és **mMARCO** [28], una versió multilingüe del corpus **MS MARCO** [29].

MS MARCO és un conjunt de corpus de classificació molt utilitzat a l'hora d'entrenar models PLN en la tasca de recuperació d'informació. És una recopilació de respostes a consultes realitzades en Bing o altres buscadors. Malgrat això, els recursos disponibles en altres idiomes són escassos, és a dir, no trobem recopilacions de dades escrites en altres idiomes. Per això, l'equip NeuralMind, format per investigadors de diferents universitats, ha creat una versió multilingüe del MS MARCO, composta per 13 idiomes diferents. Les traduccions han sigut realitzades per traducció automàtica i han sigut avaluades utilitzant models de reclassificació monolingües i multilingües, així com una sistema de recuperació dens. L'equip d'investigadors assegura el bon rendiment d'aquests nous corpus.

MS MARCO presenta un corpus compost per diferents elements:

- **Preguntes**: Les preguntes són consultes anònimes realitzades al buscador Bing, on l'usuari busca una resposta específica. Aquelles consultes de navegació en la xarxa o amb altres intencions han sigut descartades. La Figura 4.1 mostra la distribució de les preguntes basada en el tipus de contestació possible.

Question segment	Percentage of question
Question contains	
YesNo	7.46%
What	34.96%
How	16.8%
Where	3.46%
When	2.71%
Why	1.67%
Who	3.33%
Which	1.79%
Other	27.83%
Question classification	
Description	53.12%
Numeric	26.12%
Entity	8.81%
Location	6.17%
Person	5.78%

Figura 4.1: Distribució de les preguntes en el corpus MSMARCO. Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, Tong Wang. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. url: https://arxiv.org/abs/1611.09268

  Aquesta distribució mostra com la majoria de les preguntes són de tipus *What*, i poden ser contestades per mitjà d'una descripció. En total al corpus hi ha 1.010.916 preguntes.

- **Passages**: Els *passage* són extraccions de text de documents rellevants d'internet. Per cada pregunta tenim un màxim de 10 *passage* que poden contestar-la. El corpus està compost per un total de 8.841.823.

- **Respostes**: Per cada pregunta el corpus conté zero o una resposta escrita manualment per editors humans. Aquest editors han comprés la pregunta, llegits els *passages* i creat una resposta correcta amb la informació dels *passages*.

- **Respostes ben formades**: Per cada parell pregunta-resposta, les dades poden contindre una o més respostes generades en un procés posterior per altre editor. Aquest editar analitza la resposta, i si: no està ben escrita, o és una còpia literal del *passage*, o no s'entén sense el context del *passage* i la pregunta. Aleshores, escriu una nova.

- **Document**: Els documents són les notícies, articles, etc. que han utilitzat per extraure els *passage*. En total al corpus compten amb 3.565.535 documents. Estan configurats per l'URL, el contingut i el títol.

Nosaltres ens centrarem en la versió en espanyol del mMARCO, la qual està composta per les següents parts:

- **Corpus.jsonl**: Un arxiu *jsonl* compost per 8.841.823 *passage* de diferents àmbits (esports, llibres, art, etc.), on cada un dels elements és un diccionari organitzat en 3 apartats (*id*, *tittle*, *text*). La part que haurem de processar i netejar és el text, on es troba la part escrita. La Figura 4.2 mostra un extracte de com s'organitza l'arxiu.


```
{"_id": "010000", "title": "", "text": "La presència de la comunicació entre les
 ments científiques va ser igualment important per a l'èxit del Projecte Manhatt
an com ho va ser l'intel·lecte científic. La presència de la comunicació entre l
es ments científiques va ser igualment important per a l'èxit del Projecte Manha
ttan com ho va ser l'intel·lecte científic. L'únic núvol que plana sobre l'impre
ssionant assoliment dels investigadors i enginyers atòmics és el que el seu èxit
 realment va significar; centenars de milers de vides innocents van ser destruïd
es."}
```


Figura 4.2: Exemple Documents del corpus

- **Queries.jsonl**: Paregut a l'arxiu anterior, però en aquest cas sols tenim dos elements, el primer és l'id i la consulta realitzada. La Figura 4.3 mostra un exemple de com s'organitza l'arxiu. Aquestes consultes són les preguntes del corpus MSMARCO.


```
{"_id": "121352", "text": "Definir extrema"}
{"_id": "634306", "text": "¿Qué significa Chattel en el historial de crédito"}
{"_id": "920825", "text": "Lo que fue el gran salto hacia adelante cerebro"}
{"_id": "510633", "text": "Fijadores del tatuaje cuánto cuesta"}
{"_id": "737889", "text": "¿Qué es el proceso de descentralización?"}
{"_id": "278900", "text": "¿Cuántos coches entran en la jolla concours d' elegancia?"}
{"_id": "674172", "text": "¿Qué es un número de tránsito bancario?"}
{"_id": "303205", "text": "¿Cuánto puedo contribuir a la ira no deducible?"}
{"_id": "570009", "text": "¿Cuáles son los cuatro grupos principales de elementos?"}
{"_id": "492875", "text": "Temperatura del desinfectante"}
```


Figura 4.3: Exemple de consultes en l'arxius

- **Qrels**: Dos arxius tsv (dev i train) on es relacionen les consultes realitzades amb les respostes correctes. És a dir, en un mateix element està organitzat en tres aparts, id de la *querie*, l'id del document, i el *score* que té, sempre és 1. Així, obtenim un arxiu que ens indica quina és la resposta correcta a les consultes realitzades per l'usuari.

```
query-id        corpus-id       score
300674  7067032 1
125705  7067056 1
94798   7067181 1
9083    7067274 1
174249  7067348 1
320792  7067677 1
1090270 7067796 1
1101279 7067891 1
201376  7068066 1
```

Figura 4.4: Exemple dels arxius tsv

4.2.1.2. Preprocessador del Corpus

Aquest component és un script en Python anomenat *Traductor.py*, el qual conte a la classe Traductor encarregada de preprocessar el corpus, traduir-lo i netejar-lo.

Les parts més importants del procés són la de traducció i neteja, donat que el corpus ja ve processat pels seus creadors, però el procés de traducció transforma algunes frases de "soroll" en espais en blanc. A continuació explicarem amb més detall els dos processos:

- **Traducció**: Per realitzar la traducció s'ha utilitzat un model de traducció automàtica. En aquest cas el model és *PlanTL-GOB-ES/mt-plantl-es-ca* [[30].

  Model creat pel Govern d'Espanya amb més de 92 milions d'oracions diferents, aquest model va ser desenvolupat des de zero amb ferramentes de *Fairseq*. Està basat en l'arquitectura de *Transformer-XLarge* d'escrita en la Secció 2.3.5.4. Com els autor assenyalen en la web, obté resultats similars als models de SoftCatalà i Google. Pel que vam pensar que era una bona idea d'utilitzar.

  Per poder fer ús d'aquest models és necessari instal·lar-se dues llibreries *pyonmttok* [31], per poder fer ús del seu *tokenitzador* ràpid i personalitzable, i *CTranslate2* [33], que ens permet usar el seu traductor optimitzat. Seguint un poc en aquesta línia explicarem el procés complet de traducció:

  1. Carregar els elements necessaris: Hem de carregar en memòria tant el model a utilitzar, com el *tokenitzador* i el traductor que utilitzarà el model carregat. Indicant quins són els paràmetres a utilitzar en cada cas, el nostre cas utilitzem els paràmetres per defecte que t'indica la mateixa pàgina del model.
  2. Traducció: Carreguem el corpus i per cada element d'aquest traduïm el seu text. Aquesta traducció es realitza seguint el procés següent: primer tokenitzem la frase a traduir, la traduïm per mitjà del traductor, finalment la destokenitzem per reconstruir la frase. La frase traduïda és guardada en un diccionari, mantenint el format original.
  3. Escriptura de l'arxiu traduït: Una vegada hem traduït una quantitat X de frases, anem escrivint en un arxiu aquest diccionari creat amb les frases, i una vegada ha finalitzat netegem el diccionari per a ocupar menys memòria.

- **Neteja**: El procés de neteja és més simple, una vegada repassat els resultats, hem trobat els següents errors:

  – Noms propis de persones o ciutats: En aquest cas el traductor marca amb un caràcter especial les paraules. Per tant, la solució va ser eliminar aquest caràcter en totes les entrades on ocorria.

- frases de soroll: En el mateix corpus existeixen frases de soroll, on el text són caràcters especials (−) o espais en blanc tots seguits. El traductor retornava un resultat marcat i eliminava els caràcters. La nostra solució va ser copiar la frase soroll sencera al nostre corpus traduït.

La classe Traductor produirà un arxiu traduït per cada arxiu que compren el corpus en castellà a utilitzar. En conseqüència, obtindrem 4 arxius nous en català. Finalment, unirem tots els arxius iguals un de sol, per tindre tant frases en castellà com en català en el corpus d'anàlisi i entrenament, obtenint 4 arxius finals. **Precaució**: Els id del corpus traduït s'han modificat tots per no coincidir amb els originals

4.2.1.3. Analitzador de Models

Aquest component està creat per mitjà d'un script anomenat *Evaluate_SBert*. És un script proporcionat per la llibreria de Beir [26], i modificat per poder utilitzar un corpus personalitzat . El procediment d'anàlisis és el següent:

1. **Càrrega del corpus**: Carreguem el corpus en memòria, però en aquest cas, sols utilitzarem la part de *dev*, donat que volem analitzar els resultats no entrenar.
2. **Càrrega del model**: En aquest part carregarem el model a analitzar, i el transformarem en un recuperador. És a dir, crearem un recuperador d'informació que utilitze el model carregat per transformar les consultes en vectors i després per mitjà de la similitud cosinus, descrita en la Secció 2.3.3.1 trobar els resultats més similars.
3. **Obtenció dels resultats**: Utilitzarem les consultes del corpus de *dev* per obtindre les llistes dels resultats.
4. **Avaluació**: Una vegada obtingut els resultats, els analitzarem per mitjà de mesures comentades en la Secció 2.4.2 com NDCG, MAP, Exhaustivitat, Precisió. Escriurem en un arxiu tots els valors de les mesures per a la seua futura comparació.

Finalment, farem una comparació manual per trobar aquell model amb millors prestacions en la tasca que requerim.

4.2.1.4. Entrenador de Models

Com en el component anterior, es tracta d'una modificació del script creat per Beir [26], on podem utilitzar un corpus personalitzar. L'objectiu d'aquest script és entrenar un model per a la tasca de recuperació d'informació. El procés per a l'entrenament és el següent:

1. **Càrrega del corpus**: Carreguem el corpus sencer en memòria per a la seua utilització en l'entrenament. És a dir, utilitzarem les dues parts que el componen tant el *train* com el *dev*.
2. **Càrrega del model**: En aquesta part, carregarem un model des del *Hub* de ***Hugging Face*** [33]. Aquest model serà el triat en l'analitzador de Model. Una vegada carregat, el convertirem en un recuperador. Utilitzarem aquest propi recuperador per posar a punt les dades a utilitzar.
3. **Configuració de l'entrenament**: Crearem tots els elements necessaris i configurarem els paràmetres, per poder ajustar l'entrenament a la tasca que requerim i les dades que tenim. La funció de pèrdua utilitzada en aquest entrenament dependrà del model.

4. **Entrenament**: Una vegada configurat tot, passarem a entrenar el recuperador creat amb el model, per tal d'adequar el model a les dades que tenim processades. Aquest entrenament serà cronometrat per veure el temps total.

Una vegada ja hem entrenat el model, es tornarà a analitzar aquest per veure com el seu rendiment ha variat respecte a l'original.

4.2.2. Arquitectura SRI

La segona estructura implementada en aquest projecte és la d'un recuperador de la informació. En particular, utilitzarem una estructura que representa els components principals en diferents *scripts* i arxius.

Per al desenvolupament d'aquesta arquitectura hem seguit les instruccions i exemples proporcionats per la llibreria **Haystack** [27]. Haystack proporciona les ferramentes (mètodes, classes, etc.) per poder crear un SRI eficient i amb un rendiment alt. No obstant, hem hagut de treballar per poder adaptar-lo a les nostres dades i per personalitzar-ho al nosre gust.

4.2.2.1. Col·lecció de Documents

La col·lecció de documents utilitzada en aquest projecte per crear el recuperador d'informació és el corpus **DACSA** [9]. El corpus DACSA, creat per l'equip d'investigadors ELiRF, és una recopilació de més de 6 milions d'articles (2 milions en català i 4 milions en castellà), tots aquests recopilats per mitjà d'un rastrejador web. Posteriorment, l'equip d'investigador va reduir el corpus a 725.184 mostres en català i 2.120.649 mostres en castellà per mitjà d'un procés de selecció i neteja. Aconseguint un corpus en dos idiomes que conté notícies d'actualitat i de diferents tòpics. Analitzant aquest corpus trobem que és ideal per al nostre projecte, ja que, conté els dos idiomes que volem utilitzar, i a més notícies de diferents àmbits per al nostre recuperador d'informació.

Entrant en més detall, cada element de la col·lecció de DACSA és un articles o notícia publicada, on es recopila tota la informació possible en diferents claus (identificador, cos de la notícia, resum i més). La part més important en el nostre projecte és el cos de l'article, d'on extraem diferents parts per a la seua posterior indexació. Agafarem la part de *article*, i anirem dividint-la en unitats bàsiques d'indexació (paraula, frase o paràgraf) i les indexarem. La Figura 4.5 mostra un exemple del corpus DACSA.

```
{"source": "ara", "id": "5e05cd116160473f3ad85c57",
 "url": "https://www.ara.cat/politica/Puigdemont-Comin-euroordre-immunitat-Fiscalia-Suprem_0_2367363320.html",
 "summary": "El ministeri fiscal apunta que les eleccions europees van ser posteriors als fets que se'ls imputa.",
 "article": "La Fiscalia ha demanat aquest dilluns al Tribunal Suprem que mantingui les euroordres cursades contra l'expresident
Carles Puigdemont i l'exconseller Toni Comín, que la justícia belga està estudiant, i reclama a l'alt tribunal que sol·liciti al
Parlament Europeu \"amb la màxima urgència possible\" que suspengui la seva immunitat. És la resposta del ministeri fiscal a la
situació dels dos polítics catalans respecte a la sentència de dijous passat del Tribunal de Justícia de la Unió Europea, que
reconeixia la immunitat d'Oriol Junqueras com a eurodiputat i que va fer que l'Eurocambra posés en marxa el procés per
reconèixer i acreditar Puigdemont i Comín com a eurodiputats. La Fiscalia considera que la sentència europea no ha de modificar
la situació de l'expresident i l'exconseller perquè la \"gravetat\" dels fets que s'imputen a Puigdemont i Comín –anteriors a
les eleccions europees, puntualitza l'escrit– i el fet que es trobin \"fugats\" de la justícia fan necessari que es mantingui
l'euroordre i la declaració de rebel·lia. \"El fet que hagin comès delictes molt greus i es trobin en rebel·lia [...] justifica
la necessitat de mantenir les mesures cautelars que pesen sobre tots dos i, particularment, les ordres de detenció europees\",
afirma la Fiscalia. El ministeri fiscal, a més, retreu a Puigdemont i Comín, sense presentar cap prova, que es presentessin a
les eleccions europees per adquirir la immunitat i no ser processats. \"El que pretenia en realitat era acollir-se al paraigua
de la immunitat que, al seu parer, li concedia l'elecció com a eurodiputat amb el propòsit d'obtenir la llibertat i eludir el
procés penal\", valora la Fiscalia en el seu escrit al Suprem. El Tribunal Suprem va donar cinc dies a totes les parts per
presentar les seves al·legacions a la sentència europea. Per la Fiscalia, la sentència no canvia la situació de Comín i
Puigdemont. Per això exigeix al jutge del Suprem Pablo Llarena, instructor del cas del Procés, que comuniqui al Parlament
Europeu que es mantenen totes les mesures contra Puigdemont i Comín amb l'objectiu de frenar el seu reconeixement com a
eurodiputats. També demana a la justícia belga que esperi a resoldre les euroordres fins que l'Eurocambra decideixi sobre la
seva immunitat.",
 "article_nwords": 361,
 "summary_nwords": 16,
 "similarity": 0.0,
 "html_detected": false,
 "too_short": false,
 "too_similar": false,
 "not_ended": false,
 "lang": "ca",
 "lang_prob": 1.0,
 "lang_ca_prob": 1.0}
```


Figura 4.5: Exemple de mostra del corpus DACSA

Tot seguit comentarem d'on provenen aquestes mostres:

- **Català**: La part en català compren 9 diaris diferents. Els tòpics principals de les notícies són de política, economia, esport i societat. La Figura 4.6 mostra les estadístiques completes de la partició en català. On podem veure diferents estadístiques com: el dimensió del vocabulari, frases per documents, nombre total de documents i més.

			Article			Summary		
Source	#Docs	Tokens	Vocabulary size	Sents per doc	Words per sent	Vocabulary size	Sents per doc	Words per sent
Training	636,596	316,817,625	1,206,292	17.39	28.62	206,616	1.17	20.36
Validation	35,376	17,831,029	258,999	16.17	31.17	51,940	1.15	20.93
TESTI	35,376	17,704,387	262,148	16.13	31.03	51,958	1.15	20.89
TESTNI	17,836	15,882,219	247,154	35.38	25.17	45,997	1.56	25.93
Set	725,184	368,235,260	1,326,343	17.71	28.67	223,978	1.17	20.59

Figura 4.6: Estadístiques de la partició en català. [9]

- **Castellà**: La part més gran del corpus, compren un total de 21 diaris diferents, dels quals hem obtingut 2.120.649 notícies. Els tòpics principals que trobem són els mateixos que per a les notícies en català. La Figura 4.7 mostra les estadístiques completes de la partició en castellà.

			Article			Summary		
Source	#Docs	Tokens	Vocabulary size	Sents per doc	Words per sent	Vocabulary size	Sents per doc	Words per sent
Training	1,802,919	1,172,626,265	2,920,894	23.94	27.17	454,179	1.24	21.99
Validation	104,052	67,669,381	550,213	23.01	28.27	109,460	1.21	23.36
TESTI	104,052	67,363,994	550,910	22.93	28.23	109,706	1.21	23.34
TESTNI	109,626	59,603,306	447,679	16.25	33.46	116,201	1.35	36.84
Set	2,120,649	1,367,262,946	3,189,783	23.44	27.50	516,307	1.24	22.95

Figura 4.7: Estadístiques de la partició en castellà. [9]

4.2.2.2. Indexador

Aquest component és l'encarregat d'agafar el corpus i transformar-lo en representacions vectorials que s'indexaran per a la seua futura cerca. El procés és el següent:

- **Desglossament del corpus:** Per cada iteració carreguem una notícia. Aquestes notícies les anem transformant de diccionaris a documents, la classe especial d'Haystack. Aquesta classe Document és molt pareguda a un diccionari, on anem desglossant la notícia per seccions. Tots aquest documents, són guardats en una llista per al seu futur processament.
- **Processament del documents**: Una vegada tenim la llista amb tots els documents és necessari processar-los. En el nostre cas crearem un processador amb la següent configuració:
  - Netejar línies en blanc, espais en blanc al començament o final de les frases.
  - La tokenització del text es farà per paraules. Cada documents tindrà un màxim de 150-200 paraules, el recomanat per la documentació d'haystack.
  - Els documents creats respectaran els límits de les frases. En tindre un màxim de paraules per frase, es podria donar el cas de trencar frases. Nosaltres mitjançant un paràmetre, permetem al preprocessador agafar més paraules per completar la frase.
  - Màxim nombre de caràcters per documents al voltant dels 30.000 com proposa Haystack.

  Una vegada està configurat, hem d'utilitzar-lo en la llista de documents per processarlos i crear les entrades com nosaltres volem.
- **Escriure Documents**: Una vegada ja tenim tots els documents creats i processats, és moment d'escriure'ls en la base de dades. Guardem tots els documents en una base de dades creada per Haystack. Aquesta base de dades s'utilitzarà en un últim moment per recuperar el document real i retornar-lo a l'usuari.
- **Transformació en embedding**: Una vegada ja tenim tots els documents processats i escrits, l'últim pas és crear les representacions vectorials. Per fer aquest procés utilitzarem el model entrenat. Aquest és un procés lent, però sols és necessari ferho una volta.

  Per aquest procés és necessari carregar en memòria el model a utilitzar i assignar-lo a l'objecte recuperador. Una vegada ja tenim tot, podem procedir a transformar les entrades en *embeddings*. Aquestes representacions vectorials creades seran les que s'indexaran, obtenim un índex sols de vectors.

Una vegada realitzat aquest procés obtenim l'índex amb totes les entrades transformades en representacions vectorials i una base de dades on hem guardat tots els documents.

4.2.2.3. Índex

L'índex és un dels components més importants dins d'aquest projecte. Aquest component determina la capacitat del teu sistema en termes de memòria i eficiència, un bon índex permet indexar una quantitat major de documents i permet un procés de recuperació més eficient. Per exemple, en altre projecte vàrem crear un índex manualment,

utilitzant l'estructura *KDTree*, el que suposava una ràpida comparació entre *embeddings*, però no vàrem tindre memòria suficient per a poder indexar el corpus sencer.

La selecció d'un bon índex és fonamental per crear un bon sistema de recuperació d'informació. En aquest projecte l'índex utilitzat és l'índex ***Flat***. Aquest tipus d'índex són el que millor precisió obtenen, però els que més memòria utilitzen, donat que, no comprimeixen els vectors (no agreguen cap sobrecàrrega en ells), són molt pareguts als vectors de C++. L'ús d'aquest índex són recomanables quan parlem d'un número de menys d'uns 5 milions de documents. Si parlem en termes de dades, el nostre projecte ronda els 3 milions, per la qual cosa és perfecte l'ús d'aquest tipus d'índex per emmagatzemar totes les entrades.

Existeixen altres tipus d'índex que explicarem en un apartat posterior, i on entrarem més en detall en l'elecció.

4.2.2.4. Recuperador

El recuperador en aquest projecte s'encarrega d'unir dos components de l'estructura dels SRI, la interfície i l'algoritme de cerca i rànquing. Aquesta unió es graciés a les facilitats que aporta Haystack a la creació de SRI, donat que permet unir tots els procediments mitjançant *pipelines* [1]. El procés que realitza el recuperador en entrar una consulta és el següent:

1. **Definir els seus paràmetres inicials**: Abans d'atendre a l'usuari, crea tots els components principals i els interconnecta entre ells:

   (a) ***Reader***: Primer element creat que s'encarrega de llegir la consulta i donat un conjunt de documents tornar una llista ordenada dels més rellevants. En el nostre cas utilitzarem un ***FARMReader***, permet paral·lelitzar processos i obtindre una millor precisió. Aquest *Reader* utilitza un model diferent del *Retriever*, que està especialitzat a respondre preguntes, cerca informació exacta que respon a la pregunta.

   (b) ***Retriever***: Segon element creat que s'encarrega de recórrer l'índex, l'emmagatzematge de documents, i retornar una sèrie de documents rellevants a la consulta. En el nostre cas utilitzarem el ***EmbeddingRetriever*** especialitzat a utilitzar representacions vectorials i fer una comparació de proximitat. Té la facilitat de poder utilitzar models directament des d'Hugging Face.

   (c) ***Pipeline***: Finalment, creem la *pipeline* que connecta el *Reader* i el *Retriever*. Donat que, utilitzarem el *Retriever* en un primer moment per extraure un nombre major de documents, i després utilitzarem el *Reader* per crear el rànquing entre eixos documents, ja que fa una cerca més exhaustiva. És a dir, el procés funciona de la següent forma: la consulta espasada al *Retriever* i aquest per mitjà de la comparació de similituds entre representacions vectorials recupera *X* documents amb les representacions més similars. Una vegada tenim aquest documents, el *Reader* per mitjà del seu model, intenta contestar en precisió la consulta, llegint en profunditat el documents i retornant aquells documents més rellevants, creant un rànquing de rellevància.

