

Rapport Project n°1 : NLP

I. Introduction.....	2
II. Objectif.....	2
III. Approches utilisées.....	2
III.1.Explication du code fourni.....	2
III.1.1.Fonctions Principales.....	2
III.1.2. Détails de run_bm25_only.....	3
III.2. TF-IDF	3
III.3. Word2Vec.....	3
III.3. Bert-base.....	3
III.4. Bio-Bert -Sentences similarities.....	4
III.5. Biomed NLP-BiomedBert.....	4
IV. Application Streamlit.....	6
V. Conclusion.....	7

I. Introduction

Dans le domaine du Traitement Automatique du Langage Naturel (TALN), le développement de systèmes de recherche d'information robustes et efficaces représente un défi significatif et un domaine de recherche en constante évolution. Ce projet s'inscrit dans l'exploration et l'avancement des techniques de recherche d'information, avec un accent particulier sur le dépassement de l'efficacité de BM25, une référence bien établie dans le domaine.

Le projet souligne que **BM25**, une amélioration de l'approche traditionnelle TF-IDF, reste l'une des meilleures méthodes pour divers ensembles de données. Cependant, ce projet vise à défier ce statu quo en développant un système capable de surpasser BM25 dans un contexte spécifique.

II. Objectif

L'objectif principal est de créer un système de récupération de l'information (*information retrieval system*) innovant adapté au NFCorpus, un corpus médical spécialisé composé de résumés de papier de recherche scientifiques provenant de PubMed et de vulgarisations associées.

BM25 sert de référence pour ce projet. Le défi est de développer un système qui non seulement concurrence, mais améliore également le modèle BM25. L'efficacité du système proposé doit être mesurée à l'aide de la métrique NDCG score, en se concentrant sur le top 5, correspondant à la capacité du système à retourner les 5 documents les plus pertinents répondant à une requête donnée.

Le code fourni avec le projet obtient un résultat de 0.81 pour le top 5 parmi 150 documents:

```
ndcg bm25= 0.8135524489389909  
0.8135524489389909
```

III. Approches utilisées

III.1. Explication du code fourni

Notre démarche a commencé par une analyse approfondie du code source fourni, qui constitue la base de notre système de recherche d'information.

III.1.1.Fonctions Principales

1. **loadNFCorpus**: Charge les documents provenant du corpus fourni: les requêtes et les relations entre requêtes et documents (évaluations) à partir des fichiers téléchargés.
2. **text2TokenList**: (preprocessing) Traite le texte en supprimant les mots vides (stop words) et en le tokenisant. Cette fonction est cruciale pour préparer les données pour BM25.
3. **run_bm25_only**: Cœur du script. Cette fonction charge le corpus, prépare les données, et utilise l'algorithme BM25 pour évaluer la pertinence des documents par rapport aux requêtes.

III.1.2. Détails de run_bm25_only

- Sélectionne un sous-ensemble de documents et de requêtes (dans le code: les 150 premiers documents).
- Créer un vocabulaire et un corpus pour les documents et les requêtes.
- Utilisation du BM25 pour calculer les scores $\log(\text{idf})$ de chaque document en fonction de chaque requête.
- Évalue la performance en utilisant NDCG, une métrique qui prend en compte la pertinence de l'ordre des documents classés retournés.

III.2. TF-IDF

Nous avons exploré l'approche TF-IDF pour mieux comprendre le fonctionnement de BM25, qui est une version améliorée de TF-IDF. Le TF-IDF (pour « Term Frequency-Inverse Document Frequency ») est une méthode de pondération des termes utilisée dans la recherche d'informations. Elle permet de déterminer l'importance d'un terme dans un document donné, en fonction de sa fréquence d'apparition dans le document et de sa rareté dans l'ensemble des documents de la collection.

Cette exploration a donné un résultat de 35% au score TF-IDF.

```
23 average_ndcg = sum(ndcg_scores) / len(ndcg_scores)
24
25 # Display the average NDCG score
26 print(f"Average NDCG Score for all queries: {average_ndcg:.4f}")
27
✓ 43.8s
Average NDCG Score for all queries: 0.3551
```

III.3. Word2Vec

Comme proposé dans dans l'énoncé du projet, nous avons testé une approche Word2Vec. Word2vec est un modèle d'apprentissage automatique qui permet de représenter des mots sous forme de vecteurs numériques, avec la particularité de prendre en compte le contexte. Il repose sur l'idée que les mots ayant des contextes similaires ont tendance à avoir des significations similaires.

