Rapport de TER

Participation à ReNeuIR@SIGIR24

Nouh CHELGHAM

Université Toulouse III Paul Sabatier

MASTER 1 IAFA, SEMESTRE 8

Encadrant : José Moreno

Année 2023/2024

Contents

1 Introducion

L'intelligence artificielle a révolutionné diverses industries, en particulier grâce à ses applications en recherche d'information (RI). La recherche d'information se concentre sur l'extraction d'informations pertinentes à partir de vastes ensembles de données textuelles, un processus qui inclut trois phases essentielles : l'indexation, la recherche et le reclassement. C'est dans ce contexte que s'est déroulé mon TER, qui est l'objet de ce rapport. Ce 'TER' fait partie de l'unité d'enseignement Stage/TER, proposée dans le cadre du programme de Master 1 en Intelligence Artificielle, fondements et applications (IAFA) à l'Université Paul Sabatier de Toulouse, durant l'année universitaire 2023/2024

1.1 Contexte

Ce projet de TER (Travaux d'Études et de Recherche) représente une opportunité unique de s'immerger dans le domaine de la recherche d'information (RI). Initié par le Professeur José Moreno, membre de l'équipe IRIS — une équipe de recherche d'information au laboratoire IRIT — et professeur à l'Université Paul Sabatier, ce projet a été spécialement conçu pour les étudiants qui n'ont pas obtenu de stage traditionnel, leur permettant de travailler en binôme et d'acquérir une expérience de recherche précieuse.

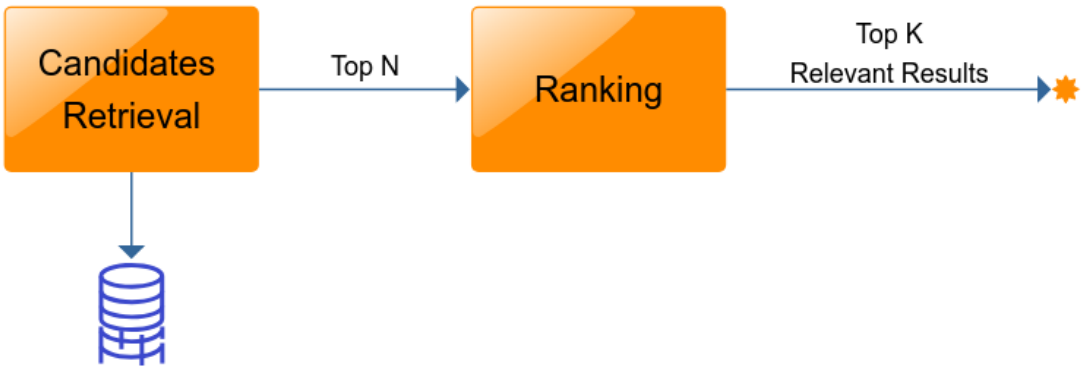
le but de ce projet est de participer à l'atelier ReNeuIR, qui fait partie de la prestigieuse conférence SIGIR (Special Interest Group on Information Retrieval), organisée par l'ACM. Lors de cet atelier, nous devons créer un système de recherche d'information neuronal (NIR). SIGIR[1] est un événement de premier plan pour les chercheurs et professionnels en recherche d'information, couvrant des sujets tels que les algorithmes de recherche, le traitement du langage naturel (NLP), l'apprentissage automatique, et les applications de la recherche d'information. La conférence attire des participants du monde entier et offre une plateforme pour présenter les dernières avancées et échanger des idées, tout en étant un lieu important pour la publication de recherches de haute qualité.

1.2 Objectif

Dans ce projet, nous proposons de développer un pipeline de Neural Information Retrieval (NIR) en utilisant des modèles de langage pré-entraînés. Notre objectif est de créer un système de recherche d'informations capable de comprendre et de répondre efficacement aux requêtes des utilisateurs, en exploitant les avancées récentes en représentation sémantique et en traitement du langage naturel.

Nous commencerons par une revue des modèles de langage pré-entraînés (comme BERT, RoBERTa..ect) et des techniques de NIR existantes. Ensuite, nous décrirons en détail les étapes de notre pipeline, incluant la collecte et la préparation des données, l'encodage des requêtes et des documents, le calcul de la similarité, et l'évaluation des performances du modèle. Nous présenterons également des résultats expérimentaux et des comparaisons avec d'autres approches de NIR pour illustrer l'efficacité de notre méthode.

Enfin, à la fin de ce projet, il nous sera demandé de créer une image Docker qui générera un fichier RUN.txt au format TREC. Cette image sera soumise sur le site de la conférence TIRA[3] pour être évaluée. Nous espérons que ce projet contribuera à l'avancement de la recherche dans le domaine de la NIR et apportera des solutions concrètes pour améliorer la pertinence et l'efficacité des systèmes de recherche d'informations.



2 Laboratoire Et Organisation

2.1 IRIT

Institut de Recherche
en Informatique de Toulouse
CNRS - INP - UT3 - UT1 - UT2J

L'IRIT (Institut de Recherche en Informatique de Toulouse)[2] est un laboratoire de recherche composé de plusieurs équipes, dont la structure est organisée autour des domaines suivants :

- Conception et construction de systèmes (fiables, sûrs, adaptatifs, distribués, communicants, dynamiques, etc.).
- Modélisation numérique du monde réel.
- Concepts pour la cognition et l'interaction.
- Étude des systèmes autonomes adaptatifs à leur environnement.
- Transformation des données brutes en informations intelligibles.

L'IRIT est structuré en sept départements différents , regroupant 24 équipes de recherche qui couvrent divers domaines de l'informatique, tels que la ville intelligente, la santé, l'aéronautique, la cybersécurité, etc. Ce projet de recherche a été realisé au sein du département de l'Intelligence Artificielle de l'IRIT.

Fondé en 1990 en partenariat entre l'Université Paul Sabatier et le CNRS (Centre National de la Recherche Scientifique), l'IRIT est reconnu comme l'un des instituts les plus renommés en Europe et dans le monde, grâce à sa participation dans de nombreux projets scientifiques.

2.2 IRIS

Ce TER m'a indirectement mis en contact avec l'une des équipes de recherche de l'IRIT, l'équipe IRIS[4], dirigée par Monsieur Gilles HUBERT. Cette équipe se concentre principalement sur deux grands thèmes de recherche :

- Recherche d'Information (RI) : L'équipe développe des modèles pour résoudre des défis complexes de recherche d'information, notamment dans des contextes contextualisés et avec des sources hétérogènes et dynamiques telles que les médias sociaux. Ils explorent également des approches collaboratives pour améliorer la recherche d'informations.

- Synthèse d'Information (SI) : L'équipe se concentre sur la création d'informations à haute valeur ajoutée. Ils conçoivent des modèles pour agréger des informations pertinentes provenant de diverses sources hétérogènes, telles que le Web. De plus, ils travaillent sur l'analyse de graphes dynamiques pour comprendre et prédire les relations entre les entités du monde réel.

2.3 Organisation

Le début de notre Travail d'Étude et de Recherche (TER) a eu lieu le 17 mai, date à laquelle Monsieur Moreno nous a convoqués pour la première fois à l'IRIT afin de nous fournir des détails supplémentaires sur notre sujet. Ma collègue Alexandra et moi-même étions convoqués chaque vendredi pour une réunion de 15 minutes sur Discord. Tous les trois semaines, nous avions également une réunion de 45 minutes, qui pouvait se tenir soit sur Discord, soit en présentiel à l'IRIT. Les réunions de 15 minutes avaient pour objectif de confirmer que nous progressions bien et que nous n'avions pas rencontré de problèmes majeurs. Les réunions de 45 minutes, quant à elles, étaient destinées à présenter notre avancement à notre tuteur, José Moreno, afin qu'il puisse nous guider et nous prodiguer des conseils.

En parallèle, Alexandra et moi-même organisions des appels sur Discord au moins deux fois par semaine pour discuter de nos avancements individuels et voir comment nous pouvions nous entraider. Un dépôt GitHub contenant nos programmes a été créé et partagé avec Monsieur Moreno. De plus, un Google Drive contenant un document Google Docs intitulé "Suivi" et un fichier Excel intitulé "Résultats" a également été partagé avec notre tuteur. Ces outils nous ont permis de suivre notre progression de manière structurée et de partager nos résultats de manière efficace.