2. **Fer la consulta**: El component agafa la consulta de l'usuari, i posa en marxa tota la *pipeline* creada, la qual transforma la consulta i crea el rànquing de resultats.

[1] canonades de dades, les quals permeten connectar processos, uneixen l'eixida d'un procés amb l'entrada d'altre

3. **Resposta**: Una vegada tenim la reposta amb el rànquing de resultats, retornem aquest rànquing per mitjà de la pantalla i d'un document escrit.

Com podem deduir, **l'algoritme de cerca i rànquing** és la *pipeline* creada, ja que, recupera els documents més rellevants i després crea el rànquing. La mesura utilitzada per crear la comparació de similitud és la similitud cosinus i pel rànquing, el *Reader* utilitza models PLN potents que entenen el llenguatge i són capaços de trobar la resposta a la consulta i ordenar-les per rellevància. Per altra banda, el recuperador també fa la funció de **interfície**, ja que, és l'element que es connecta amb l'usuari.

4.3 Tecnologia Utilitzada

En aquest nou apartat farem una explicació detallada de totes les ferramentes, entorns i llibreries que hem utilitzat per portar a terme aquest projecte. No sols les explicarem, sinó que detallarem les seues aportacions al projecte i el motiu de la seua tria.

4.3.1. Entorn de desenvolupament

En primer lloc explicarem quins han sigut els entorns de desenvolupament que hem utilitzat en aquest projecte.

4.3.1.1. Visual Stuido Code

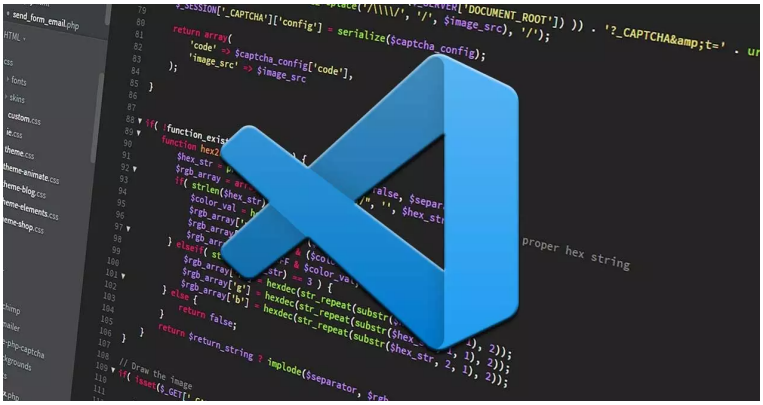
Visual Studio Code (VSC) és un editor de codi font desenvolupat per Microsoft. És un software lliure i multiplataforma, que està disponible tant per Windows com per Linus i MacOS. VSC té disponible nombroses característiques que et poden ajudar a l'hora de desenvolupar el teu projecte, com poden ser:

- **Multiplataforma**
- **IntelliSense**: Complement que ajuda a una escriptura de codi àgil i ràpida. Donat que permet l'edició de codi completa, autocompletat i ressaltat de la sintaxi. Aquest complement dona suggeriments de codi, finalitza línies per tu, etc. És a dir, pot llegir el que estàs escrivint, comprendre-ho i donar-te suggeriments de com acabar-ho. A més, és molt personalitzable.
- **Depurador**: Element que ajuda a detectar errors en el codi. Permet no sols fer-ho després de l'execució sinó abans. Ressalta en color roig el codi on probablement hi ha un error.
- **Extensions**: Les extensions ens permeten personalitzar i agregar noves funcionalitats. Com per exemple, extensions que ens permeten programar en diferents llenguatges, canviar el color de l'editor, etc. Permeten tindre una millor experiència, i millorar el rendiment del nostre codi.

Visual Studio Code ha sigut l'editor de codi seleccionat per desenvolupar el nostre projecte. Els factors de la seua tria han sigut els següents:

- Experiència utilitzant aquest editor.
- Editor personalitzable al meu gust i de fàcil ús.

- Extensions que ens permeten un ràpid i eficient desenvolupament.
- La característica de depuració és eficaç i ajuda molt a no cometre errors.



4.3.1.2. Anaconda

Anaconda és una plataforma gratuïta de codi obert, que permet la distribució del llenguatges de Python i R, centrada en el desenvolupament de projecte de *Machine learning*, *Data science* i científics. Continuum.i és l'actual propietari d'Anaconda, una empresa especialitzada en el desenvolupament en Python. Anaconda ofereix una simplificació de la gestió i implementació de paquets. A més a més, Anaconda proporciona una gran quantitat de paquets i llibreries ja inclosos.

Anaconda Navigator

Anaconda Navigator és la part gràfica d'Anaconda és una GUI (Graphical User Interface). Permet el control total d'Anaconda sense la necessitat de conéixer i utilitzar els comandaments bàsics o avançats. Anaconda Navegator permet un ús més intuïtiu i fàcil de les prestacions i funcionalitats d'Anaconda, acomodant el seu ús a les nostres necessitats.

En aquest projecte hem decidit utilitzar-les, donat que ens permet crear entorns de programació amb tots els paquets i llibreries necessàries sense el treball d'administrar totes les dependències ni els problemes que genera. No sols això, sinó que ens permet l'ús de JupyterLAB amb els paquets instal·lats en el nostre entorn. Utilitzarem d'Anaconda Navigator en la nostra màquina pròpia, ja que tinc una rang de llibertats més gran i una interfície gràfica general (pantalla ordinador), per tant, podré fer totes les proves d'instal·lació i producció d'una manera àgil i ràpida. Utilitzarem Anaconda en la màquina del laboratori.

4.3.1.3. Jupyter Notebook

Jupyter Notebook és un entorn interactiu basat en la web per crear documents. Aquest documents són arxius JSON, que segueix un esquema versionat, on hi ha una sèrie de cel·les interconnectades que contenen codi, text o altres elements. Aquestes cel·les a l'estar connectades entre si, permeten crear programes i scripts. La particularitat més gran és que permeten guardar valor entre cel·les, fer una execució asíncrona de les cel·les i, en general, utilitzar les cel·les com blocs de codi on anar executant diferents coses. El llenguatge de programació que utilitza és Python.

L'ús de Jupyter Notebook en el nostre projecte és per la funcionalitat de fer de tester. Gràcies a poder dividir el codi en diferents cel·les, podem anar executant proves sense la necessitat d'executar tot el script sencer. És a dir, la capacitat de guardar els resultats passats, i poder escriure codi en diferents cel·les i que estiguen connectades, ens dona la possibilitat d'anar fent proves a diferents parts del nostre codi.

4.3.1.4. Tardis

Tardis és una màquina pròpia del grup d'investigadors ELiRF, consta de dos gràfiques NVIDIA GeForce RTX 3090, les quals ens proporcionen els recursos necessaris per a fer l'entrenament del model i la creació i posada en marxa del SRI. Ens garanteix unes prestacions altes amb una velocitat adecuada.

En aquest projecte hem utilitzat dues màquines diferents:

1. **Màquina local**: Màquina personal que consta d'una gràfica de menor prestacions, on farem un primer desenvolupament dels dos sistemes a crear, on portarem a terme diferents experimentacions de prova. Aquesta experimentació inicial la fem en aquesta màquina per la seua llibertat d'ús i perquè arreglar errors de codi i problemes d'entorn és més senzill.

2. **Tardis**: Una vegada hem realitzat les proves a xicoteta escala i sense cap mena d'error, passem a treballar en Tardis on desenvoluparem els sistemes utilitzant el corpus sencer. Tots els errors que poden ocurrir ja els tindrem estudiats i serà tot molt més fàcil.

4.3.2. Llenguatges de Programació

En segon lloc, explicarem quin ha sigut el llenguatge de programació triat. Explicarem, també, quines funcionalitats addicionals hem utilitzat.

4.3.2.1. Python

El llenguatge de programació seleccionat per desenvolupar aquest projecte és Python, introduit en la Secció 2.5.1.1, un dels llenguatgs més utilitzats per al desenvolupament de projectes PLN. Els principals avantatges que ens proporciona Python són la gran quantitat de llibreries i paquets que podem utilitzar en aquest projecte, l'enorme flexibilitat que té com a llenguatge i el suport d'una comunitat molt gran que et poden resoldre problemes amb molta rapidesa. Un altre dels motius de la seua selecció és la meua experiència i domini del llenguatge.

El llenguatge proporciona mòduls amb diferents funcionalitats, en aquest projecte hem utilitzat els següents:

- ***json***: Mòdul centrat en el tractament d'arxius *.json* i *.jsonl*, proporcionant els mètodes necessaris per a la seua lectura i escriptura.
- ***os***: El mòdul *os* proporciona les ferramentes necessàries per a poder tractar aspectes relacionats amb el sistema operatiu d'una manera simple i ràpida. En aquest projecte ha sigut utilitzat per recórrer directoris i arxius.
- ***sys***: Mòdul que prové accés a variables i funcions que interactuen amb l'intèrpret de Python. Utilitzat per proporcionar als scripts els seus arguments d'entrada.
- ***typing***: Aquest mòdul proporciona suport per anotacions de tipat en temps d'execució.
- ***time***: Mòdul encarregat de dotar de ferramentes per poder fer un tractament del temps. Utilitzat per calcular els costos temporals dels processos.
- ***logging***: Mòdul amb mètodes per crear un sistema flexible de missatges d'esdeveniments i d'errors.

4.3.3. Llibreries i Paquets

En aquest subapartat donarem una explicació detallada de totes les llibreries utilitzades en aquest projecte i quines funcionalitats hem utilitzat.

4.3.3.1. Ctranslate2

La llibreria Ctranslate2 [32], és un llibreria per a C++ i Python especialitzada per realitzar inferències amb transformers d'una forma eficient. Aquesta llibreria implementa moltes tècniques d'optimització per al rendiment, com la reordenació de blocs, fusió de capes, etc. per accelerar i millorar el rendiment dels transformers, tant en GPU com en CPU. La biblioteca té les següents característiques claus:

- **Execució ràpida i eficient**: Millora i accelera l'execució en tasques relacionades amb models, l'entrenament, la inferència, etc. Utilitza menor quantitat de recursos per portar a terme aquestes tasques.
- **Quantificació i precisió reduïda**: La serialització i càlculs de models admeten pesos amb precisió reduïda [2].
- **Compatibilitat i detecció automàtica de CPU**: Compatibilitat amb molts CPU i la seua detecció automàtica.
- **Execució en paral·lel i asíncrona**: Capacitat de processament de diversos blocs en paral·lel i asíncronament.
- **Ús dinàmic de la memòria**: La utilització de la memòria varia dinàmicament dependent de la grandària de dades requerides sense deixar de complir els requisits de rendiment.
- **Pes lleuger en disc**: Models són 4 voltes més xicotets gràcies a la quantificació.
- **Integració simple**: Poques dependències i API simples per a C++ i Python.

Aquesta llibreria s'utilitza en el procés de traducció del corpus. Utilitzem la classe **Translator** per crear un objecte traductor que agafant una frase tokenitzada com entrada, i utilitzant el model assignat, és capaç de traduir-la. Per fer aquesta traducció és necessari utilitzar el mètode *translate_batch()*.

Aquesta llibreria ens proporciona un traductor eficient i ràpid gràcies a les característiques exposades amb anterioritat, fent que el procés de traducció siga més ràpid.

4.3.3.2. Pyonmttok

La llibreria Pyonmttok és un envoltori per a *OpenNMT/Tokenizer* [31]. OpenNMT és un ecosistema de codi obert per a la traducció automàtica i per a l'aprenentatge neuronal. El seu Tokenitzador, que nosaltres utilitzem, és una llibreria de tokenització de text ràpida i personalitzable.

En el nostre projecte l'utilitzem per crear un tokenitzador **Tokenizer** amb el model seleccionat, posteriorment, tokenitzar les frases del corpus mitjançant el mètode *tokenize()*. Una vegada traduides s'utilitza el mètode *detokenize()* per destokenitzar les frases i obtindre la frase completa traduida.

4.3.3.3. Hugging Face Hub

Hugging Face és una empresa Nord-americana que desenvolupa aplicacions i ferramentes per crear projectes utilitzant l'aprenentatge automàtic.

[2] tècnica que redueix la grandària del model i accelera l'execució, en algunos casos amb el cost de perdua de precisió

Hugging Face Hub, creada per Hugging Face, és un repositori en línia. Els seus creadors exposen que guarda més de 120.000 models, 20.000 conjunt de dades i 50000 demos.

En el nostre cas, l'utilitzem per poder carregar i guardar els models a utilitzar en aquest projecte. Tenim 4 casos d'ús, el model per a la traducció, els diferents models per l'anàlisis i comparació d'ells, l'entrenament del millor model i l'ús per crear les representacions vectorials dels models.

4.3.3.4. Sentence Transformer

Utilitzem SentenceTransformer, descrit a la Secció 2.5.3.1, per a la càrrega del models des del Hub d'Hugging Face. És a dir, utilitzem la classe i mètode *Sentence_transformer()* per carregar en memòria els models que volen utilitzar al llarg del projecte.

4.3.3.5. Beir

Beir [26] és una llibreria que aporta un punt de referència de l'avaluació sòlid i heterogènic que conté diverses tasques per a la recuperació d'informació [26]. En total, conté 18 conjunts de dades que es divideixen en diferents tasques. Beir proporciona un marc comú i senzill per l'avaluació de models PLN en les diferents tasques de RI.

Les característiques principals són:

- Possibilitat de preprocessar el teu propi conjunt de dades o utilitzar un conjunt de dades de referència ja preprocessades.
- Flexibilitat i cobertura amplia a l'hora de configurar l'avaluació, cobreix molts punts importants tant per al treball acadèmic com per al treball industrial.
- Dota de totes les ferramentes necessàries per a avaluar qualsevol model actual (lèxic, dens, dispers, etc.), és a dir, inclou tota classe d'arquitectura SRI.
- Possibilitat l'autoavaluació del teu model, per mitjà d'un marc senzill i còmode a l'ús, utilitzant les mètriques més actuals.

Aquesta llibreria consta de 18 corpus de dades diferents, aquest corpus són tots en angles. Però Beir, en les últimes actualitzacions, ha donat la possibilitat d'utilitzar corpus multilingües en el seu marc de treball.

En aquest projecte hem utilitzat els següents elements:

- **Corpus**: Hem utilitzat el corpus **mMarco**, que és una traducció directa del corpus **MsMarco** detallat en la Secció 4.2.1.1.
- **Avaluació de Models**: Script encarregat d'avaluar els models SBERT amb diferents mesures.
- **Entrenament del Model**: Script per entrenar un models amb dades personalitzades.

La llibreria Beir ens ha proporcionat un marc de treball on poder avaluar els diferents models actuals PLN i un model entrenat. És a dir, ens dona la certesa que el model que utilitzarem és el millor model possible per a la asca d'entre els avaluats.

4.3.3.6. Haystack

Haystack és un *framework* de còdi obert per crear sistemes de cerca que funcionen d'una forma intel·ligent en grans col·leccions de documents [27]. Les principals característiques d'aquest framework ja s'han detallat a la Secció 2.5.3.2.

En aquest subapartat ens centrarem en els diferents elements que hem utilitzat en el nostre projecte:

- **Document**: La classe Document conte un tros d'informació (text, imatge, taula) amb el seu id i metadades. També pot arribar a contindre la representació vectorial i la puntuació donada per un model PLN. Aquesta classe s'utilitza com a *Data Handler* [3] i conté els següents apartats:
  - content: on és emmagatzema el tros d'informació
  - content_type: Indica de quin tipus és la informació emmagatzemada.
  - id: L'id de la informació.
  - meta: Emmagatzema totes les dades extra de la informació, com per exemple, els URL, el títol, etc.
  - score: És opcional, i indica quina és la puntuació obtinguda en la tasca de similitud o altra.
  - embedding: Opcional, emmagatzema l'embedding representant del tros d'informació.
  - id_has_keys: Opcional, si id té alguna clau que la represente.
- **FileConverter**: Element especialitzat a convertir els arxius de dades en la classe Document. Accepta diferents formats de text (pdf, txt, json). En el nostre cas utilitzarem la classe ***JSONConverter*** que s'especialitza en la conversió de documents json i jsonl.
- **PreProcessor**: Element encarregat del processament dels documents. La classe **Preprocessor** agafa els documents com entrada i el processa per transformar-los en documents nets. La classe compta amb un gran nombre de paràmetres configurables per adaptar-los a les teues necessitats. Els paràmetres són:

[3] format organitzatiu que té les dades

  - clean_empty_lines: Transforma 3 o més línies blanques contínues en 2 línies blanques.
  - clean_withespace: Elimina espai en blanc al començament o final de cada línia.
  - clean_header_footer: Elimina títols o capçaleres que es repeteixen en moltes pàgines.
  - remove_substring: Elimina subcadenes específiques en el text.
  - split_by: Determina quina és la unitat de divisió del text. Pot ser, "word", "sentence" o "passage".
  - split_length: Determina nombre màxim d'unitats per document.
  - split_respect_sentence_boundary: Respectar la construcció de les oracions, no tallar-les a mitad.
  - split_overlap: Estableix la quantitat de superposicions entre dos documents adjacent després d'una divisió.
  - max_chars_chech: Longitud màxima d'un document. Si se supera el límit el document es dividirà en dues parts.

- **DocumentStore**: Aquest element es pot considerar com una base de dades que emmagatzema els textos i les seues metadades. Aquest element es connecta amb el Retriver per proporcionar tota la informació necessària per a contestar a la consulta. Hi ha diferents tipus de DocumentStore:

  - **ElasticSearch**: DocumentStore que utilitza la tecnologia d'ElasticSearch per fer les seues cerques. Els seus principals avantatges són l'escalabilitat alta i velocitat, aconseguint un DocumentStore que permet connectar-ho a la web i rebre moltes dades, sense la necessitat que siguen iguals, i emmagatzemar-les.

    ElasticSearch [34] és un motor de cerca i anàlisi distribuïda per tota classe de dades. Les seues principals característiques són que és simple, de naturalesa distribuïda, alta velocitat i escalabilitat. Aquesta ferramenta és un element d'un conjunt més gran ElasticStack.
  - **InMemory**: DocumentStore que guarda en memòria tots els documents.
  - **Milvus**: DocumentStore que utilitza la tecnologia de Milvus [35]. Milvus és una tecnologia que permet la creació de Base de dades especialitzades en representacions vectorials massives. També permet la comparació d'aquestes representacions per mitjà de la seua distància.
  - **OpenSearch**: Com en els casos anteriors és un DocumetStore que utilitza la tecnologia proporciona per OpenSearch [36]. OpenSearch és un paquet de software de codi obert que proporciona ferramentes escalables, flexibles i extensibles per crear aplicacions cerca i anàlisis. Concretament, aquest DocumentStore utilitza la tecnologia de les bases de dades vectorials que proporciona OpenSearch, que permeten guardar representacions vectorials i les seues metadades.
  - **Pinecone**: PineconeDocumentStore és una base de dades vectorial ràpida i escalable que admet cerca filtrada. És un magatzem de documents administrats, és a dir, els vectors s'emmagatzemen en el núvol. Aquest DocumentStore utilitza la tecnologia de Pinecode [37].
  - **Qdrant**: QdrantDocumentStore és una tecnologia creada per Qdrant i que el mantenen ells. S'utilitza per a la cerca de vectors d'altes dimensions i admeten diverses mètriques de similitud.

  - **SQL**: Utilitza les bases de dades sql per emmagatzemar els documents.
  - **Weaviate**: WeaviateDocumentStore utilitza Weaviate [38], que proporciona les ferramentes per crear una base de dades vectorial. La seua característica principal és l'alta escalabilitat que proporciona.
  - **FAISS**: Aquest DocumentStore utilitza la ferramenta Faiss [50]. Faiss utilitza una base de dades de SQL per emmagatzemar els documents i després un índex Faiss. Les prestacions que proporciona aquesta ferramenta és una escalabilitat gran amb SQL i gran eficiència i eficàcia amb l'índex de Faiss. Proporciona les ferramentes necessàries per a utilitzar tant els documents com les seues representacions vectorials a l'hora de respondre les consultes.

  En aquest projecte hem utilitzat **FAISS** degut a la seua eficiència a l'hora d'indexar documents i la seua capacitat de reduir la memòria utilitzada. En un pròxim apartat s'explicarà amb major detall els motius de la decisió.

- **Reader**: El Reader, és una classe especialitzada a contestar consultes dels usuaris. El Reader agafa una consulta i un conjunt de documents i retorna una resposta en forma de tros de text. El Reader utilitza els models PLN per a realitzar un control de qualitat i ordenar per rellevància. Trobem diferents Readers a utilitzar:

  - **TransformerReader**: Utilitza el marc de treball d'Hugging Face [33] per a la construcció del Reader, i està basat en transformers.
  - **FARMReader**: Utilitza la tecnologia que proporciona FARM [51] per crear el Reader, com en l'anterior està basat en models transformers. FARM és un marc de treball especialitzat en el desenvolupament de transformers, produeix que l'aprenentatge de transformers siga senzill, ràpid i preparat per a la producció.

  El Reader triat ha sigut el de FARM.

- **Retriever**: El Retriever, recuperador en català, és l'encarregat de realitzar la recuperació dels documents més rellevants del DocumentStore davant d'una consulta. El Retriever és un element essencial per contactar els DocumentStore i el Reader, donat que poden augmentar la velocitat de recuperació si fem primer que el Retriever recupere el X documents més rellevant i a continuació, el Reader agafe el Y més important i recupere els trossos més rellevants.

  En el nostre cas, com utilitzem representacions vectorials, hem d'utilitzar l'EmbeddingRetriever, que fa dues funcions: La primera crear les representacions vectorials del documents i crear un índex amb aquestes representacions. La segona s'encarrega de transformar la consulta i comparar-la en les diferents representacions de l'índex per trobar les més semblants.

Tots aquest elements són els necessaris per a crear un recuperador d'informació. Una de les principals millores que aporta Haystack és l'alta abstracció, dota aquest elements que estan enfocats ja al seu treball i on s'interconnecten a la perfecció. Aconseguint, crear sistemes amb molta eficiència i utilitzant poc de codi.

4.3.3.7. FAISS

Facebook AI Similarity Search (Faiss) és una llibreria que permet la cerca de similituds eficients desenvolupada per Facebook. Aquesta llibreria permet la construcció d'índex de representacions vectorials i la seua cerca de similitud. No sols això, sinó que presenta tecnologies revolucionàries que han accelerat els temps de cerca.

Faiss presenta la construcció d'índexs simples, on la cerca es basa en la comparació exhaustiva de tots els vectors en la consulta. Però també presenta noves tècniques per ala creació d'índexs.

Tecnologies

- **Partició de l'índex**: Aquesta tècnica consisteix a dividir l'índex en cel·les de Voronoi. Per comprendre els diagrames de Voronoi imaginarem els nostres vectors en un món 2D, en aquest món col·locarem alguns punts que es convertiran en el centre de les cel·les. A partir d'aquests punts expandirem un radi igual, on arribat algun moment xocaran entre si creant els límits de la cel·la:

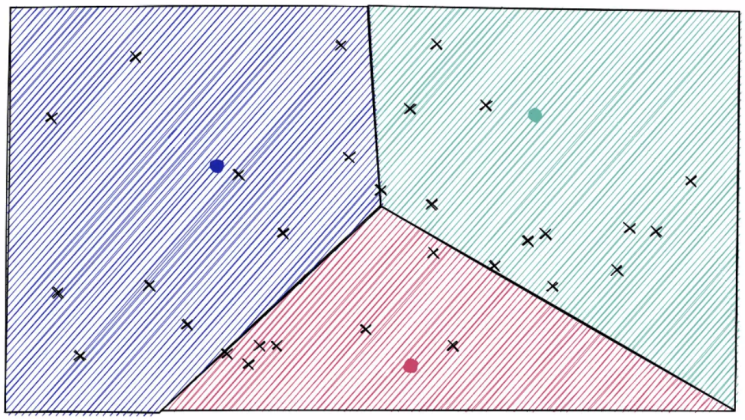


Figura 4.8: Exemple de món 2D de Voronoi

  Una vegada tenim aquest món creat, en el moment de realitzar una consulta d'un vector, no cercarem en tots els espais, sinó, en la cel·la corresponent que li pertoque. Si el vector de consulta està en el límit o molt proper, es pot corregir augmentant el rang de cerca, en compte, d'una cel·la, dues o tres.