L'utilisation d'un modèle Word2Vec entraîné sur nos documents n'a pas permis d'avoir un résultat satisfaisant. Nous obtenons un résultat de 30%, moins bien que le TD-IDF.

```
Requirement already satisfied: rank_bm25 in c:\python311\lib\site-packages (0.2.2)
Requirement already satisfied: numpy in c:\python311\lib\site-packages (from rank_bm25) (1.26.0)
fatal: destination path 'project1-2023' already exists and is not an empty directory.
[nltk_data] Downloading package punkt to
[nltk_data]   C:\Users\rosel\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]   C:\Users\rosel\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Word2Vec Average NDCG: 0.30354577459717424
```

Ainsi, nous avons tenté d'approfondir nos recherches sur des approches qui prennent en compte le contexte de manière plus performante, notamment BERT qui est un modèle qui comprend le contexte d'où notre choix de poursuivre sur une approche reposant sur la famille BERT.

III.3. BERT-base

Similarité:

Pour cette partie, nous avons utilisé le produit scalaire (dot product) en tant que mesure de similarité.

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos \alpha$$

Le produit scalaire mesure la similarité entre deux vecteurs en calculant la somme des produits élément par élément des deux vecteurs. Plus le produit scalaire est élevé, plus les vecteurs sont similaires. Il permet aussi de prendre en compte à la fois la direction et la magnitude des vecteurs. Dans le cas de vecteurs de documents ou de requêtes, la magnitude peut être interprétée comme l'importance ou la fréquence des termes présents dans le vecteur. Une magnitude plus élevée indique que le vecteur a une plus grande importance ou contient des termes plus fréquents. Dans ce contexte, la direction fait référence à l'orientation ou à la tendance des vecteurs de documents ou de requêtes. La direction d'un vecteur est déterminée par les valeurs des composantes du vecteur et représente la relation entre les différents termes ou caractéristiques représentés par le vecteur.

<https://huggingface.co/bert-base-uncased>

BERT (Bidirectional Encoder Representations from Transformers). BERT est un modèle de traitement du langage naturel (NLP) basé sur les transformers. Il a été développé par Google en 2018. L'une de ses caractéristiques principales est sa capacité à comprendre le contexte des mots dans une phrase en examinant les mots qui les entourent, à la fois en amont et en aval. Cela lui permet de capturer des informations plus riches.

En consultant HuggingFace, nous avons opté pour Bert base afin de vectoriser les mots, puis de calculer les similarités cosinus suivies du calcul du score NDCG. Nous avons adapté la base du code fourni en intégrant une fonction d'embedding pour le traitement direct des documents, obtenant ainsi un résultat de 78%.

1) BERT base

```
[25] 1 run_bert_for_document_ranking(0, nb_docs,model_base,tokenizer_base)
✓ 34.0s
... NDCG for document ranking with BERT = 0.7852518407813259
... 0.7852518407813259
```

III.4. Bio-Bert -Sentences similarities

<https://huggingface.co/pritamdeka/BioBERT-mnli-snli-scinli-scitail-mednli-stsb>

Avec des documents médicaux, Biobert a semblé être une approche pertinente. L'utilisation de cette base a mené à un résultat de 79%.

2) BioBERT for sentences similarities

```
[26] 1 run_bert_for_document_ranking(0, nb_docs,model_bio,tokenizer_bio)
✓ 39.6s
... NDCG for document ranking with BERT = 0.7932967492743007
... 0.7932967492743007
```

En explorant davantage Hugging Face, nous avons découvert d'autres méthodes basées sur Bert mais plus spécifiques à la médecine.

III.5. Biomed NLP-BiomedBert

<https://huggingface.co/microsoft/BiomedNLP-BiomedBERT-base-uncased-abstract-fulltext>

BiomedBERT, entraîné à partir de résumés de PubMed et d'articles complets de PubMedCentral, a démontré d'excellentes performances dans plusieurs tâches de NLP biomédical. Il est actuellement le meilleur dans le Biomedical Language Understanding and Reasoning Benchmark.