2.4 Récuperation des données

Pour commencer la construction de notre pipeline, nous avons d'abord installé plusieurs packages sur notre machine, y compris Tira. Cela nous a permis d'utiliser la commande tira-cli –download , qui télécharge le datasetID localement dans le répertoire root . Ce répertoire contient un sous-dossier nommé "input-data", qui inclut les fichiers suivants : queries.jsonl, qrels.txt, et documents.jsonl.gz.

Dans notre cas, nous avons téléchargé les datasets "dl-1000-top-docs" et "dl-10-top-docs". Le deuxième dataset, étant de taille et de nombre de documents plus réduits, a été utilisé comme test pour vérifier la validité syntaxique de nos scripts surtout qu'il est exactement comme le premier en termes de queries(97) sauf que dl-10-top-docs contient 6095 documents alors que dl-1000-top-docs contient 523409 documents et il est considéré comme le dataset principale pour verifier l'efficacité et l'efficience de l'ensemble des pipelines.

3 Recherches Et Tests

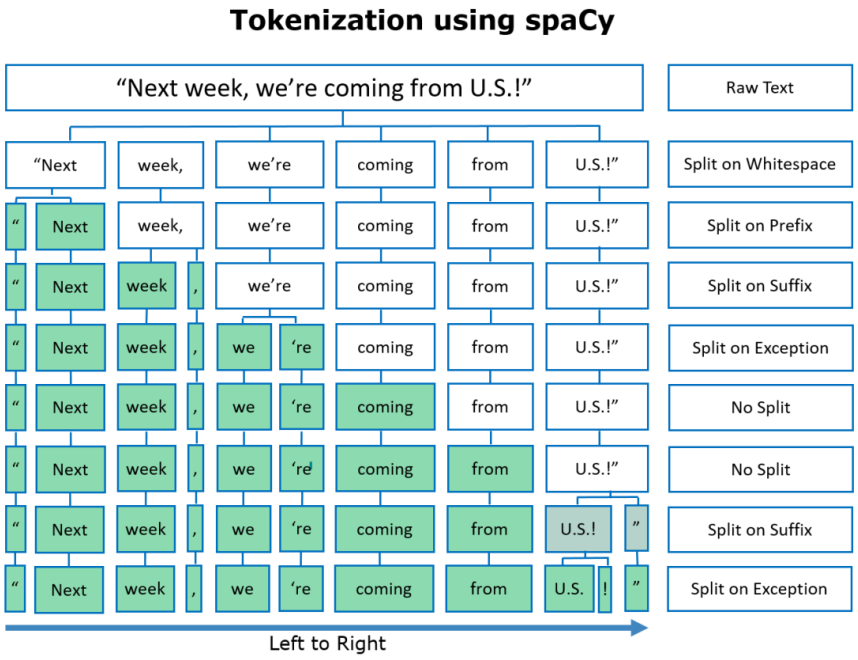
3.1 Indexation

Vu que la premiere étape de la construction de ce pipeline de recherche de l'information est l'indexation, on a commencé nos recherches avec cette partie la.L'indexation est une étape cruciale, Elle consiste à organiser et structurer les données pour permettre une recherche rapide et efficace. elle améliore la performance des requêtes en réduisant le temps nécessaire pour localiser les documents pertinents. Elle optimise également l'utilisation des ressources informatiques et contribue à fournir des résultats de recherche plus précis et pertinents. Dans des environnements où la rapidité et l'efficacité sont essentielles, comme les moteurs de recherche web et les bibliothèques numériques, l'indexation joue un rôle central.

3.1.1 SpaCy

L'intégration de SpaCy[5] dans le processus d'indexation apporte des avantages notables. En tant qu'outil reconnu dans le domaine du traitement du langage naturel (NLP), SpaCy fournit des fonctionnalités avancées pour le prétraitement du texte, essentielles à une indexation efficace. Parmi ces fonctionnalités, la tokenisation permet de segmenter le texte en unités exploitables, tandis que la lemmatisation unifie les variations d'un mot sous une forme de base commune, optimisant ainsi la précision des recherches. Par exemple, les variantes 'running' et 'ran' sont toutes deux indexées sous 'run', augmentant la cohérence des résultats.

Au-delà de ses capacités techniques, SpaCy se distingue par sa rapidité et son efficacité, des atouts qui renforcent la performance globale des systèmes de recherche. De plus, sa compatibilité avec d'autres outils de recherche d'information facilite la création de pipelines de traitement robustes, où SpaCy intervient en amont pour assurer un prétraitement de qualité avant l'indexation des documents. C'est pour ces raisons que j'ai décidé de l'implémenter, afin d'améliorer la pertinence et la performance de notre pipeline.



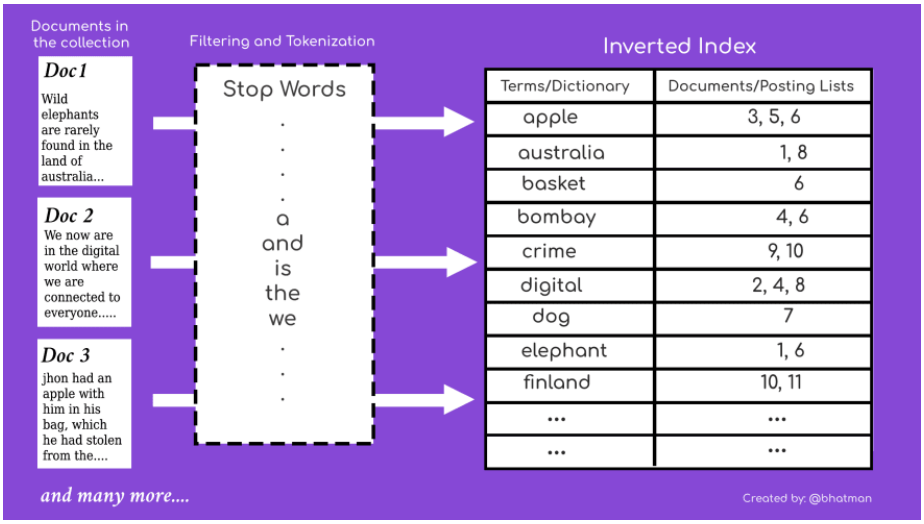
3.1.2 Anserini

Lors de mes recherches sur la meilleure façon d'indexer, je suis tombé sur Anserini[7], une bibliothèque open-source dédiée à la recherche d'information qui offre des outils puissants pour l'indexation et la recherche de documents textuels. Basée sur le moteur de recherche Lucene[6], Anserini garantit une indexation efficace et flexible, facilitant ainsi une recherche rapide et précise dans de grandes collections de documents. Grâce à Lucene, l'indexation avec Anserini est particulièrement performante, assurant des performances élevées et une gestion optimale des ressources. Les utilisateurs peuvent facilement créer et gérer des index à l'aide de commandes simples et intuitives. Anserini supporte divers formats de données, ce qui permet l'indexation de textes, fichiers JSON et autres collections de données standard. La bibliothèque offre également des fonctionnalités avancées pour l'indexation, comme la personnalisation des pipelines et l'intégration de modèles de recherche basés sur l'apprentissage automatique. Par exemple, il est possible d'utiliser des modèles de représentation de texte comme BERT pour enrichir les données avant l'indexation, améliorant ainsi la pertinence des résultats de recherche. De plus, Anserini est extensible, permettant aux utilisateurs d'ajouter des fonctionnalités et de personnaliser les processus d'indexation en fonction de leurs besoins spécifiques. Ce faisant, elle constitue un outil précieux pour les chercheurs et les développeurs souhaitant expérimenter et évaluer différentes techniques de recherche d'information. Toutefois, en raison de problèmes de compatibilité sur ma machine, j'ai dû renoncer à l'utiliser, malgré sa pertinence apparente pour mon projet.

3.1.3 Pyterrier

Après avoir constaté qu'Anserini n'était pas le choix idéal en raison de problèmes d'incompatibilité, j'ai cherché des alternatives permettant l'indexation sans dépendances, c'est-à-dire sans avoir besoin d'une bibliothèque externe. J'ai découvert, sur le GitHub de la conférence, des baselines pour l'indexation, la recherche et le reranking sans dépendances. Initialement, nous avons utilisé un index classique du type Document-based-indexing, mais après avoir testé des méthodes de recherche traditionnelles comme BM25, nous avons constaté que les résultats n'étaient pas vraiment satisfaisants. Nous avons donc décidé de modifier l'index et avons opté pour spaCy. Après l'avoir intégré, nous avons choisi un index INVERSE fusionné avec spaCy, ce qui a conduit à une amélioration significative des résultats. Cependant, j'ai estimé qu'il était encore possible d'obtenir de meilleurs résultats, ce qui m'a conduit à utiliser pyterrier.