- **Locality Sensitive Hashing (LSH)**: LSH és un tècnica que funciona agrupant vectors en cubs mitjançant el processament de cada vector a través d'una funció hash, aquesta funció té com principal característica que maximitza les col·lisions. Una col·lisió és quan en un diccionari dues entrades produeixen el mateix valor hash (codificació de la clau de l'entrada).

  En aquesta tècnica volem maximitzar-les per agrupar els vectors similars en un mateix cub. És a dir, els vectors similars crearan una codificació de la clau, valor hash, igual per agrupar-se tots en un mateix grup. Així en el procés de cera només buscar en el grup més similar al vector consulta.

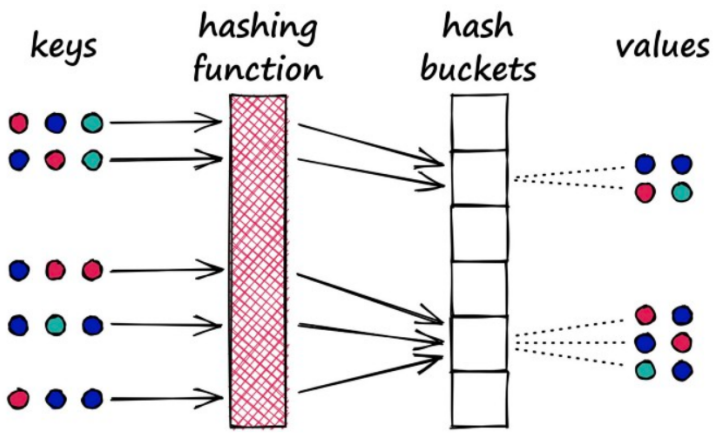


Figura 4.9: Exemple de la tècnica LSH

- **Gràfics Hierarchical Navigable Small Word (HNSW)**: La tècnica HNSW és una adaptació addicional del gràfics Navigable Small Word (NSW), on el gràfic NSW és una estructura que conté vèrtexs connectats pels límits als veïns més propers. NSW significa que els vèrtexs dins del gràfic tenen una longitud de ruta mitjana molt curta a tots els altres vèrtexs, a pesar de no estar connectats directament.

  Un exemple són els usuaris de Facebook, utilitzant NSW connectarem a cada usuari amb els seus amics més propers, i a pesar de tindre més de 1500 milions d'usuaris actius, la quantitat mitjana de salts necessària per a creuar el gràfic era de 3,57.

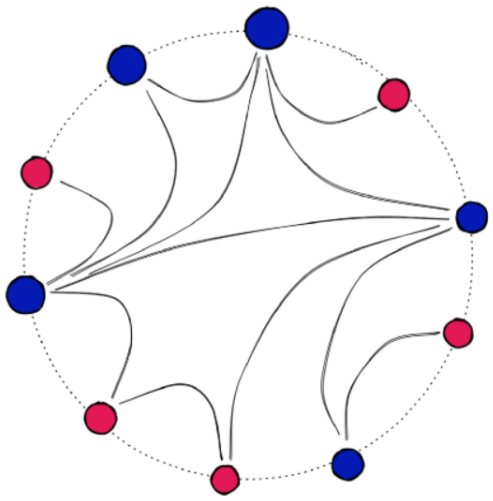


Figura 4.10: Exemple de gràfic NSW, cada usuari està connectat a tots en màxim 4 passos

Aleshores el gràfics HNSW es construeixen dividint els gràfics NSW en diferents capes. Cada capa elimina connexions intermèdies entre vèrtexs. En la següent figura podem veure com el gràfic anterior s'ha dividit en capes i transformat en un gràfic HNSW.

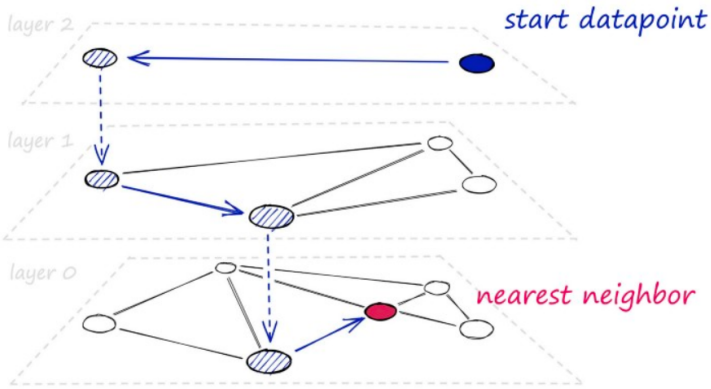


Figura 4.11: Exemple de gràfic HNSW, on dividim el gràfic en capes que recorrem durant la cerca

- **Quantificació**: La quantificació és un mètode de compressions de dades en un espai més xicotet, si tenim un vectors de 128 dimensions (D) amb valors que són floats de 32 bits en el rang de 0.0 –> 1570.0 (S), en compte de comprimir la dimensionalitat, de reduir-la, el que reduïm és S, el rang de valors.

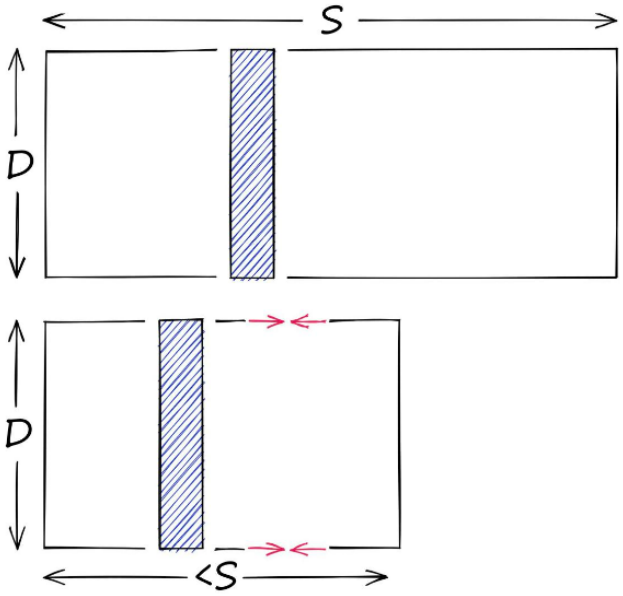


Figura 4.12: Exemple de la quantificació

Açò es produeix per mitjà de l'agrupació, on agrupem un conjunt de vectors, reemplacem l'alcans més gran de valors potencials, amb un conjunt discret i simbòlic més xicotet de centroides. Aleshores la quantificació seria la transformació d'un vector en un espai amb nombre finit de valors possibles, on eixos valors són representacions simbòliques del vector original.

És a dir, nosaltres partim d'un vector, el dividim en m (ha de ser divisor de D) subvectors i per cada subvector el processem per un algoritme de clustering específic per al seu subespai, aquest procés de clúster crea un conjunt de centroides per cada un. Una vegada està processat el subvector, li assignem un centroide més proper del seu subespai. Cada centroide té un Id, que és el que utilitzem per emmagatzemar, i es pot rastrejar a posteriori per recuperar-los. Amb aquesta tècnica podem reduir la memòria utilitzada en un total de 64 vegades.

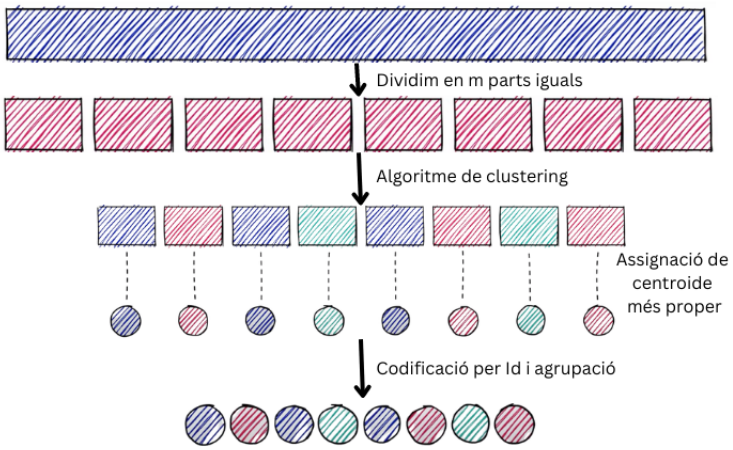


Figura 4.13: Exemple de la tècnica de Quantificació

Índexs

Amb aquestes tecnologies els índex que podem crear són els següents:

- **Flat**: Són índexs plans, és a dir, no modifiquen els vectors que indexem en ells. Donat que no hi ha aproximacions ni agrupacions de les representacions vectorials, aquest índex produeixen els resultats més precisos. Però, en canvi, d'aquesta alta qualitat de cerca tenim uns costos temporals alts.

  Aquest índex es caracteritzen per fer una cerca exhaustiva, és a dir, agafem una representació vectorial de la consulta i la comparem en totes les altres indexades, calculant la distància entre si.

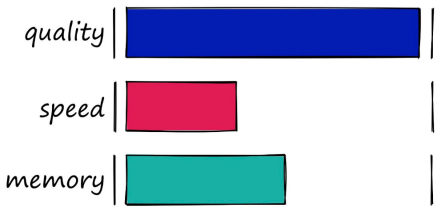


Figura 4.14: Característiques dels índexs Flat

  Com hem dit en fer una comparació exhaustiva i no comprimir els vectors obtenim un resultat altament satisfactori, però donat aquest fet també tenim un alt ús de memòria, ja que els vectors s'indexen sencers, i de temps, ja que comparem els vectors consulta en tots els possible vectors.

- **índex LSH**: Són aquells índex que fan ús de la tècnica de LSH 4.3.3.7 per indexar les representacions vectorials. Són índexs molt variables i dependents dels paràmetres que utilitzes

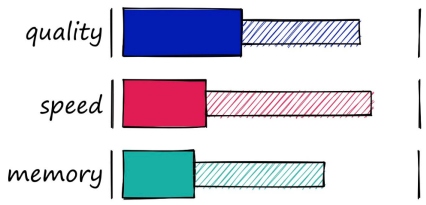


Figura 4.15: Característiques dels índexs LSH

En aquest figura podem observar com aquest tipus d'índex depén molts dels paràmetres seleccionats, els segments pintats a ratlles simbolitzen el rang de rendiment. Una bona qualitat de cerca dona com a resultat una cerca més lenta, en canvi, una cerca ràpida dona com a resultat una qualitat pitjor de cerca.

Es caracteritza per un baix rendiment per a vectors amb alta dimensionalitat, donat que és un índex molt dependent de la dimensionalitat del vector original, i del nombre de bits que requereixes per codificar el valor hash, un valor més alt obtens una major precisió a costa de memòria i rapidesa.

- **índex HNSW**: Són índex que utilitzen la tècnica de HNSW 4.3.3.7 per poder tractar totes les representacions vectorials.

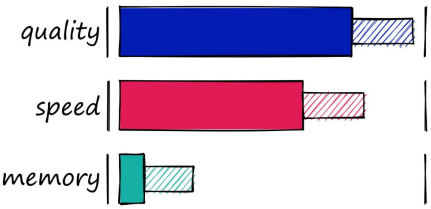


Figura 4.16: Característiques dels índexs HNSW

Aquest índex brinda una qualitat de cerca molt alta amb un tens de cerca increïble, no obstant, estem parlant d'un ús de memòria molt elevat. És el tipus d'índex que més memòria utilitza en tot Faiss.

- **Índex d'arxius Invertits**: Aquest tipus d'índex utilitzen la tècnica de partició de l'índex 4.3.3.7 per indexar els vectors. És uns dels índex més utilitzat a causa de la seua simplicitat i facilitat d'ús.

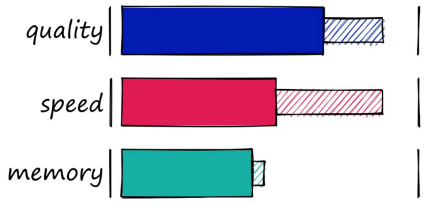


Figura 4.17: Característiques dels índexs d'arxius invertits

Caracteritzats per una alta qualitat de cerca, bona velocitat i un ús de la memòria acceptable. Els paràmetres que la modifiquen són el nombre de cel·les a crear i el nombre de cel·les a cerca, fent que siga més ràpid o més lent, i variant la qualitat de la cerca.

- **Índex Compostos**: Els índexs compostos es poden veure com un procés pas a pas de transformacions vectorials i un o més tècniques d'indexació. Podent crear el nostre índex ideal com a combinació d'altres. Per exemple podent crear un índex d'arxius invertits amb una quantificació posterior.

  Els índex compostos estan compostos per blocs que encaixen entre ells, aquest blocs poden ser els següents:

  - Transformacions vectorials: mètodes de preprocessament dels vectors com PCA o OPQ
  - Quantificador aproximat: Organització del vectors en subdominis, IVF, HNSW.
  - Quantificador fi: Comprensió més fina dels vectors en dominis xicotets, PQ.
  - Refinament: Pas final, en la cerca, que reordena els resultats utilitzant el càlcul de distàncies en els vectors originals

  Aquest concepte ens permet crear índex superpotents que agafen el millor de cada món.

Hem utilitzat aquesta llibreria, en combinació amb haystack, per crear el nostre índex personalitzat al nostre projecte.

CAPÍTOL 5
Desenvolupament de la solució proposada

Aquest capítol consistirà en una explicació del procés de desenvolupament del projecte, més concretament, quins problemes han anat sortint, quines decisions hem hagut de prendre, etc. L'estructura del capítol serà la següent: primer explicarem quins problemes hem tingut al llarg del projecte, posteriorment quines decisions hem pres, i, finalment parlarem de la implementació.

5.1 Problemes i dificultats

En aquest apartat anirem desglossant quins han sigut els principals problemes que han succeït al llarg del projecte.

5.1.1. Gestió de la memòria de les GPU

La màquina Tardis 4.3.1.4 compta amb dues GPU NVIDIA RTX 3090 les quals compten amb 24GB de VRAM. Depenent dels models de llenguatge que triem, aquesta quantitat de memòria requerida és insuficient i hauríem de buscar solucions al problema.

Les solucions que s'han contemplat en aquest treball són les següents:

1. **Reduir el Batch Size**: El *batch size* s'utilitza per indicar el nombre de mostres s'utilitzaran en paral·lel, els resultats d'aquestes mostres s'utilitzen 'per recalcular els pesos de la xarxa. Aquestes mostres s'emmagatzemen totes a la vegada en la VRAM, per tant, com més gran és el *batch* més VRAM es requerirà. Aleshores, per resoldre el problema utilitzarem un *batch size* més xicotet per utilitzar menys la VRAM.
2. **Utilitzar versions reduïdes dels Models**: Normalment, la majoria de les arquitectures de models de llenguatge solen comptar amb diferents versions del mateix model. Totes del versions compten amb la mateixa arquitectura, però, tenen més o menys paràmetres. Com menys paràmetres, el model ocuparà menys memòria. En triar una versió més reduïda d'un model de llenguatge, ens estem beneficiant de les virtuts proporcionades per l'arquitectura del model, encara que el model puga recordar menys detalls de les dades en les quals s'entrena. Depenent de la tasca i la quantitat de dades, una versió reduïda pot ser suficient i, en alguns casos, inclús beneficiós.

En aquest treball hem prioritzat utilitzar models més grans. Per tant, sempre hem tractat de solucionar el problema de memòria amb la reducció del tamany del *batch* abans

de canviar de model. Gestionant correctament el tamany del *batch* hem pogut utilitzar tots els models que ens havíem plantejat, si bé el temps d'entrenament s'ha incrementat, donat que es processen menys mostres en paral·lel.

5.1.2. Conversió de Jsonl a Document

El corpus que hem utilitzat està emmagatzemat en un format en el qual cada línia del fitxer correspon a un JSON; format que es coneix com a JSONLines. Haystack proporciona una classe denominada JSONConverter, però aquesta classe té com entrada una organització especifica del jsonl. La nostra forma d'organitzar els jsonl no era igual que la requerida. Per solucionar aquest problema, ens vam plantejar dues solucions:

1. **Adapatar el nostre arxiu**: En primer lloc, vàrem pensar a convertir el nostre arxiu JSONLines al format que requeria Haystack.
2. **Reescriure el codi**: La segona solució que vàrem plantejar era agafar el mètode de conversió de la classe i reescriure'l per adaptar-lo al nostre format. Aquesta solució implicava anar a la wiki d'Haystack [52] i trobar el codi obert que buscàvem. Una vegada trobat, transformar-ho per adaptar-se al nostre format.

Es va decidir aplicar la segona solució, ja que ens pareixia una sol.lució més neta, de forma que el nostre codi poguès acceptar JSONLines sense necessitat de fer un preprocès previ i així poder utilitzar el corpus amb el format original.

5.1.3. Memòria en Disc

Aquest problema de memòria està relacionat en la creació de l'índex de Faiss. Els índex Faiss estan formats per dos fitxers físics en disc, una base de dades de SQL i un índex per a les representacions vectorials dels documents. Per defecte, Haystack, utilitza una base de dades SQLite, la qual és una bona primera opció, ja que no es necessita cap mena de servei addicional. No obstant, no és una bona opció quan el nombre de dades a emmagatzemar augmenta notablement; com és el nostre cas en el qual tenim 3M de documents.

Haystack recomana PostgreSQL per a situacions amb tantes dades. No obstant, utilitzar PostgreSQL implicaria muntar i configurar un servei addicional, la qual cosa queda fora de l'abast del nostre treball. Com a solució pal·liativa, optarem per reduir el corpus de castellà de 2,2 M a 1,5 M de mostres.

5.1.4. Problemes amb l'Entrenament

Una de les parts més important i crítiques de tot el projecte és l'entrenament del model. Des del principi ens va generar dificultats per adaptar els scripts i personalitzar-ho amb les nostres dades. Però el principal problema va vindre que després d'efectuar-se l'entrenament el model d'eixida no va guardar-se en el disc i no es podia utilitzar de cap de les formes. En un pròxim apartat, explicarem tot l'entrenament en profunditat, quin va ser el problema i quina decisió vàrem prendre.

5.2 Decisions Importants

En aquest apartat farem un repàs de les decisions més important que hem pres al llarg del projecte. Explicarem quines han sigut i donarem les raons.

5.2.1. Elecció del Model

Pel treball hem seleccionat un conjunt de models que ens serviren per al nostre propòsit; extraure *embeddings* contextuals que serviren per a mesurar distàncies semàntiques. Els models seleccionats havien de tindre les següents característiques:

- Ser models punters
- Model entrenats per a la similitud semàntica
- Models multilingües
- Models emmagatzemats en el Hub de Hugging Face.

. Una vegada obtingut el conjunt de models de llenguatge es va fer una experimentació per triar el model final per al nostre SRI, experimentació que es detallarà més endavant.

5.2.2. Elecció de DocumentStore

L'explicació de cada tipus de DocumentStore la trobem en el capítol anterior 4.3.3.6. L'objectius dels *DocumentStores* és el mateix emmagatzemar els documents i realitzar les comparacions de similitud. No obstant, cada *DocumentStore* presenta una implementació diferent amb distintes funcionalitats.

La nostra motivació en aquesta elecció parteix de solucionar i agregar noves prestacions als anteriors SRI que hem creat, partint dels següents punts:

- **Indexar tots els documents**: Ser capaços d'indexar tota la col·lecció de documents. És a dir, solucionar els errors de memòria que vàrem tindre en l'anterior projecte i obtenir un índex eficient en la recuperació d'informació.
- **Escalabilitat de l'índex**: Volem un índex capaç d'escalar en nous documents que agreguem. Per tant l'índex que seleccionem ha de ser capaç d'agregar nous documents sense que els documents anteriorment indexats afegeixen un cost adicional.
- **Eficiència en la cerca**: Volem un índex capaç de retornar els resultats cercats en un temps reduït, donat que, els sistema ha de ser ràpid a l'hora de fer la cerca.

La Taula 5.1 mostra un resum dels diferents DocumentStore i les seues característiques:

DocumentStore	Principals Característiques	Integrat?
ElasticSearch	Recuperació dispersa amb moltes possibilitats d'ajust suport bàsic en la recuperació densa	Si
InMemory	Emmagatzematge senzill, sense serveis extra ni dependències	Si
Milvus	Open Source Recuperació densa per a cerca de similitud escalable	No
OpenSearch	Open Source Compatible amb Amazon Característiques igual que ElasticSearch, suport a comparacions entre vectors	Si
PineCone	Servei administrat per a la recuperació densa a gran escala. Filtra metadades Latència baixa en consultes a qualsevol escala Actualitzacions d'índex en viu	Si
Qdrant	Open Source, Suport de filtratge extens	No
SQL	Simple i ràpid, sense requisits de base de dades. Admet MySQL, PostgreSQL i SQLite	Si
Weaviate	Open Source, recuperació densa simple, Emmagatzematge de documents, metadades i vectors junt Combinació de cerca vectorial i filtrar	Si
Faiss	Open Source Recuperació densa a través de diferents tipus d'índex Cerca ràpida i eficient Escalabilitat	Si

Taula 5.1: Taula Resum del DocmuentStore. https://docs.haystack.deepset.ai/docs/document_store

Donada la Taula 5.1 amb les distintes tecnologies d'indexació implementades per Haystack i les seues característiques, ens vam decantar per Faiss. Els motius a destacar són els següents: conté mètodes eficients per la cerca de documents basada en la similitud entre vectors densos, es pot executar tant en CPU com en GPU i és escalable en entorns multi-GPU. A més a més, en l'actualitat, Haystack recomana l'ús de Faiss, per la qual cosa hi ha disponibles gran quantitat de documentació i tutorials.

5.2.3. Elecció de l'Índex

A conseqüència de l'ús de Faiss, hem d'elegir quin tipus d'índex volem utilitzar en aquest projecte. Una vegada hem estudiat tots els índexs que podem crear i quines tècniques podem utilitzar 4.3.3.7, ens hem guiat pels següents criteris per utilitzar un índex o altre:

- **Nombre d'usos**: Si realitzarem un nombre de cerques menor que 10000, el temps de creació de l'índex no es compara amb el temps de cerca, per tant, el càlcul directe és la millor opció. Si vol ser utilitzat moltes voltes parlem d'utilitzar índex que afavorisquen a la velocitat de cerca.

- **Resultats exactes**: Si volem resultats exactes, és a dir, resultat amb la millor qualitat possible hem d'utilitzar el "Flat". Donat que és l'únic índex que garanteix resultats exactes.

- **Memòria**: Depenent de la importància de la memòria hi ha diferents opcions:
  - Res important: Utilitzar HNSW, ja que és un índex amb unes prestacions molt bones en qualitat i rapidesa, però requereix molta memòria.
  - Un poc important: Utilitzar un índex compost amb alguna tècnica de quantificació aproximada i connectat amb un índex Flat.
  - Prou important: Utilitzar un preprocessament dels vectors per reduir la dimensionalitat, qualsevol tècnica d'agrupació i per últim utilitzar PQ, tècnica explicada en la Secció 4.3.3.7, per una compressió més fina dels mateixos vectors, utilitzant 4 bits.
  - Molt important: Igual que en l'apartat anterior, però aquesta vegada utilitzant en el PQ un valor de codificació (M) menor que 64.
- **Grandària del Corpus**: Depenent de la grandària del corpus podem trobar diferents opcions:
  - Menor que 1M de vectors: Podem utilitzar un índex d'arxiu invertit (IVF), utilitzant un nombre de cel·les igual a K. On K és un valor que anirà entre $4 \cdot \sqrt{N}$ a $16 \cdot \sqrt{N}$, on N és el nombre de vectors.
  - Entre 1M i 10M: Utilitzar un IVF65536 amb un HNSW32.
  - Entre 10M i 100M: Utilitzar un IVF262144 amb un HNSW32.
  - Entre 100M i 1000M: Utilitzar IVF1048576 amb un HNSW32.

