Projet 8

Veille Technologique : le Modèle mxbai-embed-large-v1

Octobre 2024

Lokman AALIOUI

Sommaire

Introduction

Dans un monde où les données textuelles explosent, la capacité à comprendre et exploiter les textes de manière efficace est cruciale. Les modèles de **Natural Language Processing (NLP)** basés sur des embeddings textuels sont devenus des outils essentiels pour diverses industries, qu'il s'agisse d'analyser des documents, des avis clients ou de faire des recommandations dans le domaine du e-commerce. Parmi ces modèles, **mxbai-embed-large-v1**, développé par **Mixedbread AI**, représente une avancée significative, en se positionnant comme un modèle **State Of The Art** (SOTA) pour la génération d'embeddings, avec une approche innovante basée sur l'entraînement contrastif et l'utilisation de triplets pour affiner ses représentations vectorielles.

Cette veille technique a pour but de présenter ce modèle, d'explorer son architecture et ses applications, de le comparer avec des modèles plus anciens comme **BERT**, et d'analyser ses limites ainsi que ses perspectives d'amélioration.

Chapitre 1 : Modèle mxbai-embed-large-v1

1.1 Contexte de développement

Le modèle **mxbai-embed-large-v1** a été lancé par **Mixedbread AI** en mars 2024. Cette société, bien connue dans le domaine du NLP, a développé ce modèle pour répondre à des besoins croissants d'efficacité et de précision dans le domaine des embeddings textuels. L'objectif était de proposer un modèle capable de rivaliser avec les géants de l'industrie comme **BERT** tout en restant plus léger et plus rapide en termes de calcul, sans sacrifier les performances.

La technologie d’embedding est devenue un élément central dans de nombreuses applications telles que la **classification**, la **recherche d’information**, la **reconnaissance d’entités nommées**, et même la **traduction automatique**. Avec l'augmentation exponentielle des données textuelles générées chaque jour sur internet, la capacité à extraire des informations pertinentes rapidement et efficacement est devenue cruciale.

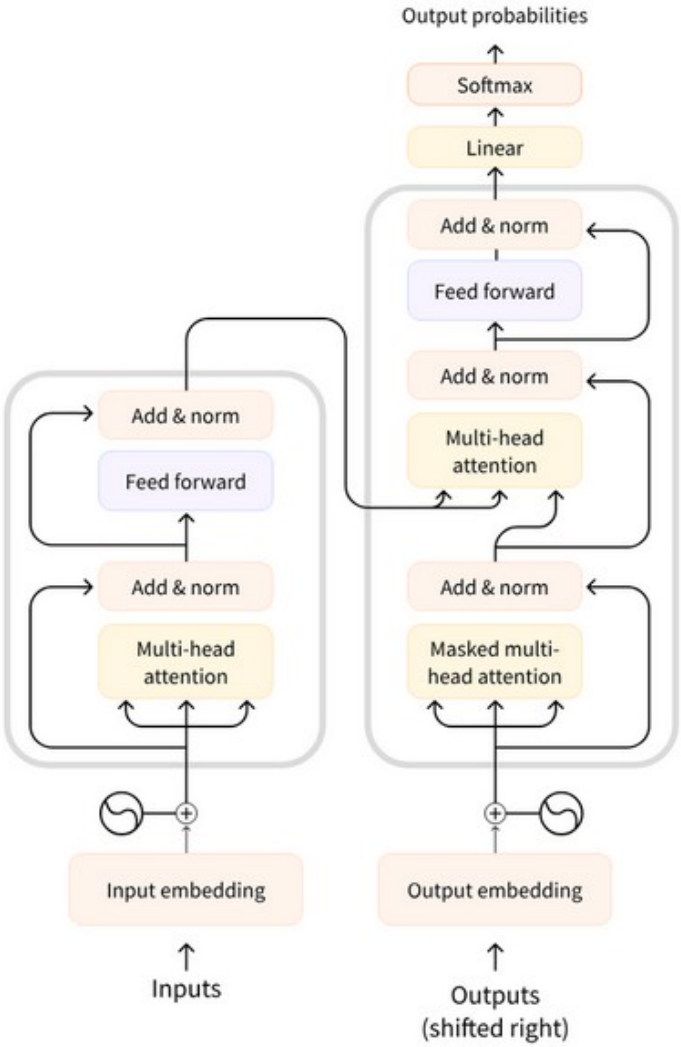
1.2 Évolution des modèles d’embeddings

Les premiers modèles d'embeddings comme **Word2Vec** et **GloVe** ont introduit la capacité de représenter les mots sous forme de vecteurs continus dans un espace vectoriel, permettant ainsi de capturer des relations sémantiques entre les mots. Cependant, ces modèles étaient statiques, ce qui signifie qu'un mot avait toujours la

même représentation, quel que soit le contexte dans lequel il apparaissait.

L'introduction de **BERT** a changé cette approche en introduisant des embeddings contextuels. Grâce à son architecture basée sur des **Transformers**, BERT peut générer des représentations différentes pour un même mot en fonction du contexte dans lequel il se trouve. Cette innovation a révolutionné le domaine du NLP, plaçant les Transformers au cœur des recherches en intelligence artificielle.

mxbai-embed-large-v1 suit cette lignée de modèles basés sur les **Transformers**, mais avec des améliorations majeures en termes d'entraînement et d'optimisation, permettant de surmonter certaines des limitations des modèles précédents, comme BERT.