Nous avons décidé alors d'explorer pyterrier[8], une bibliothèque qui se distingue particulièrement par ses capacités d'indexation. Pyterrier offre plusieurs types d'indexation adaptés aux besoins divers en recherche d'information, tels que l'indexation de termes (index inversé), l'indexation de fréquence, l'indexation de document, l'indexation par champ, l'indexation à base de modèles, et l'indexation hiérarchique. Ces options permettent une flexibilité considérable dans la gestion des données textuelles et l'optimisation des performances de recherche. Pour notre projet, nous avons opté pour un index inversé, qui est le type d'indexation le plus couramment utilisé. Il permet de créer un index qui associe les termes aux documents dans lesquels ils apparaissent, offrant ainsi une recherche rapide et efficace basée sur les occurrences des termes. En utilisant pyterrier avec un index inversé, nous avons pu bénéficier d'une approche sophistiquée pour l'indexation, ce qui a significativement contribué à l'amélioration des résultats obtenus



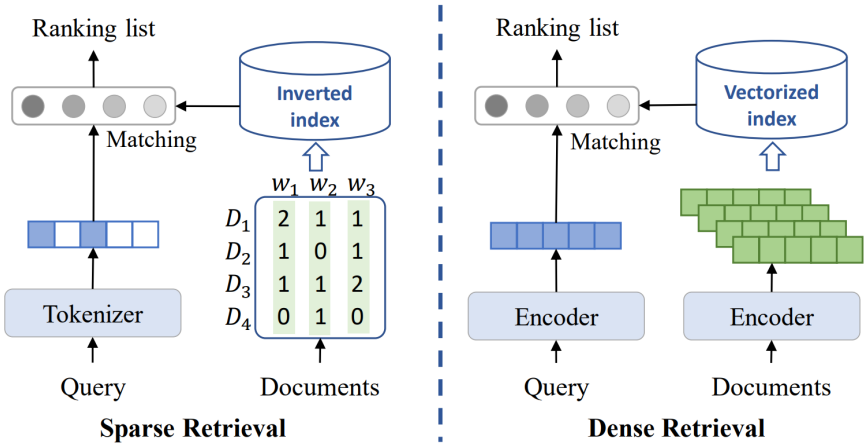
3.2 Retrieve

Dans le domaine de la recherche d'information et du traitement des données, le terme "retrieve" désigne le processus de "récupération" ou "extraction" d'informations. Cela implique de localiser et d'obtenir des données spécifiques à partir de sources comme des bases de données, des moteurs de recherche, ou d'autres systèmes d'information. Par exemple, lors d'une recherche en ligne, le moteur de recherche "récupère" les pages web pertinentes en fonction des mots-clés fournis et les affiche à l'utilisateur.

Par ailleurs, dans ce contexte, les termes "sparse" (clairsemé) et "dense" (dense)[9] sont utilisés pour décrire les différentes formes de représentations de données, telles que celles trouvées dans les vecteurs et matrices. Une représentation "sparse" se caractérise par la majorité de ses éléments étant nuls ou ayant des valeurs par défaut. Cela est typique dans les modèles de sac de mots pour les documents, où chaque document ne contient qu'un petit sous-ensemble des mots du vocabulaire global. À l'inverse, une représentation "dense" contient principalement des valeurs significatives. C'est le cas des vecteurs denses générés par des modèles d'embeddings comme Word2Vec ou BERT. Ces différences dans les représentations de données influencent la manière dont les informations sont récupérées et traitées, impactant ainsi l'efficacité et la précision des systèmes de recherche d'information.

3.2.1 Dense

Une représentation "dense" est un vecteur où la plupart, voire toutes les valeurs, sont non nulles. Cela est typiquement utilisé dans les représentations modernes, comme les Word Embeddings (par exemple, Word2Vec, GloVe) ou les Transformers (comme BERT, GPT). Dans ces modèles, les mots ou les phrases sont représentés par des vecteurs de faible dimension, où chaque valeur capture une caractéristique latente de la sémantique du texte. Ces représentations sont beaucoup plus compactes et capables de capturer des relations sémantiques complexes entre les mots, permettant à des mots similaires de se retrouver dans des zones proches de l'espace vectoriel. Les modèles "dense" sont plus adaptés aux tâches complexes de NLP, comme la recherche sémantique, où il est crucial de comprendre le contexte et la signification des mots. J'ai initialement appliqué cette approche en utilisant les k-plus proches voisins (k-NN) pour rechercher des documents similaires, mais mon tuteur m'a suggéré que cette méthode n'était pas pratique en raison de la consommation de mémoire et du temps de calcul élevés. Il m'a recommandé de passer par une couche sparse d'abord, ce qui permettrait de réduire la dimensionnalité des données et d'améliorer l'efficacité des calculs avant de recourir à des méthodes de recherche basées sur des vecteurs denses et au final , aprés avoir lu plusieurs articles on l'a bien compris que les approache de recherches neureaux font beaucoup de fautes par rapport à des approaches hybrides[19].



3.2.2 Sparse

Une représentation "sparse" désigne un vecteur où la plupart des valeurs sont égales à zéro, et seules quelques-unes sont non nulles. Cela est souvent utilisé dans les méthodes traditionnelles de représentation du texte, comme les Bag of Words (BoW) ou les TF-IDF (Term Frequency-Inverse Document Frequency). Dans ces modèles, chaque mot ou n-gramme est représenté par une dimension unique dans le vecteur. Le vecteur pour chaque document ou phrase contient principalement des zéros, avec quelques positions ayant des valeurs non nulles correspondant aux mots présents dans le document. Les représentations "sparse" sont efficaces pour gérer des données très larges (comme de grands corpus de textes), mais elles peuvent être peu performantes pour capturer des relations sémantiques entre les mots.

Des méthodes comme BM25 exploitent ces représentations pour améliorer la pertinence des résultats de recherche. BM25 évalue la pertinence d'un document par rapport à une requête en se basant sur la fréquence des termes dans une matrice clairsemée. De même, la factorisation matricielle est utilisée dans les systèmes de recommandation pour décomposer une matrice clairsemée d'évaluations en deux matrices plus petites, permettant de prédire les évaluations manquantes. Ces méthodes optimisent le traitement en se concentrant sur les éléments non nuls, tout en réduisant la consommation de mémoire et améliorant la pertinence des résultats. Voici quelques méthodes classiques de retrieve sparse :

- TF-IDF (Term Frequency-Inverse Document Frequency): TF-IDF est une méthode de pondération qui évalue l'importance d'un terme dans un document en se basant sur deux facteurs : la fréquence d'apparition du terme dans le document (TF) et sa rareté à travers tous les documents (IDF). TF-IDF attribue un score plus élevé aux termes qui apparaissent fréquemment dans un document mais rarement dans d'autres documents, ce qui aide à identifier les termes les plus pertinents pour ce document spécifique. Les

documents sont représentés par des vecteurs clairsemés où chaque dimension correspond à un terme du vocabulaire global, et les poids sont calculés en fonction de TF-IDF. Cette méthode permet de souligner l'importance relative des termes pour améliorer la recherche de documents pertinents.