En el nostre projecte utilitzarem dos tipus d'índex:

- **Flat**: utilitzem un índex Flat per comprovar la seua precisió i qualitat de resposta. Donat que és un índex ràpid i molt fiable. Però té problemes amb la memòria.
- **IVF amb PQ**: Donat que estem entre 1M i 10M de documents totals, hem decidit utilitzar aquest índex perquè obtenim una relació de qualitat-temps bona, donat que no seran els millors resultats possibles, però sí que obtindrem solucions en un temps baix i amb un ús de memòria inferior.

En apartats posteriors farem una comparació entre els dos índexs per veure quins és el que utilitzem en el nostre sistema final.

5.2.4. Elecció del Model del Reader

Aquest model s'utilitzarà per efectuar la part del Reader. El *Reader* és un element que llegeix en més profunditat els documents recuperats pel *Retreaver*, i contesta a la consulta amb la informació rellevant. En el nostre cas, després d'investigar pel Hub de Hugging Face, hem decidit utilitzar el següent model multilingüe: **timpal0l/mdeberta-v3-base-squad2**

Aquest model utilitza la tecnologia DeBERTa, que millora els models BERT i RoBERTA usant atenció desenredada i descodificació de màscara millorada. Amb aquestes millores, DeBERTa millora a RoBERTa en moltes tasques PLN amb 80 GB de dades d'entrenament.

En aquest cas usen mDeBERTa, la versió multilingüe del DeBERTa, entrenada amb grans col·leccions de corpus de diferents idiomes, en total 94.

5.3 Implementació dels Diferents Sistemes del Projecte

En aquest apartat farem un repàs a la implementació de la solució final. Implementació portada a terme gràcies a Python i Visual Studio Code.

5.3.1. Sistema Creador del Model

Aquesta implementació no difereix de l'explicada en la secció del Disseny Detallat 4.2.1. En resum, utilitzem scripts modificats de Beir, que ens permeten analitzar i avaluar els diferents models. Posteriorment, ens permeten entrenar el nostre model. Les diferències respecte a l'script original són les següents:

- Canviar la forma de carregar el corpus, en compte d'utilitzar un corpus que està en el núvol i és propietat de Beir. Carreguem un corpus propi des de memòria que després pot ser tractat amb els mateixos mètodes.
- Utilitzem un model preentrenat de Hugging Face, per tant, en compte de crear un, modifiquem el codi per carregar-ho des de Hugging Face.
- Guardem el model entrenat en un disc propi, i no en el núvol com es feia originalment.

5.3.2. Implementació SRI

Per aquest sistema hem creat les següents classes:

- **Classe Indexador**: Encarregada de dur a terme tot els procés d'indexació. Està constituïda pels següents mètodes:
  - **__init__(self, opció, path_index)**: Mètode constructor on utilitzem dos paràmetres: ´´opció´´ indica quin tipus d'opció vols realitzar o crear un documentStore nou o carregar un des de memòria i "path_index" per si volem carregar-ho des de memòria fa referència al *path* del disc. En aquest mètode inicialitzem l'objecte document_store.
  - **indexar(self, model, path_doc):** Mètode que s'encarrega d'indexar els nous documents que estan guardats en la direcció path_doc, aquest mètode en particular crida dos mètodes diferents, escribir_docs() i trasformar_emb()
  - **escribir_docs(self, path_doc)**: Mètode encarregat de llegir els arxius jsonl on estan guardats tota la informació a indexar, i transformar-la en la configuració necessària per a transformar-la en Documents. També és l'encarregat de crear el *Preprocessor* i processar els documents. Una vegada tenim tots els documents en memòria i netejats passem a escriure'ls en la base de dades.
  - **transformar_emb(self, model)**: Mètode on creem del component del *Retriever* configurat amb el model desitjat i el document_store creat, per poder entrenar l'índex, si és necessari, i a posteriori crear les representacions vectorials del documents i indexar-les.
- **Classe Recuperador**: La classe recuperador s'encarrega de connectar-se amb l'usuari i processar la consulta realitzada, retornant un rànquing del documents més rellevants. Aquesta classe està construïda de la següent forma:

- **__init__(self, path_docStore, model, model_reader)**: Mètode constructor que s'encarrega d'inicialitzar tots els elements necessaris per a la recuperació d'informació. Aquest elements són:
  * Document_store = base de dades de tots els documents indexats i el mateix índex. Crida a cargar_docu()
  * Retriever = element que retorna 10 documents més rellevants de la base de dades. Crida a definir_retriever()
  * Reader = element que agafa eixos 10 documents i els revisa en més profunditat per crear el rànquing. Crida a definir_reader()
  * Pipeline = Pipeline de tipus ExtarctiveQA, per a respondre preguntes, que connecta el reader amb el retriever. Crida a definir_pipeline()

  Ell paràmetres dels constructor són els següents: ´´path_docStore´´ que fa referència al *path* l'arxiu on està guardat l'índex de faiss. ´´Model´´ paràmetre que representa el nom del model a utilitzar en el *Retriever*. "model_reader´´ fa referència al nom del model utilitzat en el *Reader*.
- **cargar_docu(self, path)**: Mètode que carrega des de memòria l'índex anteriorment creat, i el guarda en un objecte documentStore.
- **definir_retriever(self, model)**: Crea l'objecte *Retriever* definit amb el model seleccionat i el documentStore carregat.
- **definir_reader(self, model)**: Crea l'objecte *Reader* amb el model seleccionat.
- **definir_pipeline(self)**: Crea l'obejcte *Pipeline* que s'encarrega de connectar la consulta amb el *reader* i *retriever*.
- **recuperar(self, consulta, N_Retriever, N_Reader)**: Mètode encarregat de posar en funcionament la recuperació d'informació. Per mitjà de la pipeline, amb el mètode run(), fem que la consulta siga representada vectorialment i processada pel retriever que tornara N_Retriever documents, aquest documents es tornaran a processar pel reader retornant un rànquing de N_Reader documents. Finalment, aquest rànquing és la resposta oferida a l'usuari.

CAPÍTOL 6
Experimentació

En aquest capítol ens centrarem en les diferents experimentacions portades a terme abans de construir el sistema final. Aquestes experimentacions s'han realitzat tant per a construir el model que s'utilitzarà en el SRI com per a comprovar l'estructura del SRI a una petita escala.

6.1 Construcció del corpus d'Avaluació

A les Seccions 4.2.1.1 y 4.2.1.2 hi ha una explicació detallada de la creació del corpus. En les citades Seccions s'explica com està configurat el corpus, i quins han estat els passos necessaris per a la seua creació. En aquest apartat en centrarem a mostrar quins ha sigut el temps necessari per a construir, traduir i netejar aquest corpus.

Corpus	Temps de Traducció	Temps de Neteja	Temps Total
Corpus de documents	17 dies 4 h 31 min 15 s	7 min 44 s	17 dies 4 h 38 min 59 s
Corpus de consultes	11 h 25 min 35 s	1 min 12 s	11 h 37 min 47 s

Taula 6.1: Temps de preprocessament dels diferents corpus

A la Taula 6.1 podem observar els temps emprats per traducció i neteja dels documents i de les consultes. Es pot notar que pràcticament la totalitat del temps ha estat utilitzada per la part de traducció. El procés de traducció de les frases han tingut un cost temporal bastant notable pel fet que s'han emprat models de traducció basats en xarxes neuronals.

6.2 Avaluació i Comparació dels Models

En aquest apartat explicarem l'experimentació portada a terme per tal d'avaluar i elegir el model. Aquesta experimentació s'ha realitzat utilitzant el script *evaluate_sbert.py* 4.2.1.3, on avaluem els diferents models utilitzant diferents mètriques. S'han trigat més de 17 dies en processar 3,6GB de text del corpus d'informació i 11 h per processar les 65MB del corpus de consultes.

6.2.1. Models Utilitzats

A partir del Hub de HuggingFace, hem triat una sèrie de models que s'utilitzen per a tasques de cerca o similitud semàntica. Els models són els següents:

1. **distiluse-base-multilingual-cased-v2** [14]: Aquest model utilitza representacions vectorials denses de 512 dimensions. Model creat per Reimers, Nils, Gurevych i Iryna l'any 2019.

2. **hiiamsid/sentence_similarity_spanish_es** [53]: Aquest és un model *sentence-transformer*, com l'anterior, on assigna oracions i paràgrafs a un espai vectorial dens de 768 dimensions. Aquest models s'utilitza principalment per a la tasca de similitud semàntica. Com a característica principal té el fet que ha sigut entrenat especialment per a l'idioma en castellà. Aquest model agafa com a base el model BETO (dccuchile/bert-base-spanish-wwm-cased) [53], un model BERT però entrenat amb un gran corpus en castellà.

3. **sentence-transformers/all-mpnet-base-v2**: És un model que extrau un embeddings de 768 dimensions. Aquest model té la particularitat de ser un model MPNet 2.3.5.5 [17], que agafa com a base el model microsoft/mpnet-base i fa el *Fine-Tuning* amb un corpus de 1B de parells d'oracions.

4. **sentence-transformers/multi-qa-mpnet-base-dot-v1**: És un model basat en la tecnologia *mpnet* [17] que extrau un embeddings de 768 dimensions. Model entrenat amb 215 milions de parells de frases (pregunta, resposta). Aquest model s'utilitza per codificar consultes i paràgrafs de text, per trobar aquells més rellevants a la consulta. Utilitza el model *mpnet-base* preentrenat com a base.

5. **symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli**: És un model que extrau un embeddings de 768 dimensions. La base del model són els models XLNet [16], més concretament el model prenetrenat *xlm-roberta-base*, entrenat amb diferents corpus.

6. **sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2** [14]: És un model que extrau un embeddings de 384 dimensions. És el model més xicotet de tots, ja que sols compta amb X paràmetres a la xarxa.

Tots aquest models han sigut entrenats per crear representacions vectorials de paraules, frases o paràgrafs. A més, les representacions estan creades en un espai vectorial adequat per a fer la similitud semàntica, és a dir, el càlcul de les distàncies.

La Taula 6.2 mostra els noms de referència per als models, aquests nous noms els utilitzarem en futurs apartats per fer referència als models.

Nom de Referència	Nom Real
Model 1	distiluse-base-multilingual-cased-v2
Model 2	hiiamsid/sentence_similarity_spanish_es
Model 3	sentence-transformers/all-mpnet-base-v2
Model 4	sentence-transformers/multi-qa-mpnet-base-dot-v1
Model 5	symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli:
Model 6	sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

Taula 6.2: Noms de referència dels diferents models

La Taula 6.3 mostra les diferents grandàries dels models i de les representacions vectorials que creen.

Nom de Referència	Grandària Vectors	Grandària Models
Model 1	512	480 MB
Model 2	768	420 MB
Model 3	768	420 MB
Model 4	768	420 MB
Model 5	768	1,1 GB
Model 6	384	420 MB

Taula 6.3: Grandàries de vectors i models

6.2.2. Mètriques Utilitzades

Una vegada hem explicat quins models avaluarem, en aquest apartat explicarem quines seran les mètriques que utilitzarem per fer l'avaluació i comparació entre els diferents models seleccionats.

- **Normalize Discounted Cumulative Gain**: El Guany Acumulat Descomptat Normalitzat és una mesura de qualitat del rànquing, explicada en profunditat en apartats anteriors 2.4.2.9
- **Mean average precision**: Mitjana de la Precisió Mitjana explicada en l'apartat 2.4.2.8, és una mesura que relaciona la precisió amb el recall en un únic valor.
- **Recall**: El recall o exhaustivitat mesura la quantitat documents rellevants recuperats pel model sobre el total de documents rellevants 2.4.2.2.
- **Precisió**: La precisió mesura el percentatge de documents rellevants sobre el total de documents recuperats pel model 2.4.2.1
- **MRR**: el *Mean Reciprocal Rank* (MRR) és una mesura de la classificació del primer element rellevant en una llista classificada. Per exemple, si el primer document rellevant en una llista ordenada apareix el 3, tindria una puntuació de $\frac{1}{3}$.
- **Recall cap**: El mateix que el recall, explicat en la Secció 2.4.2.2, però limitat a un cert nombre de documents.
- **Hole**: Hole mesura el percentatge de top-k documents rellevants no vist pels experts o anotadors.

Amb aquestes mesures serem capaços d'avaluar el rendiment dels models amb la tasca de la recuperació d'informació i seleccionar aquell model que obtinga el major rendiment per utilitzar-lo en el nostre SRI.

6.2.3. Resultats Avaluació

Al llarg de l'experimentació s'ha anat variant el nombre de documents recuperats (k). Si $k = 10$ recuperem un rànquing de 10 documents ordenats. Els valors que s'han utilitzat i recomana Beir [26] són [1, 3, 5, 10, 100, 1000].

6.2.3.1. NDCG

En la primera mesura de totes hem obtingut els següents resultats:

Models/K	1	3	5	10	100	1000
Model_1	0.02636	0.04251	0.04854	0.05731	0.08919	0.1183
Model_2	0.01289	0.02111	0.02452	0.02969	0.05042	0.07225
Model_3	0.04298	0.06596	0.07621	0.08839	0.11881	0.1418
Model_4	**0.07221**	**0.1055**	**0.12077**	**0.1377**	**0.17656**	**0.2021**
Model_5	0.01891	0.02994	0.03428	0.04186	0.06215	0.08236
Model_6	0.02693	0.0438	0.05011	0.06092	0.09646	0.12329

Taula 6.4: Taula Resultats de la mesura NDCG

Com podem comprovar en la taula 2.4.2.9, els millors resultats han sigut obtinguts pel model_4, que és el model sentence-transformers/multi-qa-mpnet-base-dot-v1. En el corpus que utilitzem per avaluar els models sols hi ha una resposta correcta per a una consulta. La mesura NDCG mesura la qualitat del rànquing tenint en compte el rànquing ideal, per tant, tenint en compte que sols hi ha un document rellevant és complicat per al models extraure resultats bons.

6.2.3.2. Map

Per a la mesura de Map hem obtingut:

Models/K	1	3	5	10	100	1000
Model_1	0.0274	0.03816	0.04149	0.04501	0.05056	0.05148
Model_2	0.01261	0.01888	0.02074	0.02281	0.02626	0.02692
Model_3	0.04169	0.05942	0.06508	0.07007	0.07558	0.07632
Model_4	**0.0704**	**0.0965**	**0.1045**	**0.11153**	**0.11866**	**0.11941**
Model_5	0.01855	0.02682	0.02921	0.0323	0.03583	0.03646
Model_6	0.02696	0.03871	0.04245	0.0469	0.05325	0.05409

Taula 6.5: Taula Resultats de la mesura Map

Cas paregut a la mesura anterior, el millor model que trobem és el 4. Exposar que el corpus està creat d'una forma on sols hi ha un document rellevant per consulta, per tant, és complicat obtindre bons resultats.

6.2.3.3. Exhaustivitat/Recall

En la següent mesura, l'exhaustivitat, hem obtingut els següents valors:

Models/K	1	3	5	10	100	1000
Model_1	0.02574	0.0542	0.06879	0.09608	0.25475	0.49149
Model_2	0.01261	0.02705	0.03529	0.05141	0.1561	0.3378
Model_3	0.04169	0.08343	0.1082	0.14546	0.29438	0.51657
Model_4	**0.07044**	**0.13009**	**0.16691**	**0.2198**	**0.4086**	**0.58956**
Model_5	0.01855	0.03811	0.04854	0.07173	0.17316	0.3382
Model_6	0.02629	0.0558	0.07235	0.10654	0.28078	0.4991

Taula 6.6: Taula Resultat de la mesura Exhaustivitat

Els resultats mostren com el model_4 és el model amb millors valors, demostrant com per a la recuperació de prou documents retorna un decent 41% de Recall en els 100 pri-

mers. No obstant, com ja hem explicat en al Secció 2.4.2.2, a mesura que recuperem més documents és més probable recuperar els document rellevants i obtindre uns resultats millors. donat que el recall no mesura els documents totals recuperats, sols mesura els documents rellevants recuperats.

6.2.3.4. Precisió

Per a la precisió els valors són els següents:

Models/K	1	3	5	10	100	1000
Model_1	0.02636	0.01853	0.01413	0.00989	0.00267	0.00052
Model_2	0.01289	0.00931	0.00731	0.00532	0.00164	0.00036
Model_3	0.04298	0.02875	0.02241	0.01511	0.00309	0.00051
Model_4	**0.07221**	**0.04494**	**0.03467**	**0.02275**	**0.00429**	**0.00062**
Model_5	0.01891	0.01328	0.01011	0.00754	0.00182	0.00036
Model_6	0.02693	0.01915	0.0149	0.01087	0.00293	0.00053

Taula 6.7: Taula Resultats de la mesura de la Precisió

La precisió ens mostra com el model més precís és el model_4, però al contrari del recall la precisió disminueix amb els documents totals recuperats. Com hem observat, la precisió sí que té en compte el total de documents recuperats, per això a major nombre de documents pitjor és el resultat obtingut. A més, parlem d'un corpus on sols hi ha una mostra positiva per consulta, per tant, és complicat extraure resultats alts en aquest apartat. Donat que com sols hi ha 1 document rellevant a mesura que recuperem més sols creix el número del denominador.

6.2.3.5. MRR

Els resultats de l'avaluació del MRR són els següents:

Models/K	1	3	5	10	100	1000
Model_1	0.02636	0.03904	0.04251	0.04613	0.09608	0.05266
Model_2	0.01275	0.01932	0.02127	0.02341	0.02697	0.02763
Model_3	0.04298	0.06115	0.06688	0.072	0.07757	0.0783
Model_4	**0.07235**	**0.09885**	**0.1073**	**0.11442**	**0.12155**	**0.12227**
Model_5	0.01891	0.02763	0.03011	0.03333	0.03687	0.0375
Model_6	0.02679	0.0394	0.04327	0.0478	0.05425	0.05508

Taula 6.8: Taula Resultats de la mesura MRR

Aquest taula ens mostra com és el model_4 qui ordena millor el primer element ens els seus rànquings. Aquesta mesura és prou important, pel fet de com sols hi ha un document rellevant és important en quina posició del rànquing el posicione. No obstant, és complicat extraure bons resultats en aquesta mesura, ja que posicionar el document rellevant en primer lloc és una tasca difícil.

6.2.3.6. Recall Cap

En la següent taula mostren els resultats de la mesura Recall Cap:

Models/K	1	3	5	10	100	1000
Model_1	0.02636	0.05505	0.06978	0.09706	0.2621	0.4923
Model_2	0.01275	0.02750	0.03612	0.05187	0.1645	0.3423
Model_3	0.04298	0.0912	0.1102	0.1498	0.29781	0.48123
Model_4	**0.07235**	**0.13121**	**0.16698**	**0.22098**	**0.41321**	**0.59123**
Model_5	0.01977	0.03811	0.05010	0.07564	0.18213	0.33912
Model_6	0.02679	0.05655	0.07321	0.10876	0.29871	0.5001

Taula 6.9: Taula Resultats de la mesura Recall Cap

Taula molt similar a la de la mesura Recall 6.6, en la que el millor model que observem és el model_4. En ser tan pareguda l'explicació dels resultats és igual, a més documents recuperats més probabilitat de recuperar el documents rellevant.

6.2.3.7. Hole

Els resultats de l'última mesura Hole són els següents:

Models/K	1	3	5	10	100	1000
Model_1	0.97249	0.98018	0.9847	0.98904	0.99604	0.99827
Model_2	0.98553	0.98954	0.99129	0.99318	0.99702	0.99845
Model_3	0.95616	0.97053	0.97693	0.98421	0.99617	0.99878
Model_4	**0.92607**	**0.95401**	**0.96444**	**0.97638**	**0.99487**	0.99863
Model_5	0.98502	0.98558	0.98874	0.99132	0.99701	0.99853
Model_6	0.9712	0.97923	0.98367	0.98779	0.99579	**0.99824**

Hole mesura el percentatge de documents no vist pels experts en el top-k. La forma com està construït el corpus, suposa que el document vist pels experts és aquell que contesta a la consulta, el document rellevant. Aleshores, en aquesta mesura el millor model és aquell que té uns resultats menors, perquè significa que sí que ha recuperat més voltes el document rellevant.

6.2.4. Avaluació Externa

L'organització creadora deSentence-Transformer [39]. Ha dut a terme una experimentació per avaluar els diferents models preentrenats, aquesta experimentació avalua els següents aspectes:

- **Representacions vectorials d'oracions**: Avalua el rendiment dels models en crear representacions vectorials d'oracions en 14 diferents tasques de diferents dominis.
- **Cerca semàntica**: Avalua el rendiment dels models en la tasca de cerca semàntica en 6 diferents corpus.

Els resultats mostren l'avaluació de molts models en aquestes diferents tasques, nosaltres soles mostrarem aquells models que hem utilitzat.

Models	Ren. Representacions Vectorials	Ren. Cerca Semàntica	Rendiment Mitjà	Velocitat (frases/sec)	Grandària Model
Model_1	60.18	27.35	43.77	4000	480 MB
Model_2	—-	—	—	—	—
Model_3	**69.57**	57.02	**63.30**	2800	**420 MB**
Model_4	66.76	**57.60**	62.18	2800	**420 MB**
Model_5	—	—	—	—	—
Model_6	64.25	39.19	50.74	**7500**	**420 MB**

Taula 6.10: Resultats avaluació del models feta per SBert framework

El models que no tenen resultats, és perquè no s'ha avaluat, ja que no pertanyen al framework Sentence Transformer.

En conclusió, podem observar a la taula com el millor model que tenim per crear representacions vectorials i de rendiment mitjà és el model_3 (sentence-transformers/all-mpnet-base-v2), però, la nostra tasca principal és la de la cerca semàntica que el model amb millor rendiment és el model_4 (sentence-transformers/multi-qa-mpnet-base-dot-v1).

6.2.5. Conclusions dels Resultats

Una vegada ja tenim tots els resultats, podem fer una comparació exhaustiva dels diferents models. Totes les mesures avaluades donen com a millor model el model_4, a més, l'avaluació externa també dona com millor model aquest per a la cerca semàntica.

La decisió està clara el millor model a utilitzar en aquest projecte és el model **sentence-transformers/multi-qa-mpnet-base-dot-v1**, model especialitzat en la cerca semàntica i que està entrenat per ser multilingüe. Aquest model serà el que utilitzarem en el nostre SRI.

6.3 Entrenament Model Final

Explicat en la Secció 5.1.4, vàrem tindre problemes amb la fase de l'entrenament del model. Encara tot, és un treball que va ser prou exigent i va consumir prou de temps. En aquest apartat farem un repàs complet de la fase de l'entrenament, quins resultats parcials vàrem obtindre i quina va ser la decisió a partir del problema i dels resultats obtinguts que vàrem triar.

L'entrenament del model elegit s'ha realitzat per mitjà d'un script proporcionat per Beir [26]. Aquest script està fet per utilitzar els seus corpus i models, el nostre treball ha consistit en la modificació d'aquest script per adaptar-lo a les nostres dades i el nostre model. Per a més informació aneu a la Secció 4.2.1.4.

L'entrenament està compost per diferents fases, cadascuna amb un objectiu diferent.

6.3.1. Preparació de les Dades

La primera de les fases ha sigut la preparació del corpus d'entrenament. Hem utilitzat el mateix corpus que en l'avaluació dels models, més informació en la Secció 4.2.1.1. Un corpus en castellà i català. El temps d'execució de neteja i construcció del corpus el trobem en la següent secció 6.1.