Vu que le NFCorpus est composé de résumés de PubMed, l'utilisation de BiomedBERT pour la vectorisation des mots semblait judicieuse. Cette approche a abouti à un résultat impressionnant de 95%.

3) BioMedBERT

```
1 run_bert_for_document_ranking(0, nb_docs,model_med,tokenizer_med)
[27] ✓ 30.6s
... NDCG for document ranking with BERT = 0.9533814120500341
... 0.9533814120500341
```

IV. Application Streamlit

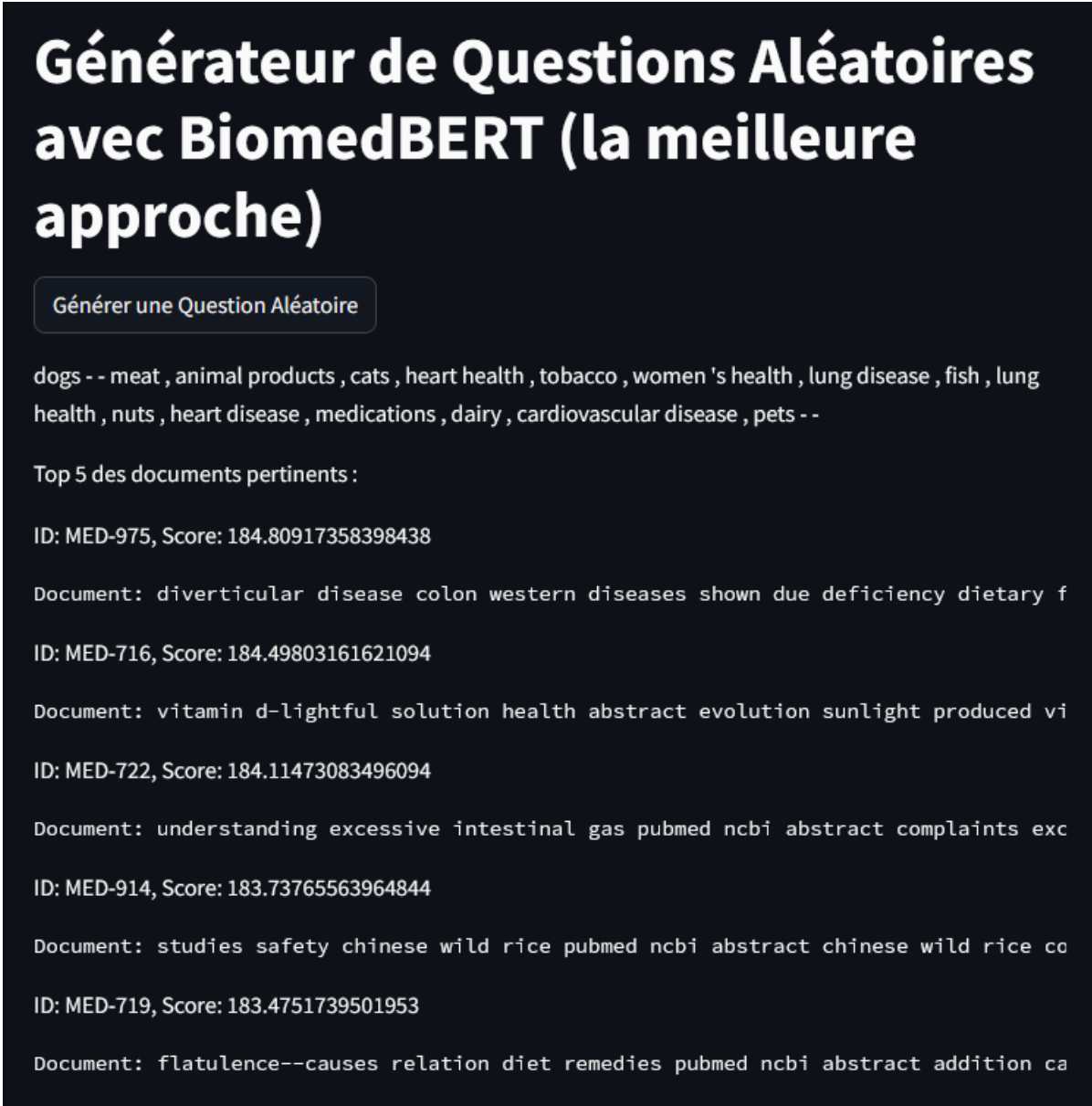
L'application Streamlit a été conçue pour sélectionner aléatoirement une question à partir de l'ensemble de données: `dev.all.queries` et pour trouver les 5 documents les plus pertinents en réponse à cette question, en utilisant le modèle BiomedBERT.