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

- BM25 (Best Matching 25) BM25 est un modèle de pondération de documents qui calcule la pertinence d'un document par rapport à une requête en tenant compte de la fréquence des termes dans le document et de la longueur du document. Le modèle ajuste les poids des termes en fonction de leur fréquence, avec des paramètres de saturation qui limitent l'impact des termes très fréquents, et il normalise les scores en fonction de la longueur du document. Les documents sont représentés par des vecteurs clairsemés avec des poids BM25 pour chaque terme, ce qui rend le modèle efficace pour gérer les variations de longueur des documents et les termes fréquents tout en optimisant la pertinence des résultats de recherche.

$$BM25 = \sum_{t \in q} \log\left[\frac{N}{df(t)}\right] \cdot \frac{(k_1+1)\cdot tf(t,d)}{k_1 \cdot \left[(1-b) + b \cdot \frac{dl(d)}{dl_{avg}}\right] + tf(t,d)}$$

- k_1, b – parameters
- $dl(d)$ – length of document d
- dl_{avg} – average document length

3.3 Ranking

En recherche d'information, le 're-rank' est une technique utilisée pour améliorer la pertinence des résultats de recherche après qu'un classement initial ait été généré. Initialement, une requête de recherche récupère un ensemble de résultats basé sur un algorithme de classement préliminaire. Le processus de ré-rank applique ensuite des méthodes plus sophistiquées pour réordonner ces résultats, visant à présenter les informations les plus pertinentes en haut. Ce processus secondaire peut inclure des modèles d'apprentissage automatique avancés, des informations contextuelles supplémentaires ou des techniques d'expansion de requête pour affiner la pertinence des résultats et améliorer ainsi l'expérience utilisateur en rendant les résultats les plus pertinents plus accessibles. Les techniques de ré-rank peuvent être classées en plusieurs types en fonction de la manière dont elles traitent et réorganisent les résultats de recherche. Voici les principaux types :

3.3.1 Pairwise

Cette approche compare les paires de documents pour déterminer lequel est plus pertinent pour une requête donnée. Le modèle apprend à classer correctement les paires en fonction de leur pertinence relative. Par exemple, si un document A est jugé plus pertinent qu'un document B, le modèle ajuste les scores en conséquence[10].

3.3.2 Listwise

Contrairement à l'approche pairwise, l'approche listwise[11] considère l'ensemble des documents dans une liste pour évaluer leur pertinence. Le modèle optimise l'ensemble du classement plutôt que de comparer des paires de documents. Cela permet de traiter la liste entière de manière plus cohérente et d'améliorer la qualité globale du classement.

3.3.3 Pointwise

Dans cette approche, chaque document est évalué individuellement en fonction de sa pertinence pour une requête spécifique. Le modèle apprend à attribuer un score de pertinence pour chaque document plutôt que de comparer des documents entre eux ou d'optimiser l'ordre des documents dans une liste[12].

3.3.4 Learning to Rank (LTR)

C'est une approche générale qui peut inclure les techniques pairwise, listwise et pointwise. Les modèles d'apprentissage à classer apprennent à

prédire la pertinence des documents en utilisant des caractéristiques extraites des documents et des requêtes. Les méthodes LTR peuvent combiner plusieurs techniques pour améliorer le classement des résultats[13].

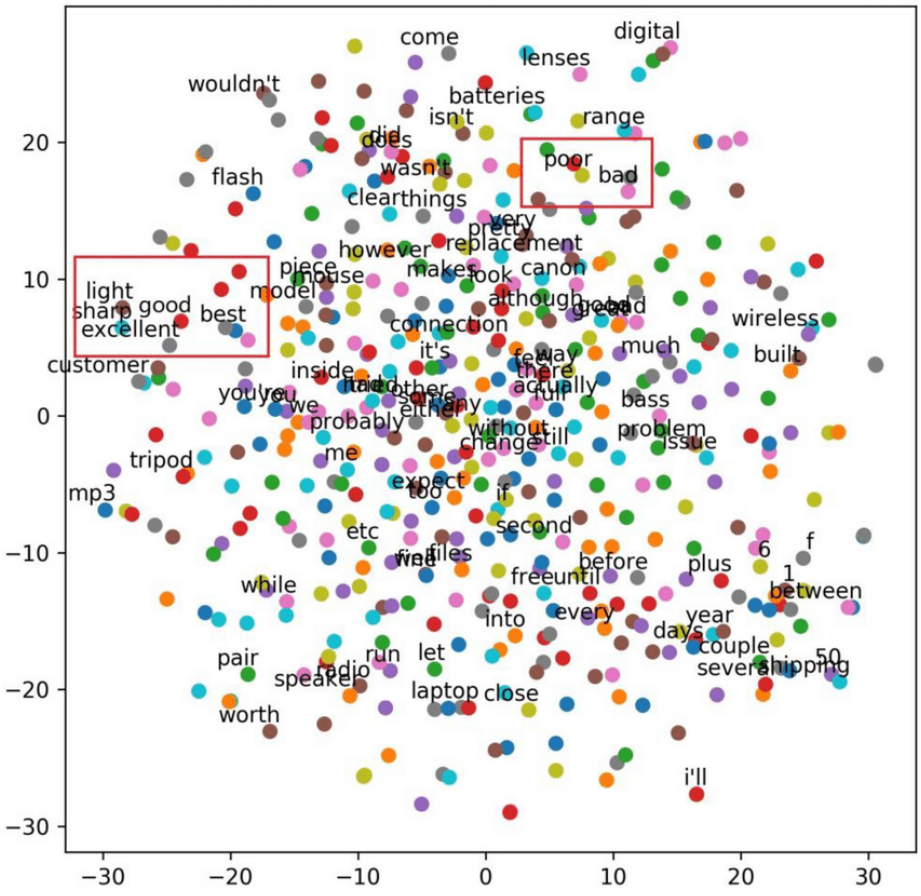
3.3.5 Re-ranking basé sur les caractéristiques

Cette méthode utilise des caractéristiques supplémentaires ou des métadonnées pour réévaluer les résultats initialement obtenus. Par exemple, elle peut intégrer des informations sur les clics des utilisateurs, les interactions passées ou d'autres données contextuelles pour affiner le classement.

4 Ressources Pour la Création Du Pipeline

Les méthodes traditionnelles de recherche d'information (RI) et les approches neurales en RI représentent des paradigmes distincts pour traiter les tâches de recherche et de récupération. Les techniques traditionnelles de RI, telles que le modèle Bag-of-Words (BoW) et le TF-IDF, reposent sur la correspondance des mots-clés et des mesures statistiques pour représenter et classer les documents. Ces méthodes évaluent l'importance des termes dans un document en fonction de leur fréquence et de leur fréquence inverse dans le corpus, offrant une approche directe de la pertinence, mais souvent au détriment de la compréhension contextuelle. Elles utilisent un classement basé sur des heuristiques et sont efficaces pour traiter de grands ensembles de données en raison de leurs exigences computationnelles plus faibles.

En revanche, les approches neurales en RI exploitent des modèles d'apprentissage profond avancés pour saisir la signification sémantique et le contexte. Des techniques telles que les embeddings de mots (Word2Vec, GloVe) et les embeddings contextuels (BERT, GPT) permettent une compréhension plus profonde du texte en générant des représentations denses et continues qui reflètent les significations nuancées des mots et des phrases. Les modèles neuronaux améliorent la compréhension des requêtes en interprétant l'intention et le contexte des requêtes, et ils soutiennent l'expansion et le classement sophistiqués des requêtes grâce aux techniques d'apprentissage pour le classement (LTR). Ces modèles peuvent réordonner les résultats de recherche initiaux en fonction des schémas appris à partir des données de pertinence, améliorant ainsi la précision et la pertinence des résultats.



Cependant, les approches neurales en RI sont gourmandes en ressources, nécessitant une puissance computationnelle significative et du temps pour l'entraînement et l'inférence, et peuvent entraîner une latence et un coût plus élevés. Malgré ces défis, les approches neurales offrent une adaptabilité et une personnalisation supérieures, s'ajustant dynamiquement aux préférences des utilisateurs et aux contenus évolutifs.

L'objectif de cette conférence est de mettre en avant l'efficacité des approches neurales en RI, en soulignant leurs avancées par rapport aux méthodes traditionnelles. Par conséquent, nous nous concentrerons sur l'exploration des capacités des techniques neurales en RI, en mettant en évidence leurs améliorations significatives en termes de compréhension sémantique et de prédiction de pertinence. Voici les principales approches neurales que nous allons examiner :

- Cross-Encoder-Ranker :

L'approche du cross-encoder-ranker utilise des modèles de transformateurs pour encoder simultanément les requêtes et les documents. En traitant les deux ensembles ensemble, le modèle évalue directement la pertinence de chaque paire requête-document, permettant ainsi une meilleure compréhension contextuelle et des classements plus précis. Par exemple, un modèle comme BERT peut être utilisé pour effectuer cette tâche en évaluant la correspondance entre la requête et le document en une seule étape.