A diferència de l'etapa d'avaluació, en aquesta fase hem utilitzat la versió d'entrenament del corpus. Una versió composta per més d'1.065.503 consultes, on cadascuna té el seu fragment de text corresponent que la respon. A més a més, la preparació de les dades es fa per mitjà de dos mètodes:

- **load_train(corpus, queries, qrels)**: Aquest mètode consisteix a crear parells consulta-resposta de tot el corpus. És a dir, per cada consulta agafem la referència de l'id del fragment del text que al respon, i construïm un parell consulta-resposta. Així, obtenim les mostres d'entrenament del nostre corpus. Totes aquestes mostres són guardades en un *array* que utilitzarem en el següent mètode.
- **prepare_train()**: Agafem l'*array* del anterior mètode i el transformem en la classe de dades necessària per a l'entrenament del model. En aquest procés fem una vareig de totes les mostres.

6.3.2. Elecció del Paràmetres d'Entrenament

En aquesta segona fase hem fet la tria del paràmetres d'entrenament. En aquesta elecció hem seguit les recomanacions que exposa Sentence Transformers per l'entrenament de models. Aquestes recomanacions són diferents dels paràmetres per defecte que exposa Beir en el seu *script* original. Els paràmetres d'entrenament són els següents:

- **Batch size**: En tindre una memòria limitada de GPU hem utilitzat un *batch size* xicotet de **16**.
- **Epochs**: Aquesta fase d'entrenament el que s'anomena *Fine tuning* i no necessita moltes *epochs* per entrenar-se. Per tant, hem utilitzat 10 *epochs* en aquest entrenament.
- **Evaluations steps**: Els *evaluation steps* fan referència al nombre d'iteracions abans d'avaluar el model. En aquest cas utilitzem 10000 *steps*, en tindre un corpus d'entrenament gran les iteracions d'entrenament són moltes, per tant, necessitem un nombre gran.
- **evaluator**: Per a l'avaluació del model utilitzem l'objecte *InformationRetrievalEvaluator*. Aquest objecte s'encarrega d'avaluar els model utilitzant el corpus d'avaluació. No avalua totes les mesures, avalua algunes de les més importants com: la Precisió, el Recall i el NDCG.
- **Funció de perduda**: Per a la funció de pèrdua utilitzem *MultipleNegativesRankingLoss*. Hem decidit utilitzar aquesta pel fet de contar només en mostres positives, consulta-resposta. Aquesta funció agafa les mostres com parells positius, i per als parells negatius utilitza les respostes de les altres consultes.

  Recomanen el seu ús per entrenar model de similitud entre representacions vectorials, donat que utilitza les respostes d'altres consultes com mostra negativa. Així tenint un vectors positiu i negatiu. No sols això sinó que utilitzem la similitud cosinus per aconseguir els *scores*.
- **Optimitzador** = Adamw. Mètode d'optimització que utilitza el descens per gradient estocàstic. Un dels optimitzadors més utilitzats a l'hora d'entrenar models.
- **Paràmetres de l'optimitzador**: Utilitzem un *learning rate* de $2e-5$, permet a l'optimitzador entrenar adequadament el model. Un *epsilon* de $1e-6$.
- **Weight Decay**: Tècnica de regularització per reduir el sobreentrenament del model. en aquest cas hem utilitzat un valor de 0.01.

6.3.3. Llançament de l'Entrenament

Una vegada preparades les mostres i establerta la configuració de l'entrenament, llancem el procés d'entrenament. El seguiment d'aquest procés es fa per mitjà de les avaluacions que es fan cada 10.000 iteracions i al final de cada *epoch*. El problema d'aquest seguiment és que els resultats de les avaluacions es mostren per pantalla i no es guarden en cap arxiu de text. Es va fer un seguiment manual dels resultats mostrats, cada cert temps revisàvem que tot funcionava correctament i es miraven els resultats.

L'evolució de l'entrenament va ser la següent: les primeres avaluacions mostren una ràpida millora del model, a mesura que l'entrenament avançava començava un estancament del rendiment del model. Cada vegada l'avaluació mostrava un millora més reduïda del model. En les faces final del model vàrem arribar a un punt d'estancament on les avaluacions no mostràvem cap millora aparent significativa.

Una vegada acabat l'entrenament, l'avaluació final mostrava un rendiment del model menor que l'aconseguit pel model base. No podem mostrar una comparativa final del rendiment a causa de problema que explicarem en el següent apartat.

El temps total d'entrenament del model va ser:

- **Temps Total de Còmput**: 6 dies 13 hores 26 minuts 54 segons

Un procés lent i costos computacionalment.

6.3.4. Fallada en l'Entrenament

Una vegada acabat l'entrenament, el nostre pla a seguir era utilitzar el sistema d'avaluacions de model, per analitzar aquest nou model entrenat i comparar-ho amb el model base. No obstant, va succeir una fallada inesperada a l'hora de guardar el model. La carpeta on hauria d'estar el model guardat estava buida, no contenia cap model entrenar ni cap guardat anterior.

Aquesta va suposar una de les decisions més importants del projecte. Aquesta decisió ve marcada per la fallada a l'hora de guardar el model entrenat. Per tant, teníem les següents opcions:

- Utilitzar el model base per al nostre sistema final
- Tornar a entrenar el model

Donada la fallada del model entrenat, hem optat d'utilitzar el model base per les següents raons: El seguiment efectuat de l'entrenament no ha mostrat cap millora sobre els resultats del model base, i el desenvolupament avançat del treball no dona opció a tornar a entrenar el model.

En conclusió, vàrem continuar amb el pla establert sols que utilitzant el model base.

6.4 Comparació dels Índex

En aquest apartat mostrarem i explicarem l'experimentació portada a terme per determinar quin tipus d'índex utilitzar. No sols compararem els tipus d'índex, sinó aquestes proves ens serviran per a testejar el sistema creat. És a dir, en aquesta experimentació tenim dos objectius:

- La comparació entre els dos tipus d'índex: Flats i IVF-PQ.
- Testejar l'estructura i implementació del sistema.

L'experimentació consistiria en la construcció de 2 diferents SRI: el tipus d'índex i la grandària del corpus. Els índex a utilitzar són: un índex Flat i un índex IVF-PQ, explicats en la Secció 4.3.3.7. Les grandàries utilitzades són: 200, 2000, 20000, 200000 documents. Realitzarem una sèrie de proves a cada sistema creat, on mesurem el temps d'execució (Temps d'indexació i Temps de consulta) i avaluarem els resultats obtinguts d'una sèrie de consultes.

Aquesta experimentació ens permetrà fer una comparació entre els dos índexs i poder realitzar la selecció d'acord amb els resultats obtinguts. No sols, sinó que ens permetrà resoldre problemes de la implementació del sistema, i, finalment, obtindre una implementació lliure d'errors i amb un funcionament comprovat. Per poder passar a la construcció del sistema final sabent que tot funciona correctament.

6.4.1. Procés d'Indexació

En aquest apartat mostrarem la comparació entre les diferents implementacions a l'hora de crear l'índex i escriure'l. Explicat amb anterioritat, el procés de creació de l'índex 4.2.2.2, està compost per 4 diferents subprocessos. Per cadascun d'ells calcularem al demora de cada sistema creat.

6.4.1.1. Temps de Desglossament del Corpus

En aquest subapartat mostrem la demora de cada sistema en agafar el corpus i transformar-lo en la classe de dades que necessitem.

Els Temps són els següents:

Com podem comprovar en la Figura 6.1, el temps d'execució en el procés de lectura del corpus i transformació en la classe de dades requerida és un procés lineal. És a dir, el temps de demora del procés creix linealment amb el nombre de documents total a processar.

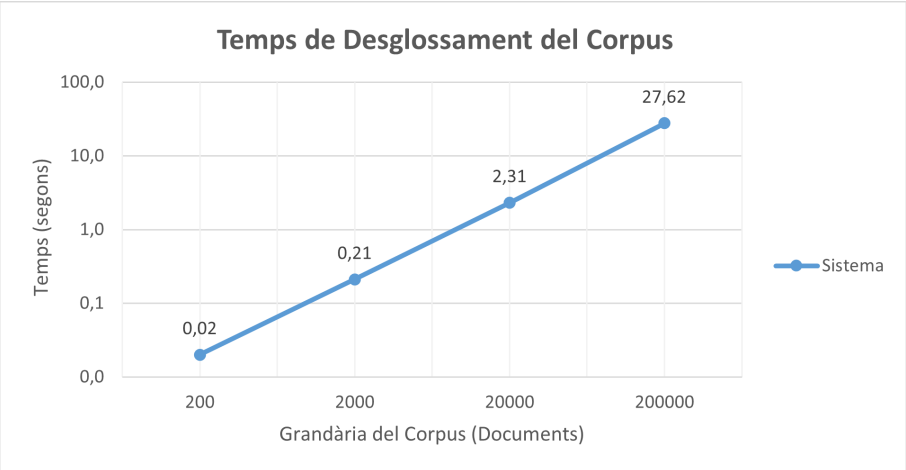


Figura 6.1: Temps de desglossament del Corpus

És un procés ràpid que no consumeix molts de recursos ni temps, a més sols requereix fer-ho una vegada per sistema. I funciona correctament sense cap mena de problema.

6.4.1.2. Temps de Processament del Corpus

Aquest procés és el de processar els documents creats i netejar-los. Els resultats que hem obtingut en calcular el temps de demora són els següents:

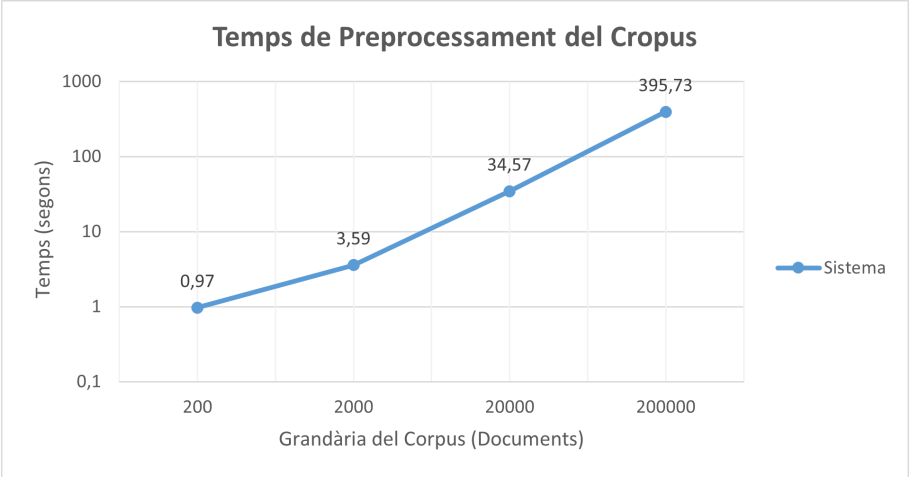


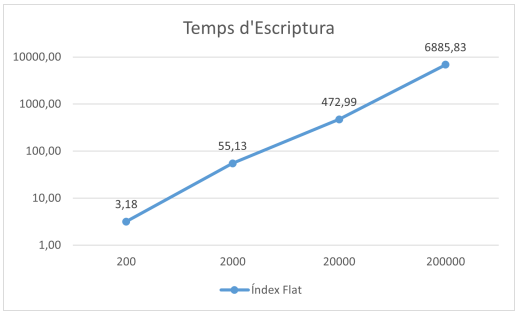
Figura 6.2: Evolució de la demora en el preprocessament del corpus

Es pot comprovar com els sistemes amb més càrrega de documents a processar consumeixen un temps major. Però la Figura 6.2mostra el creixement és lineal a partir de 2000 documents. La quantitat de documents a processar és el factor determinant en el temps que consumira el procés.

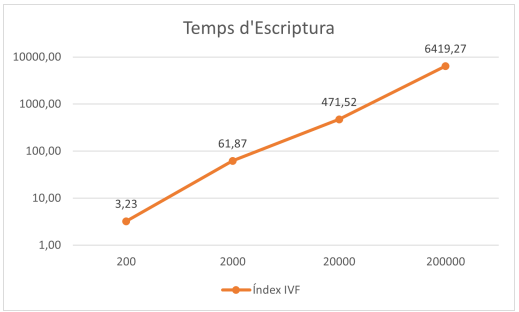
Aquest procés ja requereix un temps de còmput major, donat que hem d'agafar cada document creat i netejar-lo per al seu futur ús. El procés no està accelerat en GPU, ja que no hi havia possibilitat de fer-ho.

6.4.1.3. Temps d'Escriptura dels documents

Aquest procés s'encarrega d'escriure tots els documents creats a la base de dades. Aquesta base de dades és creada per Haystack per guardar els documents per a la seua recuperació posterior. En un principi ja podem deduir que serà un dels processos que més temps consumirà. A continuació mostrem els resultats:



(a) Temps d'escriptura de l'índex Flat



(b) Temps d'escriptura de l'índex IVF

Figura 6.3: Gràfiques dels temps de guardat de cada tipus índex.

Aquesta Figura 6.8 mostra els resultats obtinguts en els sistemes creats, els diferents temps de còmput calculats per cada sistema respecte a l'escriptura en la base de dades, i,

com podem observar, aquells sistemes amb major valors són els que més temps necessiten.

Per altra banda, l'índex Flat, en comparació amb l'índex IVF, té un temps de còmput menor per sistemes amb pocs documents, però superior per sistemes amb grans col·leccions de documents. I al contrari per als índex IVF.

El Figura 6.4 mostra la comparació entre els dos tipus d'índex.

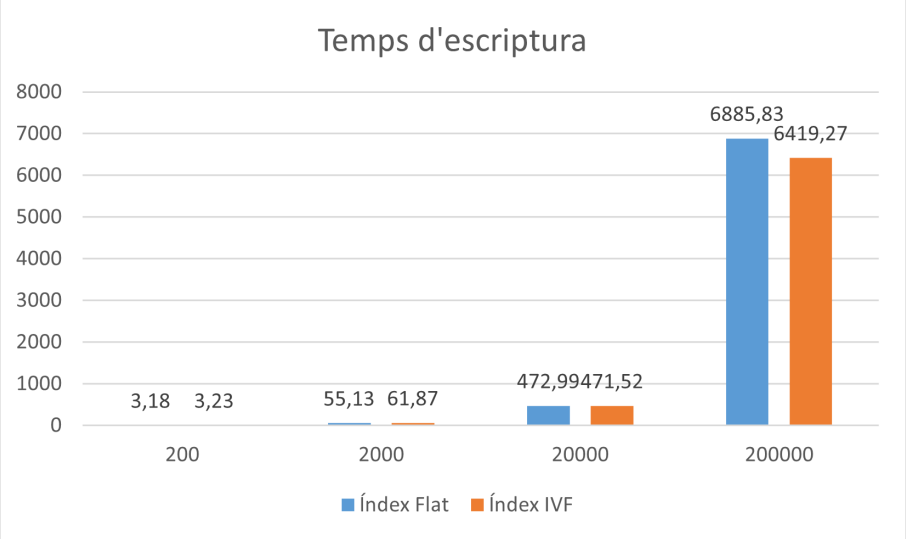


Figura 6.4: Comparació dels temps d'escriptura

Com demostra la Figura 6.4, la nostra conclusió era correcta, l'índex Flat té un creixement superior però un començament inferior. És a dir, per a grandàries superiors consumeix un poc més de temps i per grandàries inferiors menys, tot comparant-ho amb l'índex IVF. En conclusió, no trobem diferències grans entre els dos índexs en aquest procés.

6.4.1.4. Temps d'Entrenament de l'Índex

En aquest apartat, mesurem el temps d'entrenament que necessita l'índex per aprendre a crear els grups i ordenar-los correctament. En aquest apartat sols mesurem a l'índex IVF que és l'únic que necessita entrenament. L'entrenament es necessita per aprendre a fer *clustering*, a agrupar les representacions en grups que les classifiquen. Els resultats obtinguts són els següents:

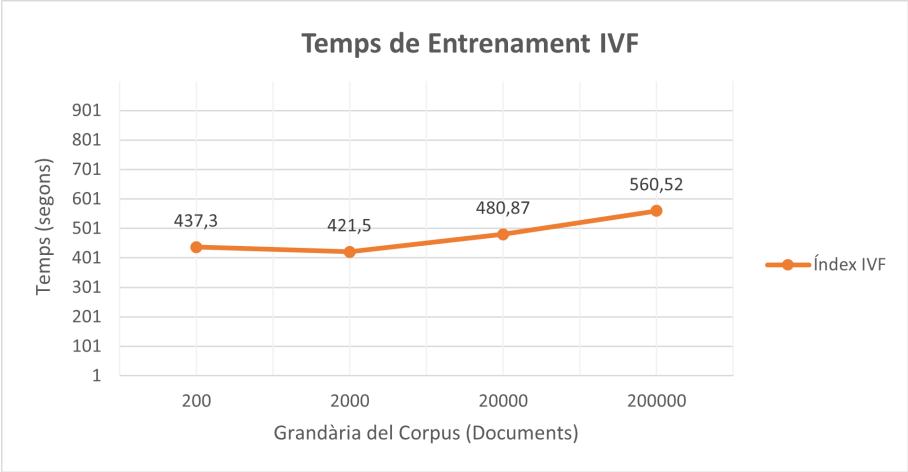


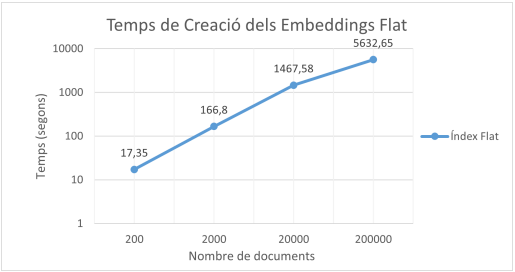
Figura 6.5: Gràfica comparativa del temps d'entrenament

La Figura 6.5 mostra resultats interessants, primer cal recalcar que el mínim de representacions vectorials necessàries per a entrenar són 250, per tant, en corpus xicotets hem d'utilitzar-ho tot o gran part d'ell, per això els resultats són tan parells. Però podem visualitzar, com el creixement no és com en els altres casos, és lineal, però no té un creixement tan clar. Els paràmetres que entren en joc són: el nombre de clústers i els vectors utilitzats en l'entrenament.

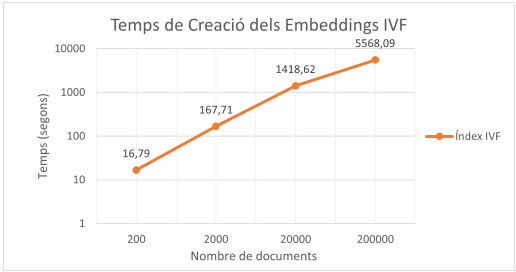
Cal recalcar, que aquest procés sols és necessari portar-ho a terme en utilitzar un índex IVF, és a dir, l'índex Flat no ho ha de fer. Per tant, és un temps que podem estalviar-nos si utilitzem un índex Flat.

6.4.1.5. Temps de creació de les Representacions Vectorials

Aquest subprocés és l'encarregat de crear les representacions vectorials dels documents. Donat que utilitzem models de xarxes neuronals serà un dels processos que més temps consumeixa, encara que s'utilitzen GPUs. Els temps obtinguts són els següents:



(a) Temps de creació de les representacions vectorials de l'índex Flat



(b) Temps de creació de les representacions vectorials de l'índex IVF

Figura 6.6: Temps de creació de les representacions vectorials per als dos tipus d'índex

La Figura 6.6 ens mostra com és un dels processos que més temps consumeix a totes les escales. Tanmateix, si la comparem en la Figura 6.3, podem observar l'efecte de la GPU, com a petita escala no aconsegueix ser més ràpid en més treball a fer, però a gran escala aconsegueix un temps menor de còmput. També trobem que no hi ha grans diferències entre els dos índexs.

Com podem observar en la Figura 6.6, el creixement que obtenim en la creació de les representacions vectorials és lineal amb el nombre de documents. Tampoc trobem cap diferència gran entre els dos índexs en aquestes gràfiques. A continuació els comparem:

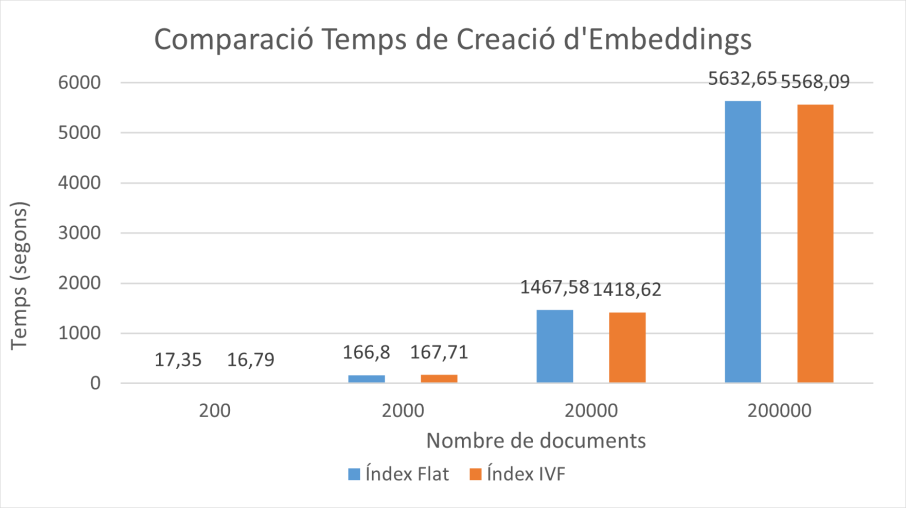
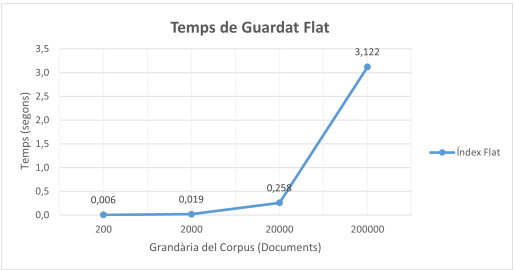


Figura 6.7: Comparació entre els temps de creació dels embeddings dels dos índexs

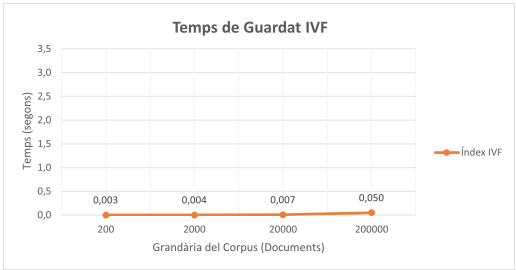
Com observem en la Figura 6.7, l'índex Flat en tots els sistemes creats consumeix més temps, però la diferència és mínima en termes relatius. Per tant, podem concloure que no hi ha diferències entre els dos índexs en aquest procés de la indexació.

6.4.1.6. Temps de Guardat

Els temps de guardat mesura la demora dels diferents sistemes i índexs a l'hora de guardar-ho en el disc. A continuació els resultats:



(a) Temps de guardat de l'índex Flat



(b) Temps de guardat de l'índex IVF

Figura 6.8: Temps de guardat per als dos tipus d'índex

Els resultats mostren com és un procés que no consumeix gran quantitat de temps per fer-ho, i, no sols això, sinó que és el procés on hi ha grans diferències entre els dos índexs. Podem veure com l'índex Flat consumeix més temps per guardar les seues representacions que l'índex IVF. Aquesta causa està relacionada per la memòria que consumeixen ambdós índexs, un no quantifica els vectors, no els comprimeix, i l'altre si.

Aquestes Figures 6.8 mostren el temps consumit en els diferents sistemes, com aquells que utilitzen un índex Flat, té un creixement lineal amb els nombre de representacions vectorials a guardar, i com l'índex IVF, manté un perfil molt baix de temps consumit, ja que quasi no necessita memòria per guardar-ho.

