

Fun-MOOC

23.05.2025

Dos Santos Steve et Proust Maximilien

Greta Centre Val de Loire

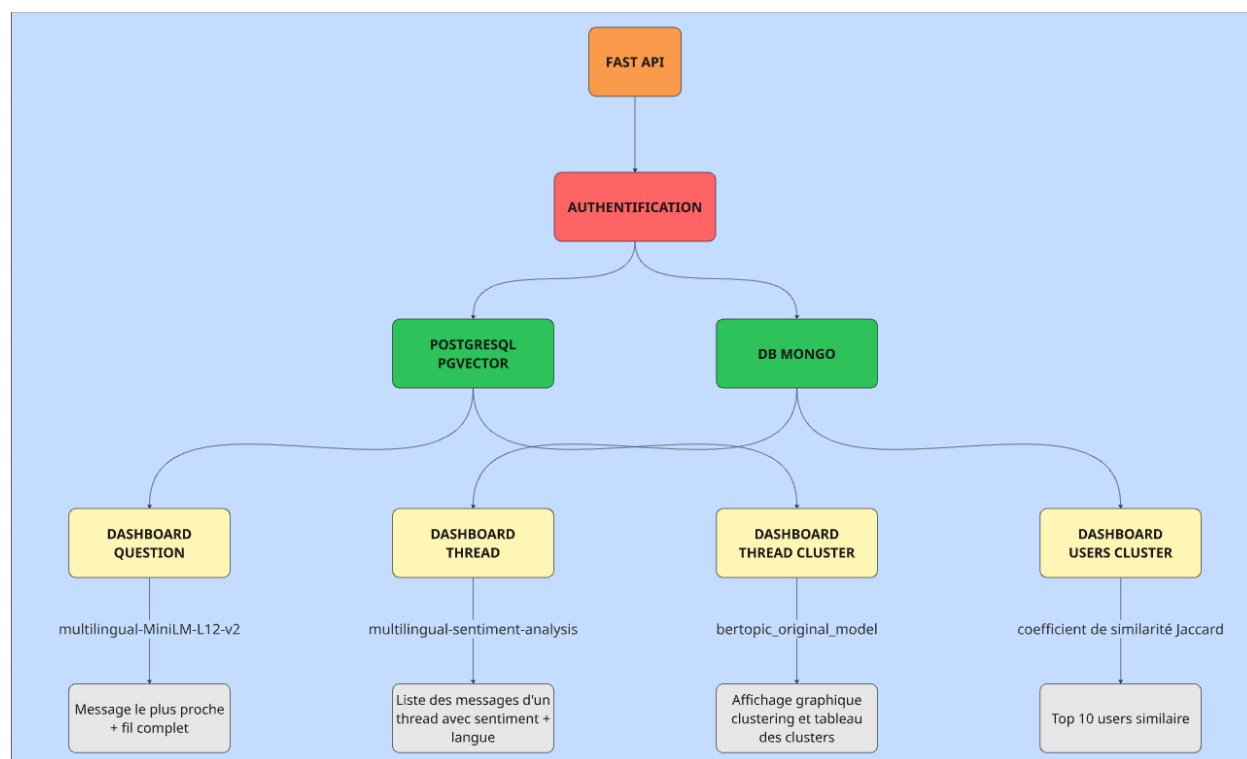
Présentation du projet	2
Les bases de données	3
Les modèles d'intelligence artificielle utilisées :	4
distilbert-based Multilingual Sentiment Classification Model	4
paraphrase-multilingual-MiniLM-L12-v2	4
Bertopic	4
Le dashboard des questions/réponses	5
Le dashboard des sentiments	5
Le dashboard des fils de discussion	5
Le dashboard des parents proches	6
Les points d'amélioration	6

Présentation du projet

L'objectif final du projet est de développer une application permettant d'observer des données récupérées d'un forum d'enseignement bien connu, fun MOOC. Pour arriver à cela, nous sommes passés par plusieurs étapes. La première d'entre elles a été de récupérer les données. Une fois cette tâche effectuée, nous avons pu analyser les données et observer ce que nous voulions faire. Nous avons décidé de répartir cela en 4 tâches distinctes qui sont listés ci-dessous :

- Nous voulions avoir la possibilité de poser une question et que l'application retourne le message le plus proche avec la possibilité de retrouver le fil de discussion complet pour avoir les tenants et les aboutissants de la discussion.
- Nous souhaitons également avoir la possibilité d'observer si les discours dans les messages sont plutôt positifs et donc recommandés le cours/le fil ou au contraire négatif et pour cela, nous avons procédé à une analyse de sentiments des messages.
- En troisième point, nous avons voulu développer une interface permettant d'observer un regroupement entre différents fils de discussion pour observer leurs similarités
- Enfin, dans le dernier point nous avons développé une interface permettant d'observer la similarité entre les différents utilisateurs.

Pour tout cela, nous avons utilisé différents outils que je vais présenter dans les parties suivantes. Mais avant tout cela, vous retrouverez ci-dessous le schéma fonctionnel de l'application.



Les bases de données

Notre première étape a été de mettre nos données dans une base de données pour pouvoir les exploiter. Pour cela, nous avons créé une base de données mongodb afin d'y importer notre fichier json. Nous avons choisi mongodb car les données sont sous format document et donc le support le plus pratique pour utiliser les données derrière était d'utiliser mongodb.