- LLM-Cross-Encoder-Ranker :

  Le LLM Cross-Encoder-Ranker est une variante du Cross-Encoder-Ranker qui utilise des modèles de langage à grande échelle, tels que GPT ou BERT. Ces modèles capturent des relations complexes et des nuances subtiles dans le texte, permettant d'obtenir des classements encore plus contextualisés et précis

- LLM-Expansions :

  Le LLM Expansions utilise des modèles de langage pré-entraînés, tels que T5, pour enrichir les requêtes avec des termes ou phrases connexes. Cette technique améliore la couverture des résultats de recherche en ajoutant des termes sémantiquement liés, augmentant ainsi la pertinence des documents récupérés.

- LLM-Expansion-Retriever :

  Le LLM Expansion Retriever combine les techniques de LLM-Expansion avec des méthodes de récupération d'information. Après avoir enrichi les requêtes avec des termes supplémentaires, cette approche utilise les requêtes étendues pour retrouver des documents pertinents, permettant ainsi une recherche plus complète et précise. Un exemple de modèle utilisé dans cette approche est RoBERTa (Robustly optimized BERT approach).

- Bi-Encoder-Retriever :

  Le Bi-Encoder-Retriever utilise des architectures de transformateurs pour encoder les requêtes et les documents en vecteurs denses distincts. La similarité entre ces vecteurs est ensuite calculée pour extraire les documents les plus pertinents, offrant ainsi une méthode efficace pour le classement initial basé sur des représentations contextuelles. Un exemple de modèle couramment utilisé pour cette approche est DistilBERT.

4.1 PLMs

Les modèles de langage pré-entraînés[14](PLM : Pre-trained Language Models) ont marqué une avancée révolutionnaire dans le domaine du traitement du langage naturel (NLP), offrant des capacités exceptionnelles pour comprendre, générer et manipuler le texte de manière plus

sophistiquée que jamais auparavant. Ces modèles sont d'abord formés sur d'énormes corpus de texte de manière non supervisée, ce qui leur permet d'apprendre les structures profondes et les nuances du langage humain en capturant des relations sémantiques et syntaxiques complexes. Par exemple, des modèles comme BERT et GPT utilisent des techniques avancées telles que la prédiction de mots masqués et la génération de texte contextuelle pour acquérir une compréhension riche du langage. Une fois pré-entraînés, ces modèles peuvent être ajustés finement pour des tâches spécifiques telles que la classification de texte, la traduction automatique, et la réponse aux questions, en utilisant des ensembles de données annotés qui permettent d'adapter les modèles à des besoins particuliers. Cette approche d'apprentissage transférable est extrêmement efficace, car elle permet d'utiliser les connaissances acquises durant le pré-entraînement pour améliorer les performances sur des tâches spécifiques tout en réduisant le besoin de grandes quantités de données annotées. Cependant, les PLM ne sont pas sans défis : leur entraînement nécessite des ressources computationnelles considérables et peut entraîner des biais si les données d'entraînement contiennent des préjugés. De plus, la complexité de ces modèles pose des problèmes d'interprétabilité, rendant difficile la compréhension de leurs décisions internes. Malgré ces défis, les PLM continuent de transformer le paysage du traitement du langage naturel, offrant des solutions puissantes et flexibles qui ouvrent la voie à de nouvelles applications innovantes et à des améliorations significatives dans de nombreux domaines de la technologie linguistique.

4.2 Huggingface

Hugging Face[15] est une entreprise et une communauté largement reconnue pour son engagement dans le développement et la diffusion des technologies de traitement du langage naturel (NLP) et de l'intelligence artificielle (IA). Fondée en 2016, Hugging Face s'est d'abord fait connaître par ses contributions à la création de modèles de langage pré-entraînés, tels que les célèbres transformateurs BERT, GPT-2, GPT-3 et bien d'autres, accessibles via leur bibliothèque Transformers. Cette bibliothèque open source permet aux chercheurs, ingénieurs et développeurs d'utiliser et de fine-tuner des modèles de pointe en NLP avec une facilité remarquable. En

plus de Transformers, Hugging Face propose une plateforme en ligne qui héberge des modèles et des datasets, facilitant le partage et la collaboration dans la communauté IA. La plateforme Hugging Face Hub offre un accès centralisé à des milliers de modèles pré-entraînés et à des jeux de données, tout en permettant aux utilisateurs de stocker et de partager leurs propres contributions. Hugging Face s'investit également dans des initiatives telles que la démocratisation de l'IA responsable et l'éthique en intelligence artificielle, tout en continuant à étendre ses outils pour les applications de vision par ordinateur, de génération de texte, de compréhension du langage et plus encore. Grâce à son écosystème intégré et sa communauté active, Hugging Face joue un rôle crucial dans l'évolution des technologies NLP, en rendant les avancées en IA plus accessibles et en facilitant leur adoption à grande échelle.

4.3 Transformers library

State-of-the-art Natural Language Processing for PyTorch and TensorFlow 2.0

La bibliothèque Transformers de Hugging Face marque une avancée significative dans le traitement du langage naturel (NLP) en offrant une plateforme intégrée pour travailler avec des modèles de langage basés sur des transformateurs. Cette bibliothèque permet de gérer facilement des modèles pré-entraînés pour diverses applications telles que la classification, la génération, la traduction automatique, et la réponse aux questions. Elle propose une interface Python cohérente, compatible avec les principaux frameworks de machine learning comme PyTorch et TensorFlow, offrant ainsi une grande flexibilité pour le développement. En outre, Transformers intègre des fonctionnalités avancées pour faciliter la personnalisation et le fine-tuning des modèles. La bibliothèque bénéficie d'une documentation détaillée et est soutenue par une communauté active, ce qui facilite son adoption par des utilisateurs de tous niveaux.

5 Création du pipeline final

Étant donné la renommée de la conférence SIGIR24, nous avons voulu créer notre propre pipeline de recherche depuis le début, ce qui a nécessité presque deux mois de recherches approfondies. Après avoir exploré diverses méthodes et techniques pour développer notre pipeline, nous avons constaté que l'outil mentionné par Monsieur Moreno pourrait s'avérer utile dans notre contexte. En effet, avant le début de ce projet, notre tuteur nous avait parlé d'un outil développé par Renault, conçu pour faciliter l'implémentation de pipelines de recherche neuronaux. Cet outil s'appelle CHERCHE[16]

5.1 Cherche

Cherche est un outil avancé pour le développement de pipelines de recherche neuronaux, offrant une solution complète pour les tâches de recherche et de classement grâce à sa capacité à construire des pipelines de bout en bout. Il excelle notamment dans la recherche sémantique hors ligne en intégrant des outils tels que FAISS (Facebook AI Similarity Search) pour une indexation efficace. FAISS optimise la recherche de vecteurs à grande échelle avec une automatisation simplifiée, nécessitant seulement quelques lignes de code pour initialiser la phase de récupération. Cherche propose aussi une variété de méthodes de récupération, y compris TF-IDF pour évaluer l'importance des termes, Lunr pour des applications web légères, Flash pour des recherches rapides en mémoire, Encoder pour une recherche sémantique fine, DPR (Dense Passage Retrieval) pour

améliorer la récupération contextuelle avec des représentations denses, Fuzz pour gérer les fautes d'orthographe et les variations, et Embedding pour mesurer la similarité sémantique.
En ce qui concerne le re-ranking, Cherche inclut des méthodes telles qu'Encoder, DPR, Cross Encoder pour évaluer conjointement les paires requête-document, et Embedding pour réajuster les scores en fonction des embeddings. De plus, l'outil facilite le passage au calcul GPU, réduisant les temps d'exécution de 15 heures à 2 heures sans nécessiter de modifications de code. Distribué sous la licence MIT, Cherche se distingue par sa flexibilité et sa facilité d'implémentation, rendant le développement de pipelines de recherche sophistiqués à la fois efficace et accessible. Pour la première couche de notre système, nous avons choisi d'utiliser **DPR** comme couche de récupération neuronale initiale, et **Lunr**, qui a montré des performances supérieures à BM25 de PyTerrier lors de nos tests. Pour la deuxième couche, nous avons opté pour **All-mini-lm-6-v2**[17].