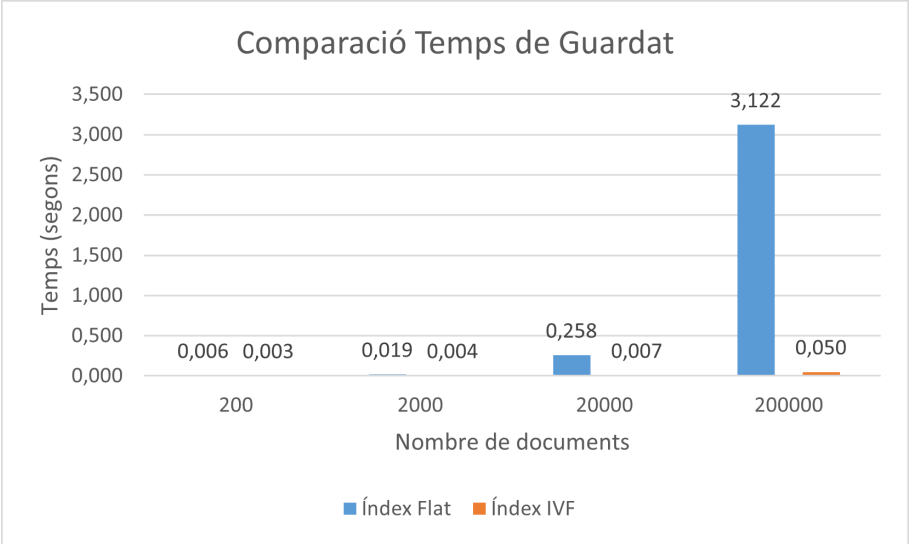


Figura 6.9: Comparació entre els temps de guardat dels dos índexs

Aquesta Figura 6.9 mostra clarament la diferència, com l'índex Flat consumeix molt més de temps. Relacionat amb l'ús de la memòria de cada índex, com podem veure a continuació.

6.4.1.7. Resultat de la indexació

Una vegada ha acabat el procés d'indexació hem obtingut els següents elements:

- **Base de dades**: Un arxiu *.db* encarregat d'emmagatzemar tots els documents creats del nostre corpus. Sol ser l'arxiu més gran, i és on després es recuperen els documents que tenen major rellevància.
- **Índex**: Un arxiu *.faiss* que la seua funció principal és la d'emmagatzemar totes les representacions vectorials de les entrades, i a posteriori fer les comparacions entre la consulta i les entrades.
- **Configuració**: Un arxiu *.json* on s'explica tota la configuració de l'índex.

6.4.2. Tamany de l'Índex en e Disc

En aquest subapartat compararem la memòria utilitzada per cada índex. Aquest és un apartat important perquè en projectes anteriors la memòria dels índex era un punt crític del projecte. En aquest cas, estem utilitzant la tecnologia de Faiss, per a resoldre-ho.

A continuació mostrarem els resultats:

Tipus de Índex	**Grandària del Corpus**			
	200	**2000**	**20000**	**200000**
Índex Flat	1.972 Kb	18.460 Kb	174.481 Kb	1.912.453 Kb
Índex IVF	854 Kb	1.069 Kb	4.695 Kb	12.876 Kb

Taula 6.11: Tamany de disc utilitzat per cada sistema

Aquesta taula mostra una gran diferència entre els dos índexs. L'índex Flat utilitza molta més tamany de disc que l'IVF, donat que no fa ús de cap mena de quantificació

o reducció de la dimensionalitat, és a dir, guarda el vectors com s'originen, amb tota la seua dimensionalitat. Per tant, utilitza molta més memòria.

A continuació mostrem una gràfica que corrobora tot l'explicat amb anterioritat:

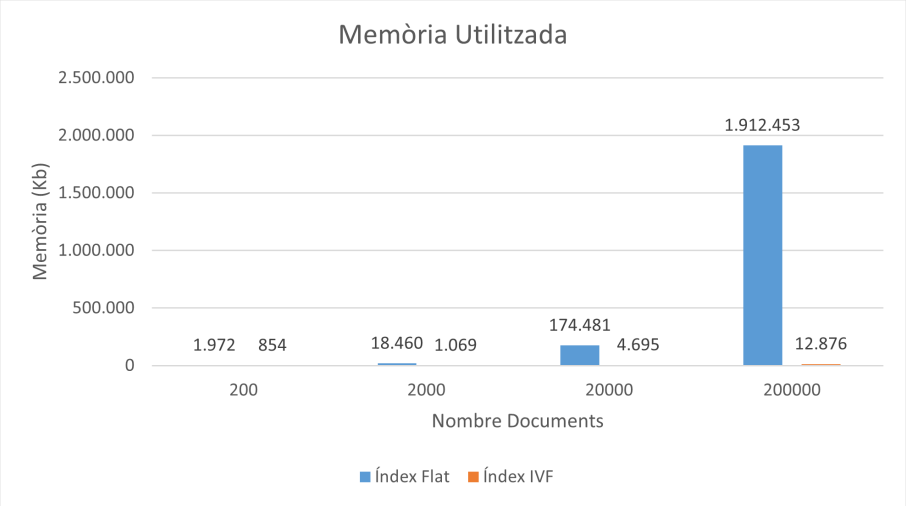


Figura 6.10: Comparació en l'ús de la memòria per cada índex

6.4.3. Procés de Recuperació

6.4.3.1. Temps de consulta

En aquest apartat compararem les diferents implementacions amb els dos índexs, mostrant el temps consumit a l'hora de realitzar les consultes. Teòricament,en termes de velocitat l'índex IVF hauria de ser més ràpid pel fet de fer menys comparacions i amb menor nombre d'operacions per comparació. Per portar a terme aquesta comparació hem realitzat una sèrie de consultes a cada sistema per poder calcular el temps mitjà de resposta. Els resultats són els següents:

Tipus de Índex	**Grandària del Corpus**			
	200	**2000**	**20000**	**200000**
Índex Flat	0.2 seg	0.205 seg	0.273 seg	0.343 seg
Índex IVF	0.183 seg	0.199 seg	0.1785 seg	0.198 seg

Taula 6.12: Temps mitjà per Consulta

La Taula 6.12 mostra en els sistemes que utilitzen un índex Flat, el temps de consulta augmenta amb el nombre de documents indexat, donat que realitza una cerca exhaustiva i compara la consulta en totes les entrades de l'índex.

Aquells sistemes amb un índex IVF, es manté més o menys constant el temps mitjà de resposta, donat que depén del nombre de clústers creats per a l'agrupacions de vectors. Si el nombre d'agrupacions puja, el nombre de vectors dins de cada agrupació baixa (per a un mateix nombre de documents). Tanmateix, el nombre de documents (n) no importa tant, donat que el nombre d'agrupacions a crear és $4 * \sqrt{n}$, que creix a poc a poc.

En la següent gràfica podem comprovar les diferències de temps entre els sistemes:

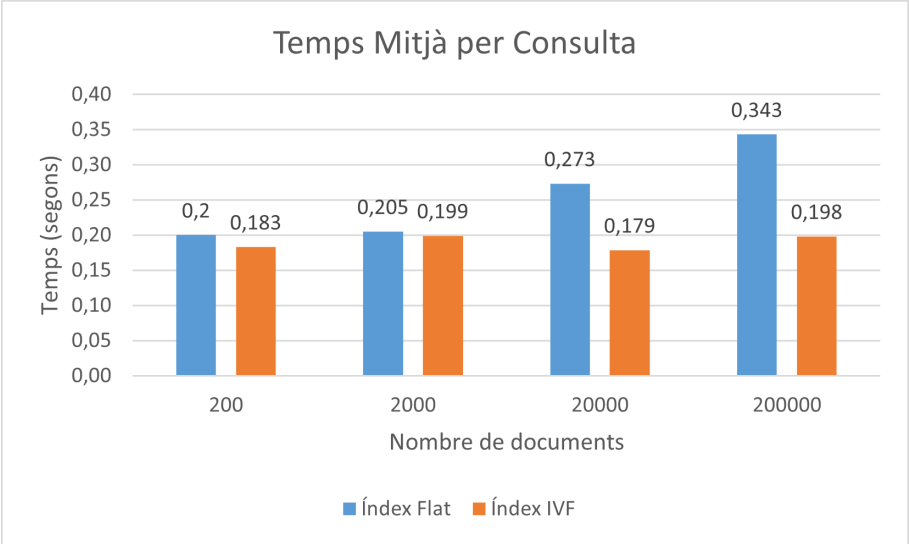


Figura 6.11: Comparació entre els temps mitjà de consulta

Com podem comprovar, l'índex Flat consumeix més temps en augmentar el nombre d'entrades en l'índex o documents en el corpus mentre que l'índex IVF es manté constant. La diferència principal és que en grans col·leccions de documents l'índex IVF funcionària en una rapidesa molt major, donat que no ha de comprar la consulta amb tots els documents indexats.

6.4.3.2. Resultat de la Recuperació

Per cada consulta realitzada el sistema retornarà un arxiu *.csv*, on estarà escrit el rànquing resultat de la consulta. Cada arxiu està compost per un nombre de files igual als documents recuperats. On per cada fila tindrem la següent composició:

- ***query***: La consulta realitzada.
- ***prediction***: La frase original de la representació vectorial que ha recuperat.
- ***prediction_rank***: Posició en el rànquing de la frase recuperada, en el nostre cas un valor entre 1 i 5.
- ***prediction_context***: Context en el qual es trobava la frase recuperada.

6.4.4. Avaluació dels Resultats

En aquest apartat avaluarem els resultats obtinguts pels sistemes amb 200.000 documents indexats. Així, obtindrem un major rang de cerca, és a dir, tindrem més documents indexats on cal esperar més documents significatius. Tanmateix, dona la possibilitat que també siga més difícil recuperar aquest documents.

L'objectiu d'aquestes proves és determinar el correcte funcionament del sistema a una escala més xicoteta i la comparació entre els dos índexs creats. Per això utilitzarem diferents consultes, en els dos idiomes, per veure si diferencia els idiomes i per veure l'eficàcia de cada índex.

L'avaluació que es farà utilitzant els criteris que s'explicaran a continuació sobre el següent conjunt de consultes:

- **Castellà:**
  - ¿Quién ganó las elecciones?
  - ¿Cómo ha quedado el Oviedo?
- **Català**:
  - Problemes en l'1-O
  - Què passa amb l'economia?

6.4.4.1. Criteris d'Avaluació

El criteris d'avaluació són mètriques utilitzades per intentar fer l'avaluació tan objectiva com siga possible. En el nostre projecte utilitzem els següents criteris:

- **Resposta clara**: Mirem de trobar respostes que contestes a les consultes d'una forma precisa i concisa.
- **Segueix els tòpics de la consulta**: Criteri que marca la necessitat que les respostes siguen coherents amb els tòpics principals de les consultes. Que les respostes retornades tinguen els mateixos tòpics o similars que la consulta.
- **Context coherent amb la resposta i la consulta**: Busquem que els contextos d'on s'extrau la resposta siguen coherents amb aquesta. És a dir, que li doten de més significat i seguisquen la línia de contestació. Exemple: Consulta: Com va l'economia?, resposta: bé, context: L'equip de futbol juga bé. Aquest exemple seria roín, donem una resposta clara, però el context no és coherent amb ella i la consulta.
- **Rànquing**: El rànquing retornat ha de ser coherent amb la rellevància presentada per les respostes. Busquem un rànquing que ordene correctament els documents per rellevància. L'anàlisi de la rellevància del document és la part més subjectiva de l'avaluació.
- **Coherència amb els idiomes**: Volem que si la consulta ha sigut realitzada en català les respostes siguen en català, i viceversa. Busquem la coherència lingüística entre consulta i resposta.

L'avaluació dels resultats es farà seguint els anteriors criteris, intentant buscar l'objectivitat.

6.4.4.2. Índex Flat

Analitzant una a una les consultes realitzades:

- **Consulta 1**: La Taula 6.13 mostra els resultats de la primera consulta.

Rànquing	Resposta	Context
1	Rajoy	Tenemos grabaciones y contamos más de 60 en la de Rajoy, dice Alejandro. Rajoy ha ganado por mayoría absoluta.
2	mujeres	A pesar de los buenos resultados que obtienen las mujeres cuando dan ellas el primer paso, los hombres envían más mensajes que ellas
3	Sánchez	Ha insistido en que su formación "ganará las elecciones"frente a un Sánchez que "da miedo", en referencia a la noche de 'Halloween'
4	Pablo Casado	pedido a los presentes que tomen la campaña çon ganas" para hacer a Pablo Casado presidente del Gobierno el 10 de noviembre.
5	Isidoro Ortega	Don de Mejor Ajo Morado; el Cordero de Calidad Diferenciada es para Isidoro Ortega (Albacete); y el mejor Pan de Cruz es el de Juan Pedro e hijos

Taula 6.13: Resultats 1a Consulta de l'índex Flat

Una vegada analitzats els resultats podem extraure la següent informació:

- **Resposta clara**: Les respostes són clares, i poden veure com sí que intenten contestar a possibles guanyadors d'eleccions.
- **Tòpic**: Tenim 3 resultats (1-3-4) on si segueix un tema polític i parla d'eleccions, però en les respostes (2-5) parla d'altra cosa diferents, on intervenen paraules relacionades amb eleccions, com: resultats, obtenen. També, anar amb compte donat que en la resposta 5, sí que parla d'unes eleccions, encara que no siguen polítiques.
- **Context**: En aquest cas passa igual, els context de les respostes (2-5) demostra com no estem parlant de les eleccions polítiques, per tant, no està sent coherent amb la consulta.
- **Rànquing**: Parlem d'un rànquing correcte, a excepció de la resposta 2, les altres sí que estan bé ordenades.
- **Idioma**: Totes les respostes són en castellà com la consulta.

La resposta és correcta i ben construït el rànquing.

- **Consulta 2**: La Taula 6.14 mostra els resultats de la 2a consulta:

Rànquing	Resposta	Context
1	(4-3)	Rayo Majadahonda durante su último partido de liga (4-3), con goles de Carlos Hernández, Johannesson y Joselu.
2	El Real Oviedo ha vencido 7 veces, ha sido derrotado en 10 ocasiones y ha empatado 3 veces	En las salidas, el Real Oviedo ha vencido 7 veces, ha sido derrotado en 10 ocasiones y ha empatado 3 veces en sus 20 duelos jugados hasta ahora
3	Usted no es para mi el alcalde de Oviedo	en absoluto lo público y actúa con el total convencimiento de que lo está haciendo bien. Usted no es para mi el alcalde de Oviedo", ha zanjado López
4	un solo euro para la temporada	Canteli ha destacado que el Principado no ha consignado ün solo euro para la temporada ovetense de Zarzuela y, en cambio, Gijón,
5	18 veces	En el rendimiento como equipo local, Osasuna ha ganado 18 veces y ha empatado 2 veces en sus 20 encuentros disputados hasta ahora

Taula 6.14: Resultats 2ª Consulta de l'índex Flat

Una vegada analitzades les respostes, poden veure:

- **Resposta clara**: Les respostes són clares, i contesten a la consulta efectuada.
- **Tòpic**: Trobem que les respostes (1-2-4-5) sí que parlen de futbol i de resultats, però les respostes (3) no. Però, es pot arribar a entendre la confusió del sistema, donat que no parla de quin Oviedo, la consulta és un pot ambigua.
- **Context**: Trobem un exemple molt clar de l'elecció d'aquest criteri, en el context de la primera resposta, trobem que està parlant del Rayo Majadahonda i no del Oviedo. Passa el mateix per a la cinquena resposta parlem de l'Osasuna. Els tòpics principals són futbol i resultats, però no parlen del Oviedo.
- **Rànquing**: Aquest rànquing ja està pitjor construït, al nostre criteri les respostes (2-4) haurien de passar al davant i les altres repartir-se els altres llocs.
- **Idioma**: Totes les respostes són en castellà com la consulta.

En conclusió, una resposta que és coherent amb el tòpic, però no acaba d'encertar les respostes buscades.

- **Consulta 3**: La Taula 6.15 mostra els resultats de la tercera consulta:

Rànquing	Resposta	Context
1	La por de possibles denúncies per suposat adoctrinament	La por de possibles denúncies per suposat adoctrinament de fet, va portar alguns claustres a minimitzar les explicacions sobre l'1-O a les aules.
2	una aturada de país	Unitaris entre partits i entitats? Només d'entitats? Com veuria una aturada de país aquella data, com es planteja l'ANC. L'1-O és de tothom,
3	que la sentència no acaba amb el conflicte	Us en els incidents i no abordar el problema de fons: que la sentència no acaba amb el conflicte i que hi segueixen havent centenars de milers de catalans.
4	allà on hi hagi coses a discutir"hi seran	ha insistit, sense negar al mateix temps que "allà on hi hagi coses a discutir "hi seran, perquè l'objectiu d'aquesta formació és "fer polític
5	Totes les defenses van qüestionar aquest fet	Totes les defenses van qüestionar aquest fet i van demanar que almenys se'ls deixi accedir a la documentació del cas.

Taula 6.15: Resultats 3ª Consulta de l'índex Flat

Analitzades les respostes, obtenim els següents punts:

- **Resposta clara**: En aquest cas, les respostes no acaben del ser clares, sí que responen a la consulta, però no d'una forma precisa i concisa. Trobem algunes respostes que sense el context no aclareixen la consulta.
- **Tòpic**: Les tres primeres respostes sí que parlen a prop de l'1-O i dels problemes que ha sortit, les altres dues consultes parlen de temes polítics, però no tenim la certesa que és sobre l'1-O.
- **Context**: Tots els contextos sostenen les respostes proporcionades dotant de la informació necessària per a entendre-la. No obstant, la (4-5) no acaben de parla sobre l'1-O.
- **Rànquing**: Un bon rànquing
- **Idioma**: Consulta en català, respostes en català.

- **Consulta 4**: La Taula 6.16 mostra els resultats de la quarta consulta.

Rànquing	Resposta	Context
1	aturar l'economia catalana durant 24 hores	Segons Canadell, aturar l'economia catalana durant 24 hores podria suposar un 0,44% del PIB si fos del 100%, però estima que l'efecte real seria del 0
2	va bé	Parla d'una economia en què les empreses creixen i l'economia va bé", ha criticat, mentre ïgnora les llistes d'espera o els barracons"
3	pèrdua de competitivitat	el fet que durant la reunió del Cercle es recordés la pèrdua de competitivitat de les empreses catalanes.
4	L'economia espanyola perd embranzida	L'economia espanyola perd embranzida. El Banc d'Espanya va prémer ahir el botó d'alarma al constatar que, tot i que en els pròxims anys el creixement
5	-Igual que el 3-	- Igual que el 3-

Taula 6.16: Resultats 4ª Consulta de l'índex Flat

Analitzat el rànquing trobem els següents resultats:

- **Resposta clara**: Respostes que contesten a la consulta de forma precisa i clara.
- **Tòpic**: Totes les respostes parlen de l'economia.
- **Context**: Tots els contextos aporten informació addicional a la resposta i són coherents amb la consulta realitzada.
- **Rànquing**: Satisfactori, molt bé construït.
- **Idioma**: Es respecta l'idioma original de la consulta.

En conclusió, parlem d'un sistema amb índex Flat que ha funcionat realment bé contestant a les consultes. Ha creat rànquings coherents on manté en les posicions principals els documents més rellevants. Encara que en algunes consultes no ha funcionat del tot bé, sempre ha retornat un contingut relacionat amb la consulta, siga pel tòpic o per algunes similituds.

6.4.4.3. Índex IVF

En aquest cas, trobem un sistema construït amb un índex IVF, que millora la velocitat, però obté resultats menys precisos que l'índex anterior. L'anàlisis a les consultes són els següents:

- **Consulta 1**: La Taula 6.17 mostra els resultats de la primera consulta.

Rànquing	Resposta	Context
1	mujeres	A pesar de los buenos resultados que obtienen las mujeres cuando dan ellas el primer paso, los hombres envían más 'mensajes que ellas.
2	qui posa i qui treu llaços	I la prova són els últims enfrontaments entre qui posa i qui treu llaços.
3	Didier Reynders	Sí que s'han validat, però, les declaracions de la francesa Sylvie Goulard, del belga Didier Reynders i de la portuguesa Elisa Ferreira.
4	Raül Romeva	acompanyat de Gabriel Rufián i Carolina Telechea i amb el company Raül Romeva al Senat. Vull continuar lluitant, com ho he fet sempre,
5	JxCat	A l'acte hi han assistit membres de JxCat, ERC, la CUP, l'ANC i Òmnium, Comunistes de Catalunya, Crida LGTBI, Dones per la República, la Intersindical

Taula 6.17: Resultats 1a Consulta de l'índex IVF

Una vegada analitzat els resultats:

- **Resposta clara**: Són respostes clares, però no contesten a la consulta
- **Tòpics**: Cap resposta segueix els tòpic de les eleccions polítiques.
- **Context**: Els context són coherents amb la resposta, però no ho són amb la consulta.
- **Rànquing**: dona igual la construcció del rànquing si les respostes són incorrectes.
- **Idioma**: No s'ha respectat l'idioma original en algunes respostes

- **Consulta 2**: LA Taula 6.18 mostra els resultats de la segona consulta.

Rànquing	Resposta	Context
1	No hay coherencia, solamente imposición	No hay coherencia, solamente imposición, con acciones pequeñas y parches, sin visión de conjunto",ha concluido Lapeña.
2	Se ve camino	Se ve camino, ahora hay que andarlo", ha expresado.
3	val la pena haver-ho fet com ho vam fer	En parlo amb ells i segueixen pensant que val la pena haver-ho fet com ho vam fer. .
4	No sabem què en pensarà Vargas Llosa	No sabem què en pensarà Vargas Llosa.
5	estem en el mateix vaixell i tothom ho passa malament	millor i qui està pitjor. Tots som represaliats, estem en el mateix vaixell i tothom ho passa malament", sentencia la Tamara, que explica que li han o

Taula 6.18: Resultats 2ª Consulta de l'índex IVF

L'anàlisi dels resultats determina:

  - **Resposta clara**: No són respostes coherents ni precises. Cap d'ella contesta la consulta d'una manera clara.
  - **Tòpics**: Cap resposta segueix els tòpics principals de la consulta. Ni futbol, ni resultats, ni Oviedo.
  - **Context**: Els contextos ni aporten informació a la resposta, ni són coherents amb la consulta.
  - **Rànquing**: Cap resposta és rellevant, per tant, la construcció del rànquing és indiferent.
  - **Idioma**: No es respecta l'idioma original de la consulta.

- **Consulta 3**: Aquest cas és estrany, el sistema no ha sigut capaç de trobar informació rellevant sobre la consulta i no retorna cap resultat.

- **Consulta 4**: La Taula 6.19 mostra els resultats de la quarta consulta.

Rànquing	Resposta	Context
1	va bé	Parla d'una economia en què les empreses creixen i l'economia va bé", ha criticat, mentre ïgnora les llistes d'espera o els barracons
2	reduir de manera progressiva les bonificacions	En canvi, proposava, per a aquest tribut, reduir de manera progressiva les bonificacions i plantejava fins a quatre escenaris
3	se aprobará un nuevo Estatuto de los Trabajadores	l estado "será del 2,5 por ciento"y que, además, " se aprobará un nuevo Estatuto de los Trabajadores". Para el candidato socialista al Congreso.
4	disminució de la despesa pública	econòmica presidida per l'austeritat i que impliqui la disminució de la despesa pública i l'aplicació de noves retallades. Des de la formació lila
5	dificulten la creació de llocs de treball fixos	com el turisme o el sector agroalimentari, dificulten la creació de llocs de treball fixos, un fenomen que no passa amb la mateixa intensitat

Taula 6.19: Resultats 4ª Consulta de l'índex IVF

Analitzats els resultats obtenim:

  - **Resposta clara**: En aquest cas les respostes són clares i contesten a la consulta realitzada.
  - **Tòpics**: Totes les respostes mantenen el tòpic de la consulta, parlant sobre l'economia directament o indirectament.
  - **Context**: Els contextos aporten informació significativa a la resposta i són coherents amb la consulta realitzada.
  - **Idioma**: A excepció de la tercera resposta, les altres mantenen l'idioma original de la consulta.