A partir de ce fichier, nous avons réalisé des manipulations d'embeddings que nous verrons plus tard. Ces embeddings ont été stockés dans une base de données postgres azur avec les composantes suivantes retenues :

Nom de la colonne	#	Type de données	Identité	Collation	Non null
 id	1	uuid			[v]
A-Z text	2	text		default	[]
A-Z title	3	varchar		default	[]
A-Z course_id	4	varchar		default	[]
A-Z username	5	varchar		default	[]
 created_at	6	timestamp			[]
 embedding	7	public.vector			[]

Les modèles d'intelligence artificielle utilisées :

Pour réaliser ce projet, nous avons dû utiliser 3 intelligences artificielles différentes.

distilbert-based Multilingual Sentiment Classification Model

Ce modèle a été récupéré via le site web d'huggingface. Ce modèle est basé sur le Distilbert base multilingual cased. C'est un modèle qui a une très bonne fiabilité puisqu'elle est estimée à 93%. Pour arriver à ce score, le modèle a été entraîné avec des données auto-générées et a été fine-tuné. Un avantage non négligeable de ce modèle et son côté multilingue. En effet, cela permet d'étudier le sentiment des messages dans plusieurs langues et non uniquement le français. Ce modèle est capable d'interpréter plus de 20 langues différentes ce qui correspond parfaitement à notre projet qui a plusieurs langues. Enfin, ce modèle va être capable de catégoriser les messages en 5 catégories allant de très négatif à très positif en passant par neutre. Cela permet de visualiser si les propos des utilisateurs sont approuvateurs ou le contraire !

paraphrase-multilingual-MiniLM-L12-v2

Ce modèle, également récupéré d'huggingface est un modèle largement répandu pour sa fiabilité, montré par la régression de Spearman. De plus, ce modèle a la capacité de détecter plus de 50 langues ce qui en fait un atout non négligeable pour notre projet. Le but de cette IA est donc de détecter les messages et d'en faire des embeddings. Un embedding, c'est une représentation vectorielle d'un message en plusieurs dimensions de façon à pouvoir analyser un grand nombre de messages entre eux. L'objectif final de ce modèle est de reconnaître la sémantique et donc le sens des phrases qu'on lui fournit.

Bertopic

Bertopic est un modèle récupéré d'un compte sur github qui possède également un site web. Cette intelligence artificielle va dans un premier temps encoder les documents. L'encodage est en réalité un embedding qu'on a réalisé à l'aide du modèle ci-dessus. Ensuite, une fois l'encodage réalisée, Bertopic va réaliser une réduction du nombre de dimensions pour permettre une meilleure analyse. Enfin, il va réaliser un clustering (le clustering permet de classer des données en plusieurs groupes) sur les données et proposer la meilleure classification possible selon le modèle. Cette représentation est faite à l'aide de mots clés.

Le dashboard des questions/réponses

Pour ce dashboard, nous avons enregistré les embeddings effectués avec le modèle paraphrase-multilingual-MiniLM-L12-v2 dans une base de données postgres. Cela nous permet donc d'y avoir accès dans l'application à tout moment. Ensuite, nous avons créé une interface web permettant à l'utilisateur d'entrer son message. Une fois son message entré, celui-ci va subir une transformation embedding et être comparé grâce à une requête SQL et fournir l'élément le plus proche. Ensuite, l'utilisateur a la possibilité de voir le fil de discussion entier pour lire la conversation entière. Ci-dessous, vous trouverez la requête SQL.

```
sql = text("""  
    SELECT id, text, title, course_id, username, created_at, embedding <=>  
    CAST(:embedding AS vector) AS distance  
  
    FROM public.embeddings_pgvector  
  
    ORDER BY distance ASC  
  
    LIMIT 1  
  
    """)
```

Le dashboard des sentiments

Pour ce dashboard, nous avons appliqué le modèle d'intelligence artificielle distilbert-based Multilingual Sentiment Classification Model. D'un point de vue utilisateur, celui-ci va choisir le cours qu'il souhaite analyser, puis va choisir le fil de discussion et verra alors les conclusions du modèle.

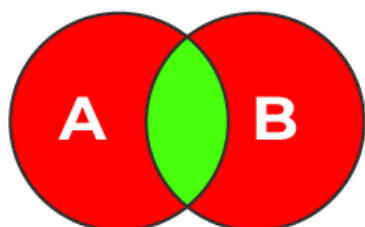
Le dashboard des fils de discussion

Pour ce dashboard, nous avons appliqué le modèle d'intelligence artificielle Bertopic. D'un point de vue utilisateur, celui-ci va pouvoir observer le résultat du clustering en ayant la possibilité d'interagir avec l'interface graphique et observer ainsi les clusters allant dans notre cas de 0 à 330 dans l'ordre décroissant. C'est à dire du clusters ayant le plus de sujet au cluster ayant le moins de sujet. Ensuite, en dessous de ce graphique, l'utilisateur va pouvoir voir les différents clusters et les mots clés associés. Ces mots clés permettront

ainsi de faire une première observation des clusters avec la possibilité d'afficher les messages présents dans ce cluster si l'utilisateur le souhaite.

Le dashboard des parents proches

Pour ce dashboard, nous avons appliqué la méthode de similarité aussi appelée Jaccard dont vous trouverez la formule ci-dessous :


$$\text{Jaccard} = \frac{\text{Intersection (A, B)}}{\text{Union (A, B)}}$$

Cette formule nous permet ainsi de simplement regarder les interactions des différents utilisateurs avec les différents fils de discussion et les différents cours. Ainsi, si l'on souhaite voir si un utilisateur A a en commun avec d'autres utilisateurs, on a qu'à rentrer l'utilisateur en question et cela nous fournit les informations sur ses 10 plus proches parents. Il faut donc connaître le pseudo de l'utilisateur pour avoir ces informations.

Les points d'amélioration

Les points d'améliorations du projet vont plutôt être tournés autour de l'interface utilisateur qui pourrait être grandement améliorée. Les autres points que nous pourrions progresser vont se situer sur le preprocessing qu'on va appliquer sur nos différents modèles.