5.2 DPR

Dense Passage Retrieval (DPR)[18] est une méthode avancée utilisée dans la recherche d'informations et le traitement du langage naturel pour améliorer de manière significative les tâches de recherche et de récupération. Développé par Facebook AI Research (FAIR), DPR va au-delà des techniques de récupération traditionnelles qui reposent sur des représentations éparses telles que TF-IDF, en utilisant des représentations vectorielles denses pour les requêtes et les documents. Cette approche implique l'encodage des requêtes et des passages en vecteurs denses à l'aide d'encodeurs de réseaux neuronaux séparés, capturant ainsi plus efficacement le sens sémantique. Le modèle est entraîné de bout en bout en utilisant de grands ensembles de données de paires question-réponse, apprenant à aligner les vecteurs des paires pertinentes plus près dans l'espace vectoriel tout en éloignant les vecteurs des paires non pertinentes. Pour la récupération, DPR calcule la similarité entre le vecteur de la requête et les vecteurs des passages, en utilisant des techniques de recherche efficace des plus proches voisins telles que FAISS (Facebook AI Similarity Search) pour identifier rapidement les passages les plus pertinents. Cette méthode est conçue pour gérer des ensembles de données à grande échelle avec une grande efficacité, ce qui la rend adaptée aux applications nécessitant des capacités de recherche rapides et précises à travers des collections de documents étendues. Cependant, nous avons rencontré des problèmes lors de l'utilisation de DPR, notre processus étant tué à chaque fois.

5.3 Lunr

Lunr est un moteur de recherche léger en JavaScript conçu pour intégrer facilement des fonctionnalités de recherche textuelle. Il est particulièrement adapté aux projets de petite à moyenne taille, offrant une solution rapide et simple pour l'indexation et la recherche dans des ensembles de données de taille modeste. Lunr construit un index inversé qui facilite des recherches rapides en utilisant des termes de recherche et des poids associés. Sa capacité à fonctionner entièrement en mémoire permet une intégration aisée sans nécessiter de serveur de recherche externe. Nous avons choisi de l'utiliser car il a fourni de meilleurs résultats que BM25 de PyTerrier, ce qui a amélioré la performance de notre système de recherche.

5.4 Plaid-x

Dans ce projet, nous avons exploité les capacités avancées de PLAID-X, un système de recherche d'informations multilingue de pointe, dans le cadre de notre participation au défi ReNeuIR@SIGIR 2024. PLAID-X, qui s'inscrit comme une extension du cadre ColBERT-X[23], est spécialement conçu pour gérer la complexité des ensembles de données multilingues, permettant ainsi une récupération efficace des documents pertinents à travers différentes langues. La conception de ce système repose sur les principes robustes de la recherche dense, où PLAID-X génère des représentations hautement contextualisées pour les requêtes et les documents.
Ces représentations sont ensuite utilisées pour calculer les similarités, assurant ainsi un processus de recherche à la fois efficace et précis.

Pour implémenter ce système, le processus commence par la récupération des résultats de la plateforme TIRA depuis le dépôt spécifié. Les résultats sont stockés dans un fichier texte structuré. En chargeant ce fichier dans un DataFrame pandas pour enfin débuter la phase du re-rank. Lors de la présentation de notre pipeline initial avec Lunr à M. Maik Fröbe, l'un des organisateurs de l'atelier, il a validé notre travail et a recommandé d'intégrer Plaid-X en tant que première couche. Cette suggestion a permis d'optimiser notre approche, conduisant à des résultats convaincants. Suite à cette validation, j'ai sollicité les organisateurs pour la possibilité de soumettre deux travaux, ce qui a été approuvé.

5.5 Re-rank : All-mini-lm-6-v2

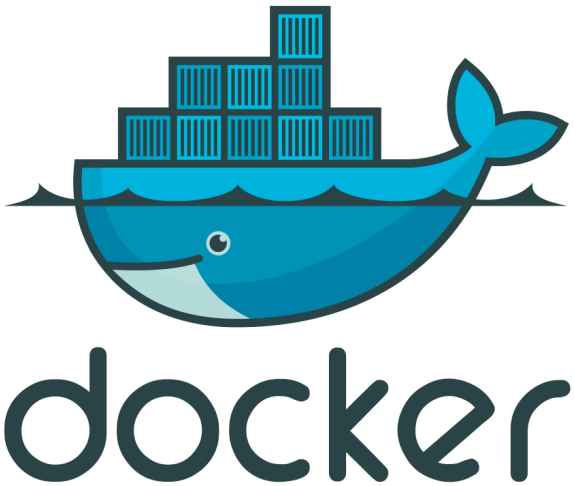
Le modèle **all-MiniLM-L6-v2** [20] représente une avancée significative dans le domaine du traitement du langage naturel, en exploitant l'architecture MiniLM pour fournir des embeddings de phrases à la fois efficaces et performants. Conçu pour équilibrer performance et efficacité computationnelle, ce modèle est une variante de MiniLM basé sur les transformateurs, offrant une alternative plus compacte et rapide aux modèles plus grands comme BERT. Le "L6" dans son nom indique qu'il fonctionne avec 6 couches de transformateurs, permettant une représentation à la fois compacte et robuste des informations textuelles. Avec une dimension d'embedding de 384, ce modéle traduit les phrases en vecteurs de haute dimension qui capturent leur signification sémantique, le rendant particulièrement efficace pour des tâches telles que la similarité textuelle sémantique, la recherche sémantique et la détection de paraphrases. Son efficacité de traitement lui permet de performer de manière compétitive dans des scénarios où la rapidité et les contraintes de ressources sont critiques, tout en maintenant une bonne compréhension des nuances contextuelles. Le modèle est entraîné sur des ensembles de données diversifiés pour garantir que ses embeddings sont polyvalents et applicables à divers domaines. Cela en fait un excellent choix pour des applications pratiques en analyse de texte, y compris le clustering de documents, la récupération d'informations et la classification de texte, où une compréhension sémantique rapide et précise est essentielle.

Dans le cadre de notre atelier, qui met l'accent sur l'efficacité et l'efficience dans les tâches de NLP, le modèle all-MiniLM-L6-v2 s'est révélé être un choix exceptionnel. Notre processus d'évaluation a impliqué une comparaison approfondie de diverses alternatives, dont BERT, RoBERTa et DistilBERT, chacune testée pour ses performances dans la génération d'embeddings de phrases et son impact sur le temps de calcul. Malgré les points forts de ces modèles, ce dernier les a systématiquement surpassés, notamment en termes de nDCG@10, une métrique cruciale pour évaluer la qualité des systèmes de classement. L'architecture du modèle, avec ses 6 couches de transformateurs et ses embeddings de 384 dimensions, offre une combinaison optimale de rapidité et d'efficacité, le rendant particulièrement adapté aux tâches de re-ranking. Sa capacité à fournir des embeddings de haute qualité de manière efficace s'aligne parfaitement avec les objectifs de notre atelier, visant à atteindre une performance supérieure et une efficacité opérationnelle dans les applications NLP. La facilité d'intégration dans les applications via la bibliothèque Sentence-Transformers renforce encore son utilité, permettant aux développeurs d'incorporer de manière fluide des capacités sophistiquées de traitement du texte dans leurs projets.

6 Dépot

La date limite de soumission pour le ReNeurIR 2024 était fixée au 26 juin. Cependant, en raison des examens que nous avions à cette période, nous n'avons pas eu le temps de finaliser notre pipeline avant cette date. Nous avons donc dû contacter M. Mike Frobe, un des organisateurs de l'atelier, pour lui expliquer notre situation. M. Frobe a accepté de faire une exception et a prolongé la date limite jusqu'au 20 août. Cette extension nous a permis de disposer de suffisamment de temps pour terminer notre pipeline et entamer le processus de soumission sur le site web de TIRA. Grâce à cette extension, nous avons pu finaliser notre travail avec le niveau de qualité que nous souhaitions et soumettre notre projet dans les meilleures conditions.

6.1 Docker



Docker[21] est une plateforme qui permet de créer, déployer et exécuter des applications dans des conteneurs. Les conteneurs Docker encapsulent une application et toutes ses dépendances (bibliothèques, configurations, etc.) dans une unité isolée, ce qui assure que l'application fonctionne de manière cohérente et fiable, peu importe où elle est déployée. En utilisant Docker, les développeurs peuvent éviter les problèmes liés aux différences d'environnement entre le développement, le test et la production, et ainsi simplifier le déploiement d'applications.