Nous allons décrire les étapes qui nous ont permis de construire l'application streamlit:

1. Chargement des Données :
 - Les données de questions (`dev.all.queries`) et de documents (`dev.docs`) ainsi que les relations entre questions et documents (`dev.2-1-0.qrel`) sont chargées à partir de fichiers.
2. Initialisation de BiomedBERT :
 - Le modèle BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext et son tokenizer associé sont chargés pour traiter le texte des questions et des documents.
3. Fonctionnalités Clés de l'Application :
 - Sélection Aléatoire de Questions : Une fonction sélectionne aléatoirement une question à partir de l'ensemble de données chargé.
 - Calcul des Embeddings BERT : Une fonction calcule les embeddings BERT pour les questions et les documents.
 - Trouver les Documents Pertinents : Une autre fonction calcule les scores de similarité entre la question sélectionnée et chaque document, puis trie ces documents en fonction de leur pertinence par rapport à la question.
4. Interface Streamlit :
 - Une interface utilisateur simple est créée avec Streamlit, où un bouton permet de générer une nouvelle question aléatoire et d'afficher les cinq documents les plus pertinents en réponse à cette question.
 - Les résultats, y compris la question et les informations sur les documents pertinents (comme leur ID et score de similarité), sont affichés dans l'interface Streamlit

<https://roselineren-nlp-esilv-projet1-streamlit-zkuzqz.streamlit.app/>

Voici ce qu'on obtient sur l'application streamlit:



Générateur de Questions Aléatoires avec BiomedBERT (la meilleure approche)

Générer une Question Aléatoire

dogs - - meat , animal products , cats , heart health , tobacco , women 's health , lung disease , fish , lung health , nuts , heart disease , medications , dairy , cardiovascular disease , pets - -

Top 5 des documents pertinents :

ID: MED-975, Score: 184.80917358398438
Document: diverticular disease colon western diseases shown due deficiency dietary f

ID: MED-716, Score: 184.49803161621094
Document: vitamin d-lightful solution health abstract evolution sunlight produced vi

ID: MED-722, Score: 184.11473083496094
Document: understanding excessive intestinal gas pubmed ncbi abstract complaints exc

ID: MED-914, Score: 183.73765563964844
Document: studies safety chinese wild rice pubmed ncbi abstract chinese wild rice co

ID: MED-719, Score: 183.4751739501953
Document: flatulence--causes relation diet remedies pubmed ncbi abstract addition ca

Le score est le score de similarité calculé entre une question spécifique et le document dans notre application utilisant BiomedBERT.

Un score élevé suggère que, selon le modèle BiomedBERT, le contenu du document est très pertinent ou similaire par rapport à la question posée. Cela peut indiquer que le document contient des informations qui sont susceptibles de répondre à la question ou d'être en rapport avec celle-ci.

V. Conclusion

Ces différentes approches traduisent nos efforts à explorer diverses techniques de traitement du langage naturel pour améliorer la récupération d'information pour ce challenge, tout en se concentrant spécifiquement sur des données médicales et scientifiques complexes. Notre objectif est de maximiser la précision et la pertinence des résultats de recherche, en surpassant les méthodes traditionnelles. Résultats nous obtenons de très bons résultats sur ce corpus précis avec les modèles BERT pré-entraîné sur des données médicales. L'inconvénient de ces approches réside plutôt dans le fait qu'on doit se restreindre à un secteur d'application précis et ce sont des approches qui auraient du mal à être généralisées sur des données ne traitant pas de sujets médicaux. Dans ces cas-là nous préconisons alors l'approche BERT base ou BM25.

Enfin, nous avons tenu à construire une application streamlit interactive pour pouvoir mieux explorer les performances de notre modèle. Cela nous permet de voir comment une requête aléatoire est liée aux documents de notre corpus de données, en utilisant l'analyse de similarité sur le modèle de word Embedding BiomedBERT. Tout en permettant une utilisation pratique des modèles de traitement du langage naturel et de l'apprentissage automatique dans une application web facile à utiliser.