En conclusió, aquest sistema ha funcionat mal, en el 75% de les consultes realitzades ha retornat un rànquing insatisfactori, on no hi havia informació rellevant per l'usuari. Una de les possibilitats del mal funcionament és el fet de dividir els vectors en grups, i seleccionar un sols grup per fer la cerca en profunditat, pot resultar en un rang xicotet de possibilitats de resposta. Produint resultats insatisfactoris.

6.5 Conclusió de l'Experimentació

Per una part, la primera conclusió a què hem arribat ha sigut la tria de l'índex Flat en el sistema final. Primer per la seua recuperació d'informació satisfactòria en totes les consultes realitzades i a un gran nivell de rellevància. Segon, les seues carències d'espai i velocitat no són tan rellevants per baixar el nivell de recuperació exitosa. Finalment, cal exposar que l'ús de l'índex Flat ha resultat en un sistema amb un baix rendiment, ja que no retornava la informació més rellevant per a la consulta; encara que tingués velocitat de consulta més ràpida que els obtinguts amb l'índex Flat.

Per altra part, l'obtenció de resultats satisfactoris basats en els nostres criteris, demostra un funcionament del sistema adequat als objectius d'aquest projecte. Aquest sistema ens dona la certesa d'un bon funcionament en el sistema final.

Finalment, voldria exposar com el model utilitzat per a la feina de la recuperació ha sigut un encert, ja que com podem comprovar en els resultats de les consultes, retorna una resultats que diferencien el català i el castellà, i proporcionen informació rellevant a la consulta. Aquests resultats són gràcies, en part, per la creació de bones representacions vectorials, donat que crea *embeddings* contextuals que contenen tota la informació semàntica i contextual per a la seua futura recuperació.

CAPÍTOL 7
Model Final

Finalment, després d'un procés d'experimentació on: primer hem analitzat diferents models per a la tasca de recuperació d'informació i triat aquell que millor s'ajusta a les nostres dades, segon hem entrenat aquest model, i, finalment la creació de dos sistemes amb diferents tipus d'índex per a la seua comparació. En aquest capítol ens centrarem en l'avaluació detallada dels resultats obtinguts pel sistema final. A més, exposarem també quins han sigut els temps de còmput en tots els apartats de construcció i la posterior recuperació.

7.1 Temps de Construcció

En aquest apartat desglossarem el temps necessari per a fer la construcció d'aquest sistema final. Els temps han estat els següents:

Tipus de Procés	**Temps de Còmput**
Desglossament del Corpus	1 min 51 segons
Processament del Documents	29 min 14 segons
Escriptura de la BD	24 hores 43 min 22 segons
Creació de les representacions vectorials	5 hores 39 min 19 segons
Guardar índex	5.67 segons

Taula 7.1: Temps de còmput dels diferents processos del sistema

Com es pot comprovar en la Taula 7.1, que mesura el temps de còmput dels diferents processos de construcció del sistema, la necessitat de temps augmenta, a causa del gran nombre de documents a indexar, tots els processos han pujat una escala en consumició de temps, respecte a les proves.

Estem parlant d'un total de **111234.13** segons que equivalen a unes **30 hores 53 minuts**. És un temps relativament alt, no obstant, com ja hem explicat és un procés que sols cal fer-ho una vegada.

Com ens mostra la Taula 7.1, el procés que més temps ha consumit ha sigut el d'escriure els documents a la Base de dades, ocupant més d'un 80% del temps total de la construcció del sistema.

7.2 Avaluació dels Resultats

L'avaluació realitzada ha sigut subjectiva, donat que no tenim cap mètode o mesura objectiva adaptada al nostre corpus. El nostre corpus no té cap arxiu de referència on estan les soluciones, les respostes correctes. Tanmateix, aquesta avaluació s'ha dut a terme seguint uns criteris preestablerts explicats en la Secció 6.4.4.1

7.2.1. Consultes

Per fer una avaluació el mes objectivament possible, hem seleccionat consultes en els dos idiomes, i una alta alternança de tòpics. Els tòpics més predominants al corpus són: Economia, Política i Esports. A continuació mostrem la llista de consultes realitzades:

1. ¿Quién ganó las elecciones?
2. ¿Cómo ha quedado el Oviedo?
3. Problemes en l'1-O
4. Què passa amb l'economia?
5. Los nuevos partidos crecen
6. Com va el Barça en la lliga?
7. ¿Cómo va la economia?
8. Qui serà el nou president?
9. ¿Cómo evoluciona la guerra?
10. Quins són els canvis principals?

Aquestes consultes formen un conjunt variat tant en idiomes com en tòpics, obtenim respostes variades. Aquestes consultes ens possibiliten l'avaluació dels sistema davant dels diferents tòpics presenta i dels idiomes.

7.2.2. Avaluació

Com hem explicat abans, anirem avaluant cada consulta per separat, per veure el rendiment del sistema individualment, i, finalment, exposarem una conclusió de totes les avaluacions.

7.2.2.1. 1a Consulta

La primera consulta: "¿Quién ganó las elecciones?". Té com a tòpics la política i les eleccions. L'objectiu és documents que parlen sobre els guanyadors de les eleccions. La Taula 7.2 mostra els resultats de la 1a consulta.

Rànquing	Resposta	Context
1	Todos	Como regla general, las elecciones siempre las ganan todos. O si quieren decirlo de otra manera, nunca las pierde ninguno.
2	Rajoy	Tenemos grabaciones y contamos más de 60 en la de Rajoy, dice Alejandro. Rajoy ha ganado por mayoría absoluta.
3	Los dos	Entonces, volveríamos al inicio: las elecciones las habrán ganado los dos, a pesar de haberlas perdido ambos. Uno en las urnas, otro en los post elecciones
4	Líder del PP	El resultado es desolador para el líder del PP, que ha ganado por mayoría absoluta.
5	Rajoy	Los autores del invento han sacado los siguientes resultados. “Rajoy tenía, por lo menos, 150 cacas. Pablo Iglesias tiene 30. Pedro Sánchez

Taula 7.2: Resultats 1a consulta

Utilitzant els criteris d'avaluació, podem observar el següent:

- **Respostes clares**: En aquest apartat podem veure com es respon perfectament a la consulta realitzada, consultem pel guanyador de les eleccions, i ens respon en respostes com “Rajoy, el PP“. Que són respostes valides. Encara que pot ser les respostes “Todos“ i “Los dos“ no contesten clarament la consulta, acompanyades del context són respostes valides.

- **Tòpic**: Observem com en totes les respostes generades, el tema principal és el mateix que el de la consulta realitzada, en aquest cas la Política.

- **Context**: El context de l'1-4 ens mostra com la resposta és correcta, i dota de més significat. Però en la resposta 5, observem que parla d'un invent, pel que potser mentida que Rajoy guanyara. Encara així ens mostra un context relatiu a la consulta i que és rellevant.

- **Rànquing**: Al nostre criteri pensem que el rànquing és coherent amb la consulta donat el context. Totes parlen del guanyador de les eleccions, i és complicat fer un rànquing propi. Encara que aquelles respostes que parlen de líders polítics podrien anar en els primers llocs del rànquing.

- **Idioma**: En tot moment s'ha respectat l'idioma original de la consulta.

En conclusió, parlem d'un treball ben fet, donat que el rànquing retornat té documents molt rellevants per a la consulta realitzada. On tots respecten l'idioma, tracten del mateix tòpic i són rellevants a la consulta.

7.2.2.2. 2a Consulta

La segona consulta: “¿Cómo ha quedado el Oviedo?”, tòpics principals Oviedo i Futbol. La Taula 7.3 mostra els resultats.

Rànquing	Resposta	Context
1	Cerrarà	La de Oviedo (Colloto), que también cerrará, aún no ha iniciado los paros, pero el Comité de Empresa ha anunciado que comenzará la huelga indefinida
2	Octavo	el conjunto chicharrero es decimoséptimo, mientras que el Oviedo es octavo tras la finalización del encuentro.
3	Real Oviedo 1, Lugo 1	Eduard Campabadal con un centroal área. 90'+1' ¡Goooooool! Real Oviedo 1, Lugo 1.
4	Capital de las compras	innovadores y colaborativos y proyectar a Oviedo como capital de las compras,
5	(1-1)	Real Oviedo como local ante un Numancia con el que finalmente repartó puntos (1-1) al no se capaz de cerrar el encuentro

Taula 7.3: Resultats 2a consulta

Analitzat el rànquing i els documents utilitzant els criteris:

- **Respostes clares**: Quasi tots els llocs del rànquing responen amb una resposta clara i coherent, dins del seu àmbit. Donat que en el lloc 1 i 4, la resposta no està relacionada amb futbol, però sí amb Oviedo, podríem dir que no és una resposta del tot clara. Tot i això, les respostes 2-3-5 contesten perfectament a la consulta formulada.
- **Tòpic**: Com hem exposat abans, es manté el tòpic en els lloc 2-3-5 i parcialment en l'1-4. Donat que la consulta no parla específicament del futbol, encara que se sobreentén, és complicat per al model agafar el seu significat. Donat que si parlàrem del "Real Oviedo" segurament el tòpic principal seria futbol.
- **Context**: Un context que aporta informació addicional que ens ajuda a col·locar la resposta en un espai coherent. En general, tots els contextos ajuden a dotar d'informació a la resposta i veure si és correcta.
- **Rànquing**: Al nostre criteri col·locaria les respostes 2-3-5 com les primeres i després les altres dues, encara i tot, retorna un rànquing prou coherent i informació molt rellevant.
- **Idioma**: Totes les respostes respecten l'idioma de la consulta.

En aquest cas tenim dos respostes que no han respectat del tot els tòpics de la consulta, però més que un error del sistema, és que la consulta és poc específica. És a dir, deixa caure que parla d'un equip de futbol, però en cap cas ho diu específicament. En conclusió, un bon treball fet pel sistema.

7.2.2.3. 3a Consulta

La tercera consulta: "Problemes en l'1-O", tòpics principal Societat i Política. El rànquing resultat disponible en la Taula 7.4.

Rànquing	Resposta	Context
1	Insulta i agressions	Davant del relat d'insults i agressions que van oferir fa setmanes els agents que van intervenir l'1-O, els primers testimonis de les defenses
2	La por de possibles denúncies per suposat adoctrinament	La por de possibles denúncies per suposat adoctrinament de fet, va portar alguns claustres a minimitzar les explicacions sobre l'1-O a les aules.
3	Els que ni tan sols van poder trobar les urnes	Muntar una operació seriosa per evitar l'1-O, els que ni tan sols van poder trobar les urnes, tenen el seu cap turc. Juguen, tot s'ha de dir
4	Allunyament de moltes persones que viuen a Catalunya	no hi ha cap dubte, han contribuït a un cert allunyament de moltes persones que viuen a Catalunya respecte del vincle que fins aleshores mantenien
5	No va ser legal	"Va reconéixer que l'1-O no va ser legal, que no acceptarà ser conseller i que plegarà com a diputat si el seu grup aposta de nou per la via unilateral"

Taula 7.4: Resultats 3a consulta

L'anàlisis del resultat és el següent:

- **Respostes clares**: Totes les respostes responen clarament a la consulta realitzada aportant informació rellevant al respecte. Uns exemple: "Insults i agressions" , "No va ser legal", etc.. trobem com responen clarament a la consulta i retornen un document rellevant per a l'usuari.

- **Tòpic**: En totes les respostes parlem clarament del procés 1-O i dels temes socials i polítics, i també de Catalunya.

- **Context**: Els contextos aporten informació necessària i rellevant per a entendre millor la resposta. Comprovant que els resultats parlen dels tòpics requerits. Tots ells doten de més informació sobre l'1-O.

- **Rànquing**: La part més subjectiva de l'anàlisis, en aquest cas tots els resultats tenen una rellevància molt parell, i no està res clar un millor rànquing. Totes les respostes són rellevants a la consulta.

- **Idioma**: Es respecta l'idioma de la consulta en totes les respostes.

En conclusió, ens troben davant d'un altre bon treball del sistema en crear un rànquing de documents ordenats per rellevància davant d'una consulta realitzada. Donat que tots els documents recuperats aporten informació rellevant a la consulta.

7.2.2.4. 4a Consulta

La quarta consulta: "Què passa amb l'economia?" Tòpic únic i principal l'Economia. L'objectiu és documents que parlen de l'economia catalana o espanyola. La Taula 7.5 mostra els resultats.

Rànquing	Resposta	Context
1	nou terratrèmol	sostenible, i accelerar-ne les mesures. Ni que sigui perquè aquest nou terratrèmol que arribarà ens agafi econòmicament més sòlids
2	La bombolla torna a inflar-se	La bombolla torna a inflar-se, però amb nous fonaments.
3	L'economia de l'euro es va enfonsar un 0,6%	Publicades aquesta setmana, confirmen l'avís de l'FMI: l'economia de l'euro es va enfonsar un 0,6% l'últim trimestre de l'any passat, encadenant tres trimestres
4	va bé	Parla d'una economia en què les empreses creixen i l'economia va bé", ha criticat, mentre ïgnora les llistes d'espera o els barracons"
5	L'economia espanyola perd embranzida	L'economia espanyola perd embranzida. El Banc d'Espanya va prémer ahir el botó d'alarma al constatar que, tot i que en els pròxims anys el creixement

Taula 7.5: Resultats 4a Consulta

Una vegada avaluats els resultats extraem els següents punts:

- **Respostes clares**: Totes les respostes són rellevants i clares a la consulta. Totes parlen sobre l'economia i responen a la consulta aportant informació rellevant. L'única consulta dubtosa és la primera, encara que el context la defineix com a vàlida.
- **Tòpic**: Tots els documents recuperats parlem sobre l'economia i la seua evoluciona. Per tant, en totes les respostes es manté el tòpic central.
- **Context**: Tots els contextos d'on es recupera la resposta aporten informació necessària per a entendre-la millor, i doten de més rellevància als documents recuperats. Donat que demostren que parlen sobre el tòpic.
- **Rànquing**: Al nostre paréixer el rànquing funciona correctament, encara que no estem segurs sobre la primera posició. Pensem que és millor canviar-la per alguna de les altres posicions.
- **Idioma**: En tot moment es manté l'idioma seleccionat per la consulta.

En conclusió, trobem una consulta ben resposta amb el rànquing oferit, donat que aporta 5 documents rellevants.

7.2.2.5. 5a Consulta

La cinquena consulta: "Los nuevos partidos crecen", amb tòpics principals els partits polítics i la política. Els resultats esperats són documents que parlen sobre els nous partits polítics i les seues accions en política. La Taula 7.6 mostra els resultats.

Rànquing	Resposta	Context
1	El viejo duopolio se desvanecerá	Cómo, con quién y para quienes van a gobernar. El viejo duopolio se desvanecerá por la irrupción de dos partidos nuevos.
2	Podemos. El otro, Ciudadanos	reflexiones sobre lo que puede ser el partido más nuevo: Podemos. El otro, Ciudadanos, ya lleva tiempo a nivel de Cataluña, junto con alguna
3	La formación de un nuevo gobierno	tienen una inercia que te acaba arrastrando" , añade. La formación de un nuevo gobierno era el primer gran examen para los nuevos partidos.
4	el Congreso Nacional	El nuevo partido también dispondrá de un órgano máximo, el Congreso Nacional, que se reunirá cada cuatro años, y una Asamblea Nacional, que lo hará,
5	sobreesfuerzo de votos que han de obtener	Los dos nuevos partidos pequeñas posibilidades, por el sobreesfuerzo de votos que han de obtener. Esto, que el PP y el PSOE lo tienen muy estudiado,

Taula 7.6: Resultats 5a consulta

Una vegada analitzats aquest resultats amb les criteris:

- **Respostes clares**: Trobem unes respostes poc clares, cap d'elles parlen del creixement com a tal, però sí que encerten de parlar dels nous partits polítics, com Podemos i Ciutadans (el corpus recull notícies del 2019 cap darrere). Encara i tot, no respon del tot amb la consulta realitzada.
- **Tòpic**: En tot moment parlen de política i dels nous partits, per tant, sí que mantenen el tòpic els documents recuperats.
- **Context**: El context és molt necessari en aquestes respostes, perquè en on trobem realment les respostes requerides. El context dota de tota la informació necessària per entendres les respostes i saber si són rellevants els documents recuperats.
- **Rànquing**: El rànquing és correcte, excepció del 4 document, que o el posaria el 5 o el llevaria. Donat que si parla d'un nou partit, però no explica molt més.
- **Idioma**: Tots els documents recuperats estan escrits en el mateix idioma que la consulta

En conclusió, a excepció del 4 document que parla d'un nou partit, però tampoc ens aporta molta informació rellevant. tots els altres documents parlen sobre els temes principals i estan bé ordenats. És a dir, un bon treball

7.2.2.6. 6a Consulta

La sisena consulta: "Com va el Barcelona en la lliga?", consulta relaciona amb els esports i el Barça. L'objectiu és obtindre documents que ens parlen de la temporada del Barça, no especifiquem quin equip del Barcelona. La Taula **??** mostra els resultats de la consulta.

Rànquing	Resposta	Context
1	Sense celebracions oficials	Més de quatre mesos que la Lliga se celebrava al Barça. Però sense celebracions oficials. D'una manera invisible i discreta, el títol és nostre
2	Indestructible	Lliga en la qual el millor qualificatiu per al Barça ha sigut el d'indestructible. Fins a nou vegades va remuntar i en altres ocasions va eixir viu
3	Campiones de Lliga	Barça. L'equip femení, que també va participar de la festa com a campiones de Lliga, i que van ocupar el primer dels tres autocars de la comitiva
4	Campió	I com veu la Lliga, ara. Si el Granada s'estigués jugant el descens, encara, però crec que tot quedarà igual i el Barça serà campió.
5	el Barça ha sumat 11 victòries en 12 partits i un sol empat	Queda un partit per acabar la primera volta, el Barça ha sumat 11 victòries en 12 partits i un sol empat, superant en 10 punts el Liceo i el Noia

Taula 7.7: Resultats 6a consulta

Analitzat el rànquing utilitzant els criteris obtenim:

- **Respostes clares**: Totes les respostes són rellevants i clares, aporten informació relacionada amb la consulta. La més dubtosa és la primera la qual no acaba de parlar de com va el Barça, però el context sí que aporta informació sobre l'estat en al lliga.
- **Tòpic**: Tots els documents recuperats tenen com temes central el Barça i el seu estat. Així, que podem dir que sí que mantenen el tòpic principal.
- **Context**: Tots els contextos aporten informació necessària per a entendre la resposta i veure que el document és rellevant.
- **Rànquing**: Exceptuant la primera posició, que sense el context té menteix un poc, que la baixaria de posició, totes les altres segueix un rànquing coherent.
- **Idioma**: L'idioma és respectat per tots els documents recuperats.

En conclusió, ens trobem davant d'un rànquing que retorna 5 documents rellevants a la consulta, amb un grau de satisfacció alt per part de l'usuari.

7.2.2.7. 7a Consulta

La setena consulta: "¿Cómo va la economia?", consulta relacionada directament amb l'economia, amb dos objectius: retornar documents en castellà que parlen de l'economia i fer un comparació amb la quarta consulta realitzada, molt pareguda però en català. Els resultats es poden observar en la Taula 7.8.

Rànquing	Resposta	Context
1	ralentización es cada vez más evidente	De manera directa sobre una economía, la española, cuya ralentización es cada vez más evidente, enmarcada en un elevado endeudamiento del sector público
2	buena marcha de las cifras macroeconómicas	todas las fuerzas políticas, incluido Podemos , por la buena marcha de las cifras macroeconómicas. Sin embargo, estos mismos representantes políticos
3	ralentización -Igual que 1-	Igual que 1
4	saneada	Crecimiento imparable de las desigualdades, ¿se puede hablar de economía saneada, es decir, libre de cargas, como precisa el diccionario?
5	mi economía la que está de capa caída	más de 4000 empresas públicas, etc. ¿La economía de quien?.¿Por que no será la mía? .Y perdonarme si es solo mi economía la que está de capa caída

Taula 7.8: Resultats 7a consulta

Una vegada analitzat el resultat obtenim els següents punts:

- **Respostes clares**: En aquest cas trobem respostes clares, però poc encertades. És a dir, les respostes diuen una cosa però el context altra, com l'exemple del document 4. Tanmateix, totes aporten informació rellevant del tema, donat que parlen de l'economia.
- **Tòpic**: Tots els documents recuperats aporten informació rellevant dels tòpics principals. En aquest cas l'economia.
- **Context**: Com hem explicat abans, el context de la resposta 4 es contradiu a les respostes generades. tot i això, totes parlen de l'economia. Els context de les primeres consultes aporta informació rellevant a la consulta i té coherència amb les respostes.
- **Rànquing**: Un rànquing coherent.
- **Idioma**: Tots els documents recuperats estan escrits en castellà

En conclusió, torna a passar el mateix que en altres consultes, la poca especificació de la consulta genera en resultats poc concrets. De quina economia parlem?, les consultes si són més específiques poden funcionar millor. Encara així els documents aporten informació rellevant sobre economia.

7.2.2.8. 8a Consulta

L'octava consulta: "Qui serà el nou president?", consulta relacionada amb els presidents i la política. Sabent que l'ambigüitat treu resultats inesperats, hem volgut utilitzar aquesta consulta per veure si també extrau resultats sobre president d'equips d'esports o altres. Encara que el seu principal tema siga la política. Els resultats es mostren en la Taula 7.9.

Rànquing	Resposta	Context
1	Pau Presas	Pau Presas (Cassà de la Selva, 1989) ja és el nou president d'ERC a les comarques gironines, fent tàndem amb Laia Cañigueral (Cassà de la Selva, 1981)
2	Ghanuchi	El primer ministre Ghanuchi assumeix la presidència interina del país.
3	Jordi Turull	català quan expliqui davant el ple la nova composició del Govern, sinó que ho seguirà fent Jordi Turull com a president del grup parlamentari de CiU.
4	Tomás Gómez	al costat de Trinidad Jiménez: "A partir d'ara tots a secundar [a Tomás Gómez] per a impulsar el projecte socialista a Madrid i portar-lo a la victòria
5	Poblet	Des de l'any 2007, Poblet també és el president de la Diputació de Tarragona

Taula 7.9: Resultats 8a consulta

Una vegada analitzats els resultats seguint els criteris obtenim:

- **Respostes clares**: totes les respostes més concretes són noms de persones que fan referència a presidents. En ser la consulta en català quasi totes les respostes són de presidents catalans o de grups de Catalunya.
- **Tòpic**: Si revise tots els documents trobem com tots mantenen el tòpic de parlar de la política i de presidents.
- **Context**: Els context aportem molta informació necessària, ja que sols un nom no significa res, però si mirem els contextos trobem que fan referència a presidents de partits polítics o de diputacions, etc. Ens dona la certesa del fet que els documents recuperats parlem dels temes principals. El context és coherent amb les respostes.
- **Rànquing**: un rànquing molt coherent, donat que totes les respostes parlen de presidents. sobre tot el número 1 que parla del nou president, just la consulta.
- **Idioma**: Tots els documents estan escrits en català.

En conclusió, una resposta molt encertada, un rànquing on els seus 5 documents aporten informació rellevant i ordenat d'una forma coherent.

7.2.2.9. 9a Consulta

La novena consulta: ´´¿Cómo ha evolucionado la guerra?", una consulta amb temes principals com la guerra i la societat, un poc diferent de les altres provades. L'objectiu és aconseguir un rànquing que retorne notícies que parlen de la guerra, com és habitual, no especifiquen de quina guerra per veure quins resultats obtenim. La Taula 7.10 mostra els resultats.