Dans notre cas, nous avons utilisé Docker pour créer une image contenant notre pipeline. L'image Docker est définie par un fichier Dockerfile, qui spécifie les étapes nécessaires pour construire l'image. Voici une explication des instructions contenues dans notre Dockerfile :

- FROM pytorch/pytorch:2.3.0-cuda12.1-cudnn8-runtime : Cette ligne indique que l'image Docker est basée sur une image de PyTorch préconfigurée avec CUDA 12.1 et cuDNN 8, fournissant un environnement optimisé pour le calcul sur GPU.
- RUN pip3 install : Cette commande installe plusieurs packages Python nécessaires au fonctionnement du pipeline, notamment transformers, lightning, pandas, cherche[gpu], et tira. De plus, elle installe ir-datasets sans ses dépendances (grâce à –no-deps), puis nettoie le cache de pip pour réduire la taille de l'image. La ligne huggingface-cli download sentence-transformers/all-MiniLM-L6-v2 télécharge un modèle pré-entraîné de Hugging Face.

En fournissant notre soumission sous forme d'image Docker, nous avons facilité la reproductibilité et la portabilité de notre pipeline. Les évaluateurs peuvent ainsi exécuter notre pipeline dans un environnement contrôlé et cohérent, sans se soucier des différences d'infrastructure ou de configuration.

Grâce à cette extension de délai jusqu'au 20 août, nous avons pu finaliser notre travail avec le niveau de qualité souhaité et soumettre notre projet sous forme d'image Docker sur le site web de TIRA. Cette approche a non seulement garanti la cohérence du déploiement mais aussi simplifié le processus d'évaluation.

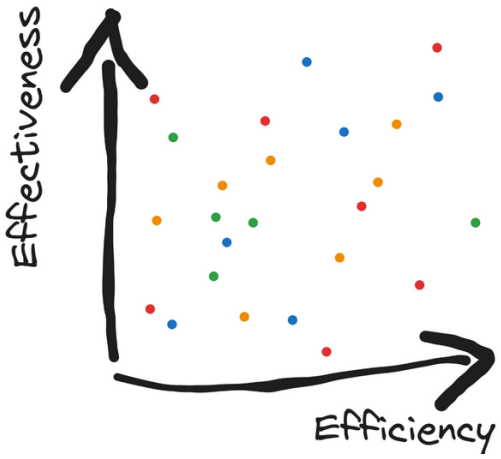
6.2 Tira

TIRA (Testbed for Information Retrieval and Analytics) et TIREx (Testbed for Information Retrieval Experiments) sont des plateformes sophistiquées conçues pour faciliter l'évaluation et la comparaison des systèmes de récupération

d'information (RI). TIRA fournit un environnement standardisé permettant aux chercheurs de tester et de comparer leurs systèmes en utilisant des ensembles de données prédéfinis et des métriques d'évaluation, tout en offrant des outils pour gérer et analyser les expériences de récupération. De son côté, TIREx se concentre spécifiquement sur la recherche expérimentale, en fournissant un cadre pour réaliser des expériences contrôlées et analyser les performances des systèmes dans diverses conditions. Ces deux plateformes sont essentielles pour faire progresser la recherche en RI en garantissant des évaluations cohérentes et des résultats comparables. Dans notre cas, nous avons utilisé ces plateformes pour soumettre notre travail sous forme d'image Docker. Docker assure que notre pipeline est déployé dans un environnement cohérent et reproductible, en éliminant les différences potentielles d'infrastructure et de configuration. Cette approche garantit non seulement l'intégrité et la portabilité de notre soumission, mais s'aligne également avec les normes rigoureuses d'évaluation définies par TIRA et TIREx.

6.3 Métriques

Une métrique est une mesure quantitative utilisée pour évaluer, comparer ou suivre divers aspects d'un système ou d'un processus. Elle fournit des données objectives pour analyser la performance, la qualité ou le progrès. Pour ce workshop, nous utiliserons les métriques RR, R@10, P@10 et NDCG@10, cette dernière étant la principale pour évaluer l'efficacité des systèmes en tenant compte de leur performance globale, y compris le temps de compilation.



6.3.1 NDCG@10

NDCG@10, ou Normalized Discounted Cumulative Gain à 10, est une métrique essentielle utilisée pour évaluer la qualité des systèmes de récupération d'information, notamment dans les moteurs de recherche et les systèmes de recommandation. Cette mesure évalue la pertinence des documents retournés en prenant en compte non seulement leur pertinence, mais aussi leur position dans la liste des résultats. Le Cumulative Gain (CG) calcule la somme des scores de pertinence des documents jusqu'à une certaine position dans la liste, mais il ne tient pas compte de la diminution de l'importance des positions plus basses. Le Discounted Cumulative Gain (DCG) ajuste ce calcul en appliquant un facteur de réduction logarithmique basé sur la position des documents, reconnaissant que les documents situés plus bas dans la liste ont généralement moins de valeur pour les utilisateurs. Le Normalized Discounted Cumulative Gain (NDCG) normalise ensuite le DCG en le divisant par le DCG idéal (IDCG), représentant le DCG d'une liste parfaitement ordonnée selon la pertinence. Ainsi, NDCG@10 mesure l'efficacité d'un système en évaluant les 10 premiers résultats, offrant une vue précise de la qualité des résultats les plus visibles pour les utilisateurs. Un score élevé de NDCG@10 indique que les documents les plus pertinents sont bien positionnés en haut de la liste, améliorant l'expérience utilisateur en fournissant des informations pertinentes et accessibles dès le début. De plus, NDCG@10 est la métrique principale utilisée pour mesurer l'efficacité des systèmes soumis dans le cadre du workshop, garantissant ainsi une évaluation rigoureuse et comparative des performances des différents systèmes de récupération.

6.3.2 RR

RR, ou Reciprocal Rank, est une métrique fondamentale pour évaluer les systèmes de récupération d'information, mesurant la position du premier document pertinent dans la liste des résultats. Calculé comme l'inverse de la position de ce premier document pertinent, RR fournit une mesure simple mais efficace de la capacité d'un système à identifier rapidement des informations pertinentes. Par exemple, si le premier document pertinent apparaît en 3ème position, RR est 130.33310.33, tandis qu'un document pertinent en 1ère position donne un RR de 1.00, indiquant une performance optimale. RR est particulièrement utile pour évaluer la réactivité d'un système à fournir des résultats pertinents dès les premières positions, en complément de métriques plus détaillées comme NDCG, qui prennent en compte la pertinence des documents à différentes positions et leur normalisation. Dans le cadre du workshop, RR est utilisé aux côtés de NDCG@10 pour offrir une évaluation complète de l'efficacité des systèmes soumis, en mettant l'accent sur la capacité à délivrer des résultats pertinents de manière rapide et efficace.

6.3.3 R@10

R@10, ou Recall at 10, est une métrique essentielle pour évaluer la performance des systèmes de récupération d'information en mesurant la proportion de documents pertinents parmi les 10 premiers résultats retournés par le système. Il est calculé en déterminant le nombre de documents pertinents présents dans ces 10 premiers résultats et en le divisant par le nombre total de documents pertinents disponibles pour la requête. Un R@10 élevé indique que le système est efficace pour placer des documents pertinents parmi les premiers résultats que les utilisateurs sont susceptibles de consulter. Contrairement à NDCG, qui prend en compte la pertinence et la position des documents, ou à RR, qui mesure la position du premier document pertinent, R@10 offre une évaluation directe de la capacité du système à récupérer des documents pertinents dans une fenêtre de résultats limitée, fournissant ainsi une vue complémentaire de l'efficacité du système dans un contexte pratique de recherche.

6.3.4 P@10

P@10, ou Precision at 10, est une métrique clé pour évaluer la qualité des systèmes de récupération d'information, en mesurant la proportion de résultats pertinents parmi les 10 premiers documents retournés par le système. Calculée en comptant le nombre de documents pertinents parmi ces 10 premiers résultats et en le divisant par 10, P@10 fournit une mesure directe de la précision dans cette fenêtre de résultats. Un score élevé de P@10 indique que le système est efficace pour fournir des résultats pertinents dès le début, ce qui est crucial car les utilisateurs se concentrent souvent sur les premiers résultats affichés. Contrairement à NDCG, qui prend en compte la pertinence et la position des documents tout en normalisant les scores, et à R@10, qui mesure la proportion de documents pertinents parmi les 10 premiers par rapport au nombre total de documents pertinents, P@10 se focalise uniquement sur la qualité des premiers résultats, offrant ainsi une évaluation claire et immédiate de la capacité du système à répondre aux besoins des utilisateurs dès les premières positions.

7 Résultats

comme déja mentionné , pour ce TER , on est passé par deux phases , une phases de recherches et de tests et une autre de création finale. voici alors quelques résultats obtenus pour les deux phases :

Les lignes en gras montrent les meilleurs résultats en termes de ndcg@10 et temps de compilation. j'ai fait des tests sur les deux jeux de données dans la phase de recherches, vu que 'DL-10-top-docs' est plus leger pour anticiper les résultats potentiels et eviter les erreurs en fin de compilation.

Méthodes sur Spacy + index inverse	**DL10**	**DL1000**
Bm25	0.4569	0.4061
TF-IDF	0.4537	0.4098
Bm25 + bert-base-uncased	0.3734	0.1079
Bm25 + tiny-bert	0.1697	run-out
Bm25 + monobert	0.4425	run-out
Bm25 + monobert + duobert	0.2814	run-out
Bm25 + T5	0.3211	0.3199
Bm25 + distil-bert	0.3372	0.1926
Bm25 + T5-mono	**0.4129**	**0.3611**

Table 1: nDCG@10 de la phase de recherches sur les dataset DL10 et DL1000

Méthodes utilisées avec CHERCHE	**DL1000**
Lunr	0.4851
TF-IDF	0.4423
Lunr + paraphrase-albert-small-v2	0.5280
Lunr + All-mini-lm-6-v2	**0.6564**
Lunr + LaBSE	0.6564
Lunr + all-mpnet-base-v2	0.6767
Lunr + bert	0.5312
Plaid-X + All-mini-lm-6-v2	**0.6824**

Table 2: nDCG@10 de plusieurs Méthodes de CHERCHE

ndcg@10	RR	R@10	P@10
0.6564	**0.9295**	**0.1990**	**0.7392**
0.6824	**0.9429**	**0.2087**	**0.7753**

Table 3: Metrics des deux meilleurs résultats

8 Conclusion

8.1 Bilan des Résultats

Au début de notre TER, après avoir participé au workshop, ma binôme Alexandra et moi étions confiantes et avons entamé nos recherches et tests en utilisant PyTerrier ainsi que les baselines fournies par la conférence. Cependant, les résultats n'étaient pas satisfaisants. Nous avons alors décidé de stopper ces tests pour commencer à construire notre pipeline final, moment où nous avons intégré l'outil **Cherche**. En testant ses retrievers, nous avons constaté que **Lunr** offrait les meilleurs résultats initiaux, ce qui nous a conduits à l'utiliser comme première couche de notre

pipeline.

Après avoir finalisé cette étape, nous avons évalué plusieurs PLMs pour sélectionner les meilleurs candidats. Parmi eux, **all-MiniLM-L6-v2,** qui est un bi-encoder, s'est distingué en nous offrant les résultats souhaités avec un temps de compilation raisonnable. Bien que nous ayons testé plusieurs autres modèles de Hugging Face, y compris les plus connus, les temps de compilation étaient souvent trop élevés, ce qui nous a confortées dans notre choix.

Ensuite, j'ai contacté l'organisateur de la conférence, Maik Fröbe, pour confirmer si notre pipeline répondait aux exigences de l'événement. Il s'est montré intéressé par notre travail et a suggéré d'intégrer **PLAID-X** comme première couche. Cette recommandation a conduit à la création de notre second pipeline.

Pendant ce temps, Alexandra s'est chargée de rédiger un rapport détaillant le travail effectué, rapport qui a été soumis sur EasyChair, en décrivant minutieusement toutes les étapes de notre processus.

Finalement, je suis très satisfait du travail accompli et des résultats obtenus, qui nous ont permis de surpasser d'autres équipes[22].

8.2 Bilan personnel

Cette expérience a été particulièrement formatrice, tant sur le plan technique que personnel. Travailler sur ce projet m'a permis de renforcer mes compétences en recherche et en développement, notamment dans la création de pipelines complexes et l'évaluation de modèles de traitement du langage naturel. Les défis imprévus, comme les résultats initiaux décevants, m'ont appris à faire preuve de flexibilité et à m'adapter rapidement.

Le stress et la pression liés aux délais et aux attentes élevées ont également été des éléments marquants de cette expérience. J'ai appris à gérer ces situations en restant concentré et méthodique, ce qui m'a permis de surmonter les difficultés et de faire avancer le projet malgré les obstacles.

L'interaction avec des experts, comme Maik Fröbe et mon tuteur Jose Moreno, a été un moment clé pour valider et affiner notre approche, ce qui a renforcé la qualité du travail accompli. Au final, je suis très satisfait des résultats obtenus, et cette réussite a consolidé ma confiance en mes compétences ainsi que ma capacité à mener à bien des projets ambitieux dans le domaine de l'IA.

8.3 Perspectives

À la suite de ce projet et de mes expériences, je m'oriente vers un rôle d'ingénieur spécialisé en intelligence artificielle. Je suis particulièrement

attiré par l'aspect pratique du développement et de la mise en œuvre de solutions basées sur l'IA. Travailler dans un environnement où je peux appliquer mes compétences techniques pour résoudre des problèmes concrets est une perspective qui m'enthousiasme. Un poste d'ingénieur en entreprise me permettrait de rester connecté aux avancées technologiques tout en contribuant de manière directe à des projets innovants et impactants. De plus, je souhaite participer aux prochaines conférences pour enrichir mes connaissances et rester à la pointe des évolutions dans le domaine de l'intelligence artificielle.

9 Remerciments

Avant de conclure ce rapport, je souhaite exprimer ma profonde gratitude à Monsieur Moreno pour son précieux tutorat. Bien que le début de ce travail de recherche n'ait pas été optimal, Monsieur Moreno a constamment su nous encourager et nous soutenir tout au long de notre projet.

Je tiens également à remercier Madame Lynda Lechani-Tamine pour ses cours sur les fondements de la recherche d'information. Grâce à son enseignement, nous avons acquis une base solide qui a grandement contribué à la réussite de ce travail de recherche.

Enfin, je souhaite adresser mes sincères remerciements à Monsieur Mike Frobe, l'organisateur de cette conférence. Monsieur Frobe a fait preuve d'une grande gentillesse et a toujours répondu à nos courriels dans les plus brefs délais. Il a également accepté de prolonger le délai pour le dépôt de notre pipeline, ce qui a été d'une aide précieuse pour nous.

10 Réferences

[1]https://sigir-2024.github.io/

[2]https://www.irit.fr/

[3]https://www.tira.io/

[4]https://www.irit.fr/departement/gestion-de-donnees/iris/

[5]https://spacy.io/

[6]Andrzej Białecki, Robert Muir, Grant Ingersoll Lucid Imagination in apache lucene 4

[7]https://github.com/castorini/anserini

[8]https://pyterrier.readthedocs.io/en/latest/

[9]https://induraj2020.medium.com/what-are-sparse-features-and-dense-features-8d1746a77035

[10]Yuncheng Li, Yale Song, Jiebo Luo; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3617-3625 Improving Pairwise Ranking for Multi-Label Image Classification

[11]Supplemental movie, appendix, image and software files for, Ranking-Oriented Collaborative Filtering: A Listwise Approach

[12]Pairwise versus Pointwise Ranking: A Case Study, Vitalik Melnikov1, Pritha Gupta1, Bernd Frick2,DanielKaimann2, Eyke H ullermeier1

[13]Tie-Yan Liu (2009), "Learning to Rank for Information Retrieval", Foundations and Trends® in Information Retrieval: Vol. 3: No. 3, pp 225-331. http://dx.doi.org/10.1561/1500000016

[14]OpenMatch-v2: An All-in-one Multi-Modality PLM-based Information Retrieval Toolkit

[15]https://huggingface.co/

[16]https://github.com/raphaelsty/cherche/tree/main

[17]Sourty, R., Moreno, J. G., Tamine, L., Servant, F.-P. (2022). CHERCHE: A new tool to rapidly implement pipelines in information retrieval. In *Proceedings of SIGIR 2022*.

[18]Another Look at DPR: Reproduction of Training and Replication of Retrieval

[19]An Introduction to Neural Information Retrieval

[20]Readability Classification with Wikipedia Data and All-MiniLM Embeddings

[21]https://www.docker.com/

[22]https://www.tira.io/task-overview/reneuir-2024/dl-top-1000-docs-20240701-training

[23]https://github.com/hltcoe/ColBERT-X