Rànquing	Resposta	Context
1	La guerra sigue	La guerra sigue
2	La guerra sigue aquí	que era esencia del pasado. Os engañé. Pido perdón por ello. La guerra sigue aquí. Y vosotros no habéis sido preparados siquiera para saber enfrentaros.
3	descoratjadora	Efectivament, la guerra és descoratjadora. Però és que així ho és també la nostra vida de cada dia. Una de les escenes de «Guerra», estrenada al Grec
4	Espero que la guerra acabe pronto	Espero que la guerra acabe pronto para volver y que podamos reconstruir nuestro país"
5	La 'guerra' contra la ganadería porcina intensiva continuará	La 'guerra' contra la ganadería porcina intensiva continuará y es previsible que arrecie con la llegada de las próximas Elecciones Autonómicas

Taula 7.10: Resultats 9a consulta

Una vegada analitzats els resultats obtenim:

- **Respostes clares**: Obtenim resultats bons, que parlen de la guerra i la seua evolució.
- **Tòpic**: Tots els documents recuperats parlen d'alguna guerra en particular, ja siga, una guerra militar, una guerra porcina o una guerra inventada.
- **Context**: Tots els contextos aporten informació rellevant per entendre la resposta i deixar clar que el document recuperat és rellevant. A excepció de la primera resposta que no tenim molt clar de què guerra parla, ni res.
- **Rànquing**: Excepció de la primera resposta, que entenem perquè està la primera donat que respon molt bé la consulta, però realment no aporta informació rellevant. Totes les altres respostes estan bé ordenades.
- **Idioma**: La segona resposta trobem que és en català, açò ve donat, perquè els idiomes català i castellà comparteixen moltes paraules en comú i "guerra" es una d'elles. Aleshores contextos similars creen embeddings similars, pot ser, que els contextos de la paraua guerra en català i castellà tinguen embeddings similars.

Una consulta que ha funcionat relativament bé. Podem veure, com exposen diferents tipus de guerres, donat que no hem especificat. Però, no acabat de retornar tots els documents en castellà, ni tampoc tots han sigut rellevants. A més, pareix que la 1a resposta algun aspecte de la generació de la seua representació vectorial ha fallat o el fet que siga una frase sencera també.

7.2.2.10. 10a Consulta

La dècima consulta: ´´Quins són els canvis principals?", és una consulta prou ambigua, no especifiquem de quins canvis estem parlant. No obstant, imaginem que els principals tòpics són els canvis i la societat. Els resultats són mostrats en la Taula 7.11.

Rànquing	Resposta	Context
1	protocols d'actuació	situació es va complicar quan Torra va plantejar "canvis en els protocols d'actuació" dels Mossos. El president va emplaçar Buch a parlar-ne diumenge
2	tractats per decidir la mida de la representació d'una Escòcia independent a la UE	que els canvis més bàsics als tractats per decidir la mida de la representació d'una Escòcia independent a la UE serien ""fàcils""perquè es podrien"
3	frenar el relleu als Mossos	"fora de qualsevol debat partidista". Un canvi de criteri –frenar el relleu als Mossos– que la CUP va criticar ahir. "Volem que JxCat i ERC
4	L'aparició de la joventut com a nou subjecte històric	L'aparició de la joventut com a nou subjecte històri.Abans de Maig del 68 els xiquets i les xiquetesno anaven a la mateixa aula a primària ni albatxillerat i només una ínfima minoria arribava a la Universitat
5	les normes també han de canviar	del PSOE, ara a l'oposició, va justificar que «la societat canvia» i que, amb ella, «les normes també han de canviar». Més informació de Societat.

Taula 7.11: Resultats 10a cosulta

Una vegada avaluats els resultats:

- **Respostes clares**: Totes les respostes parlen de cavis presents o a futur, per tant, sí que són respostes clares i coherents.
- **Tòpic**: tots els documents recuperats tenen informació rellevant de al consulta, és a dir, mantenen el tòpic central, que són els canvis i la societat.
- **Context**: Els contextos, com en tots els casos, aporten informació necessària per a entendre quins són els canvis, i si els documents són rellevants o no.
- **Rànquing**: Al nostre criteri, pensem que els últims documents recuperats són els més rellevants, encara així és un bon rànquing.
- **Idioma**: tots mantenen com l'idioma principal el català.

El rànquing és coherent i els documents recuperats aporten informació rellevant a la consulta. En conclusió, el sistema ha funcionat correctament.

7.3 Conclusió dels Resultats

Els resultats han sigut positius, hem creat un sistema amb la capacitat de respondre consultes amb un llistat ordenat de documents amb informació rellevant. Les consultes realitzades mostren com en quasi tots els documents recuperats els tòpics principals de la

consulta es mantenen i contesten a aquesta consulta. És a dir, hem sigut capaços de crear un SRI amb un alt grau de satisfacció seguint els criteris explicats en la Secció 6.4.4.1.

Com demostren els resultats, quasi tots els rànquings són coherents i aporten informació rellevant en els documents recuperats, però hi ha algun punts on tindre cura:

- L'ambigüitat de les consultes afecten en prou grau als documents recuperats en les respostes.
- El català i el castellà comparteixen paraules, per tant, tenen contextos pareguts i, per tant, les representacions vectorials poden ser similars. Afectant la recuperació, ja que, d'una consulta en castellà pot eixir un document en català.
- Documents que es repeteixen, donat que indexem per frases i paraules, hi ha espai que es poden recuperar dues voltes. Per tant, produir un duplicat en el rànquing.

L'ús d'un índex Flat ha permés obtindre aquest resultats que considerem satisfactoris, però sí que cal dir que el temps mitjà per consulta ronda el **20-25 segons**, un temps massa elevat per poder posar el projecte en producció en l'estat actual.

En conclusió, aquest SRI creat ha sigut capaç de recuperar informació rellevant en gran part de les consultes, un SRI creat amb les últimes ferramentes i models més potents, com hem demostrat. Un SRI capaç de distingir entre el català i el castellà en gran part. A continuació unes estadístiques sobre el seu rendiment.

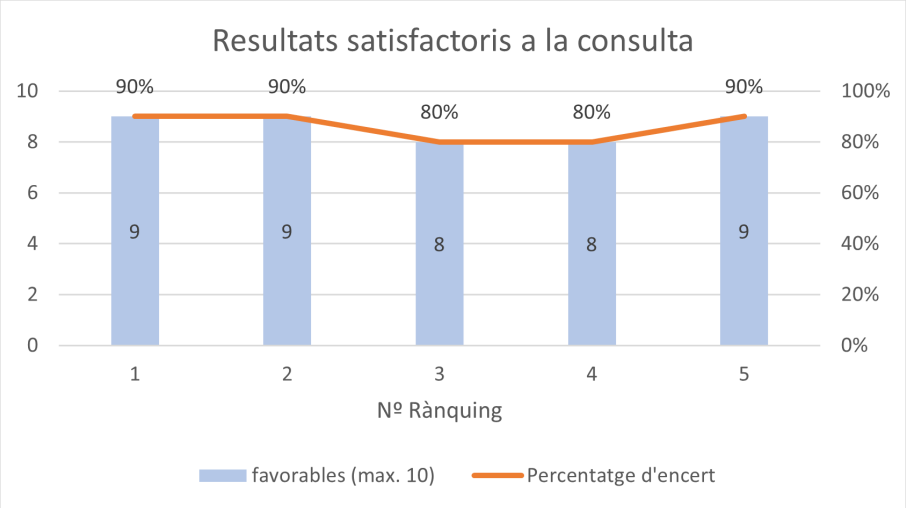


Figura 7.1: Resultats satisfactoris per posició en el rànquing

Aquesta gràfica mostra els encerts, resultats favorables, en cada posició pel nombre de consultes. És a dir, en la primera posició del rànquing quants documents eren rellevants i coherents. En observar la gràfica, ens n'adonem com quasi sempre els documents recuperats són rellevants, no importa tant la posició, donat que les diferències són mínimes i subjectives, però donant prioritat a les primeres posicions. Així i tot, trobem un SRI que en les 5 primeres posicions retorna documents rellevants en alt percentatge.

CAPÍTOL 8
Conclusions

En aquest projecte hem aconseguit dissenyar, desenvolupar i avaluar un Sistema de Recuperació d'Informació basat en representacions vectorials denses (*embeddings*) contextuals. Per tant, hem aconseguit assolir l'objectiu principal que ens havíem marcat per a aquest projecte.

L'objectiu principal l'hem subdividit en subobjectius o fases que hem anat assolint.

Una primera fase que ha consistit en l'estudi de les diferents tecnologies, ferramentes, models, teoria, etc. On hem aprés tots els coneixements necessaris per a poder desenvolupar aquest projecte, en particular els diferents models (SBert, MPNet, XLNet), diferents llibreries i frameworks (Bier, Haytstack). Parlem de coneixements durs d'aprendre, donat que són les últimes tecnologies i models que han eixit, i tenen una profunditat teòrica gran. Aquesta primera etapa d'estudi m'ha ajudat a aprofundir en els meus coneixements teòrics dels SRI, sobretot en la part de la seua avaluació.

Una segona fase d'implementacions i experimentacions, la qual dividirem en dues parts. En la primera part l'avaluació dels diferents models de creació de representacions vectorials actuals i un SRI. Aquestes implementacions no han sigut fàcils. El sistema per avaluar models ha sigut creat seguint les directrius i exemples de Beir, amb les que hem hagut de treballar per poder adaptar els seus scripts a les nostres dades. Aquest sistema utilitza les mesures estudiades d'avaluació per poder comparar els diferents models i dotar-nos d'aquell que millor funcione. Després d'un anàlisis exhaustiu, vam poder concloure que el model que més s'adaptava a les nostres necessitats era el model **sentence-transformers/multi-qa-mpnet-base-dot-v1**. Aquesta fase també implicava l'objectiu d'entrenar un model dens, aquest objectiu no es va poder assolir per la fallada en l'entrenament del model.

En la segona part avaluarem l'ús de dos índexs diferents en Faiss (Flat i IVF-PQ) i l'ús de grandàries distintes de corpus a indexar. Aquesta experimentació ens aportava les dades de cada índex en les diferents etapes que necessitem per a la construcció d'un SRI i els resultats davant diferents consultes. Facilitant-nos la decisió de quin índex utilitzar i dotar-nos de la capacitat de solucionar algun problema abans de la construcció de l'índex final. Finalment, ens decidirem per utilitzar un índex Flat per maximitzar la precisió a l'hora de la recuperació.

En l'última fase implementarem el Sistema de Recuperació d'Informació, el qual utilitza tecnologies de l'estat de l'art tals com: models de similitud semàntica basats en arquitectura Transformers o Faiss per la cerca de documents per similitud semàntica, amb els quals aconseguim un resultats més que satisfactoris.

Personalment, aquest projecte ha suposat un repte molt gran, he hagut d'aprofundir en molts coneixements i estudiar molta teoria nova que ha sigut imprescindible per al

seu desenvolupament. No sols ha millorat els meus coneixements teòrics, sinó que també m'ha ajudat a millor les meues habilitats de programador i investigador. Les meues habilitats i coneixements de les llibreries punteres en l'àmbit del PLN s'han vist augmentades i molt. Hores davant de les documentacions buscant com poder resoldre un problema o buscant en la web ajuda en fòrums o altres, ajudant-me a desenvolupar aquestes habilitats tan necessàries en els món dels informàtics. No sols això, sinó que he descobert el gran món que és el PLN i la gran comunitat que el rodeja.

CAPÍTOL 9
Treball Futurs

En aquest últim capital, una vegada finalitzat el projecte, parlarem de possibles aportacions que es poden implementar, per millorar certs aspectes del SRI. Algunes de les millores podrien ser:

- Ús d'una base de dades gran i completa, com pot ser Postgress, que permet l'escriptura de tot el corpus complet.
- Creació d'un servei en xarxa per utilitzar aquest SRI creat. Amb la seua interfície i processos de recuperació i indexació.
- L'avaluació d'altres tipus d'índexs per augmentar la rapidesa a l'hora de la recuperació.
- Resoldre el problema amb els procés d'entrenament, per utilitzar un model més ajustat a les nostres dades.

Encara que el projecte ha sigut satisfactori i hem obtingut bons resultats, sempre hi ha millores a implementar. Un dels problemes més grans ha sigut el de no poder utilitzar el nostre corpus sencer, i creem que un altre treball futur consistiria a afrontar el repte de la grandària del corpus. Cal remarcar, que aquest projecte està contextualitzat dins d'un altre projecte més gran, on l'objectiu és utilitzar aquest SRI desenvolupat per ser el motor de cerca d'un prototip d'anàlisi i catalogació de continguts multimèdia.

Bibliografia

[1] Salton , G. and Mc Gill, M.J. Introduction to Modern Information Retrieval. *Mc Graw-Hill Computer Series*, New York, 1983.

[2] WANG, S. Toward a general model for web-based information systems. *International Journal of Information Management 21*,2001. p. 385–396

[3] Dominich, S. A unified mathematical definition of classical information retrieval. *Journal of the American Society for Information Science*, 51 (7), 2000. p. 614-624.

[4] Ido Dagan, Oren Glickman, Bernardo Magnini The PASCAL Recognising Textual Entailment Challenge *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, 2006.

[5] Blair, D.C. *Language and representation in information retrieval*. Amsterdam [etc.]: Elsevier Science Publishers, 1990.

[6] John Feather, Paul Sturges *International Encyclopedia of Information & Library Science*. London: Rotledge, 1997.

[7] Chowdhury, G. G. *Introduction to modern information retrieval*. London: Library Association, 1999

[8] Nielsen, J. *Hypertext and hypermedia.*. Oxford, Academia Press, 1990.

[9] Encarna Segarra, Vicent Ahuir, Lluís-F. Hurtado, José Ángel Gonzalez. *DACSA: A large-scale Dataset for Automatic summarization of Catalan and Spanish newspaper*. Articles proceding of the Annual Conference of the North American Chapter of the Association for Computational Linguistics 2022

[10] Rijsbergen, C.J. *Information Retrieval*. Glasgow, University, 1999. Consultat a 13 Juny de 2023 https://www.dcs.gla.ac.uk/Keith/Preface.html

[11] Baeza-Yates, R. and Ribeiro-Neto, B. Modern information retrieval. New York, ACM Press, Addison-Wesley, 1999 Consultat a 14 Juny de 2023 https://web.cs.ucla.edu/~miodrag/cs259-security/baeza-yates99modern.pdf

[12] Villena Román, J. Sistemas de Recuperación de Información Universidad Valladolid, Departamento Ingeniería Sistemas Telemáticos Consultat a 17 Juny de 2023 http://www.mat.upm.es/~jmg/doct00/RecupInfo.pdf

[13] Mikolov, T., Chen, K., Corrado, G. y Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space, pp. 1–12. Consultat a 11 Juny de 2023 http://arxiv.org/abs/1301.3781

[14] Nils Reimers, Iryna Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks (2019) Consultat a 11 Juny de 2023 https://arxiv.org/abs/1908.10084

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention Is All You Need (2017) Consultat a 11 Juny de 2023 https://arxiv.org/abs/1706.03762

[16] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le, XLNet: Generalized Autoregressive Pretraining for Language Understanding (2019) Consultat a 12 Juny de 2023 https://arxiv.org/abs/1906.08237

[17] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, Tie-Yan Liu, MPNet: Masked and Permuted Pre-training for Language Understanding (2020) Consultat a 14 Juny de 2023 https://arxiv.org/pdf/2004.09297.pdf

[18] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel Bowman, GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding (2019) Consultat a 29 Juny de 2023 https://aclanthology.org/W18-5446.pdf

[19] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments (2019) Consultat a 29 Juny de 2023 https://arxiv.org/pdf/1805.12471.pdf

[20] Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng and Christopher Potts, Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank Consultat a 29 Juny de 2023 https://aclanthology.org/D13-1170.pdf

[21] William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. (2005) Consultat a 29 Juny de 2023 https://aclanthology.org/I05-5002.pdf

[22] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, Lucia Specia. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. (2017) Consultat a 29 Juny de 2023 https://aclanthology.org/S17-2001.pdf

[23] Adina Williams, Nikita Nangia, Samuel Bowman. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. (2018) Consultat a 29 Juny de 2023 https://aclanthology.org/N18-1101.pdf

[24] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text Consultat a 29 Juny de 2023 https://aclanthology.org/D16-1264.pdf

[25] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The Winograd Schema Challenge. (2012) Consultat a 29 Juny de 2023 https://cdn.aaai.org/ocs/4492/4492-21843-1-PB.pdf

[26] Ehsan Kamalloo and Nandan Thakur and Carlos Lassance and Xueguang Ma and Jheng-Hong Yang and Jimmy Lin. Resources for Brewing BEIR: Reproducible Reference Models and an Official Leaderboard. (2023) Consultat a 11 Maig de 2023 https://github.com/beir-cellar/beir

[27] Deepset. Haystack Consultat a 5 Maig de 2023 https://docs.haystack.deepset.ai/docs

[28] Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, Rodrigo Nogueira. mMARCO: A Multilingual Version of the MS MARCO Passage Ranking Dataset. (2022) Consultat a 2 Juliol de 2023 https://arxiv.org/abs/2108.13897

[29] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, Tong Wang. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset Consultat a 2 Juliol de 2023 https://arxiv.org/abs/1611.09268

[30] Text Mining Unit (TeMU) at the Barcelona Supercomputing Center PlanTL Project's Spanish-Catalan machine translation model Consultat a 17 Juny de 2023 https://huggingface.co/PlanTL-GOB-ES/mt-plantl-es-ca

[31] OpenNMT, Tokenizer Consultat a 17 Juny de 2023 https://github.com/OpenNMT/Tokenizer

[32] OpenNMT, CTranslate2 Consultat a 17 Juny de 2023 https://github.com/OpenNMT/CTranslate2

[33] Hugging Face, Hugging Face, Inc Consultat a 22 Maig de 2023 https://huggingface.co/

[34] ElasticSearch Consultat a 12 Juliol de 2023 https://www.elastic.co/es/what-is/elasticsearch

[35] Milvus Consultat a 12 Juliol de 2023 https://milvus.io/docs/overview.md

[36] OpenSearch Consultat a 19 Juliol de 2023 https://opensearch.org/

[37] Pinecode Consultat a 17 Juliol de 2023 https://docs.pinecone.io/docs/overview

[38] Weaviate Consultat a 17 Juliol de 2023 https://weaviate.io/

[39] SBERT.net, Sentence-Transformers Consultat a 19 Juliol de 2023 https://www.sbert.net/index.html

[40] Blair *Searching bases in large interactive document retrieval systems*. Journal of the American Society for Information Science 1980 (31) 4 p. 271-277

[41] Saracevic, T. Relevance: A review of and a framework for the thinking on the notion in information science *En Readings in Information Science edited by Karen Spark Jones, Peter Willet*, San Francisco: Morgan Kaufmann Publisher 1997.

[42] Barry, C.L. *User-defined Relevance Criteria: An Exploratory Study Journal of the American Society for Information Science*. 1994 45 (3) p. 149-159.

[43] Korfhage, R. *Information Storage and Retrieval*. New York.: John Wiley, 1997.

[44] Keen, E.M. *Evaluation parameters. The SMART retrieval system Experimentes in automatic document processing*. New Jersey:Prentice-Hall, 1971 p 74-111.

[45] Saracevic, T. *A study of information seeking and retrieving, background and metodology. Journal of the American Society for Information Science*. 39 (3) p. 161-176

[46] Cleverdon, C.W. The Significance of The Cranfield Tests on Index Languages *Bookstein, editor, Proceedings of the 14 th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Chicaco, Illinois, USA, October 1991.

[47] Harman, D. Overview of the Third Text Retrieval Conference (TREC-3) Consultat a 5 Juliol de 2023 trec.nist.gov/pubs/trec3/t3_proceedings.html

[48] Frakes, W. B. and Baeza Yates, R. Retrieval: data structures and Algorithms. Mexico: Prentice-Hall, 1992

[49] Boyce, B. Beyond Topically: A two storage view of relevance and retrieval process Information procesing and Management. 1992 18 p. 105-109

[50] Faiss, FacebookResearch Consultat a 1 Juliol de 2023 https://github.com/facebookresearch/faiss

[51] FARM, deepset.ai Consultat a 1 Juliol de 2023 https://github.com/deepset-ai/FARM/tree/a11ea46a71d708aa48947a640daa30283dd7289c

[52] Haystack, Github wiki Consultat a 12 Juny de 2023 https://github.com/deepset-ai/haystack/tree/main/haystack

[53] Shijie Wu, Mark Dredze Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT *Department of Computer Science, Johns Hopkins University* https://arxiv.org/pdf/1904.09077.pdf

[54] G. salton, A. Wong, C. S. Yang A vectors Space Model for Automatic Indexing *Information retrieval and Language Processing C.A. Montgomery Editor*

[55] J. Devlin, Ming-Wei Chang, Kenton Lee, K. Toutanova BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding *Communications of the ACM, vol. 18, nr. 11, pages 613–620.*

1 OBJECTIUS DE DESENVOLUPAMENT SOSTENIBLE

Grau de relació del treball amb els Objectius de Desenvolupament Sostenible (ODS).

Objectius de Desenvolupament Sostenible	**Alt**	**Mitja**	**Baix**	**No procedeix**
ODS 1. **Fi de la pobresa.**				X
ODS 2. **Fam cero.**				X
ODS 3. **Salut i benestar.**				
ODS 4. **Educació de qualitat.**	X			
ODS 5. **Igualtat de gènere.**				X
ODS 6. **Aigua neta i sanejament.**				X
ODS 7. **Energia assequible i no contaminant.**				
ODS 8. **Treball decent i creixement econòmic.**				X
ODS 9. **Indústria, innovació i infraestructures.**	X			
ODS 10. **Reducció de les desigualtats.**	X			
ODS 11. **Ciutats i comunitats sostenibles.**				X
ODS 12. **Producció i consum responsable.**				X
ODS 13. **Acció pel clima.**				X
ODS 14. **Vida submarina.**				X
ODS 15. **Vida d'ecosistemes terrestres.**				X
ODS 16. **Pau, justícia i institucions sòlides.**				X
ODS 17. **Aliances per aconseguir objectius.**	X			

Reflexió sobre la relació del TFM amb els ODS i amb el/els ODS mes relacionats.

Aquest projecte forma part d'un projecte més gran integrat per diferents universitats, on l'objectiu és construir un prototip competitiu per l'anàlisi afectiva d'informació multimèdia. Aquest prototip és un sistema capaç de recuperar informació de xarxes socials, notícies, etc. El nostre projecte consisteix en la construcció d'un Sistema de Recuperació d'Informació utilitzant les tecnologies punteres disponibles, que el seu treball en el projecte comunitari serà el de motor de cerca. Aquest projecte està desenvolupat per a dos idiomes: català i castellà.

Aquest projecte es relaciona amb diferents ODS:

- ODS 4 / ODS 10: El desenvolupament d'aquest projecte permetrà l'ús d'aquest SRI en les institucions públiques, col·legis, i altres. Aconseguint una reducció de la desigualtat de les llengües minoritàries com el català, donat que el SRI funciona tant per al castellà com per al català. Aleshores obtenim una ferramenta que equipara els dos idiomes en termes de rendiment. També el desenvolupament del SRI permetrà una millora en la qualitat de l'educació en català. Ja que, permetrà la recuperació d'informació de milers de documents, articles, llibres o altres en català, ajudant als estudiants i professorat a trobar la informació requerida per al seu aprenentatge o treball.
- ODS 9: El desenvolupament d'aquest projecte utilitza les tecnologies disponibles més punteres, assolint una nova eina d'innovació tecnològica.
- ODS 17: Com hem explicat, aquest projecte forma part d'un altre més gran. Per tant, diferents universitats han format una aliança per desenvolupar un projecte innovador i que ajudara a universitats i institucions.