1.3 Comparaison avec BERT et autres modèles

Les performances du modèle **mxbai-embed-large-v1** dépassent largement celles des modèles plus anciens comme **BERT** ou **USE** (Universal Sentence Encoder). Sa capacité à générer des embeddings plus précis réside dans son utilisation d'un

entraînement contrastif sur un énorme volume de données, et son **fine-tuning** à l'aide de **triplets**.

Contrairement à BERT, qui utilise principalement une approche d'entraînement masqué (MLM - Masked Language Model), **mxbai-embed-large-v1** apprend directement à distinguer les paires de textes similaires et non similaires, ce qui améliore considérablement sa capacité à comprendre la relation sémantique entre deux phrases ou documents.

Chapitre 2 : Architecture du modèle

2.1 Transformer Architecture

Le modèle **mxbai-embed-large-v1** est basé sur l'architecture **Transformer**, qui repose sur des mécanismes d'attention pour capturer les relations entre les éléments d'une séquence. Contrairement aux modèles séquentiels comme les **RNN** (Recurrent Neural Networks) ou **LSTM**, les **Transformers** traitent les séquences en parallèle, ce qui améliore considérablement l'efficacité des calculs et permet de gérer des textes beaucoup plus longs.

L'encodeur du modèle est chargé de transformer les séquences textuelles en vecteurs continus denses, encapsulant les informations contextuelles des phrases. Cette approche permet de gérer des tâches de **similarité sémantique**, de **classification**, et même de **résumé automatique**.

2.2 Formation contrastive et ajustement avec triplets

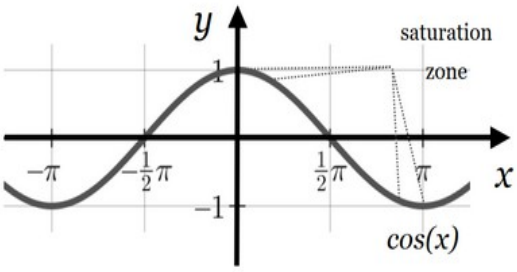
L'innovation principale de **mxbai-embed-large-v1** réside dans son **entraînement contrastif**, où il apprend à identifier des paires de textes similaires et non similaires à partir de plus de **700 millions de paires**. Cela permet au modèle de mieux capter les nuances sémantiques des textes et de distinguer des éléments très proches.

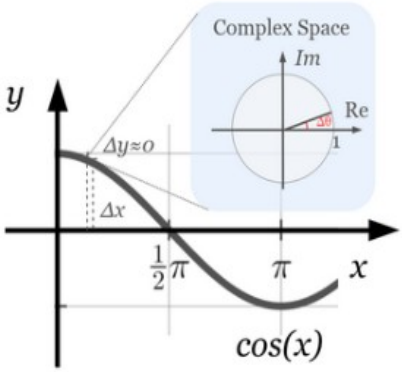
Le **fine-tuning** avec des triplets va encore plus loin. Chaque triplet comprend un texte de référence, un texte similaire, et un texte non similaire. Cette approche aide à ajuster finement les embeddings pour renforcer la cohérence des relations entre les vecteurs dans l'espace vectoriel.

2.3 Fonction de perte AngLE

La fonction de perte utilisée par **mxbai-embed-large-v1** est également un élément crucial dans ses performances. Plutôt que de se baser uniquement sur la **similarité cosinus** couramment utilisée, le modèle adopte une nouvelle fonction de perte appelée **AngLE** (Angular Loss Embedding). Cette fonction permet de capturer des

différences d'angles entre les vecteurs, ce qui permet au modèle de mieux se comporter dans les zones où les gradients cosinus sont faibles, accélérant ainsi son apprentissage.





Chapitre 3 : Applications en NLP

3.1 Cas d'utilisation en e-commerce, santé, finance

Les cas d'utilisation de **mxbai-embed-large-v1** sont variés, mais certaines applications se démarquent particulièrement :

- **E-commerce** : Le modèle peut être utilisé pour classer automatiquement des descriptions de produits, trouver des produits similaires ou encore analyser les avis clients pour en extraire des tendances et des opinions.
- **Santé** : Dans le domaine médical, l'analyse des documents textuels (rapports médicaux, publications scientifiques) est facilitée par l'utilisation de ce modèle. Il peut aider à extraire des informations pertinentes et fournir des recommandations basées sur la similarité sémantique des documents.
- **Finance** : En finance, **mxbai-embed-large-v1** peut être utilisé pour analyser des rapports financiers, identifier des tendances dans les publications économiques ou automatiser certaines tâches de veille stratégique.

3.2 Performance et comparaison avec BERT et USE

Les benchmarks réalisés montrent que **mxbai-embed-large-v1** se classe parmi les meilleurs modèles en termes de performances. Dans les classements tels que le **MTEB leaderboard**, il obtient régulièrement des scores plus élevés que **BERT** ou **Universal Sentence Encoder (USE)**. Par exemple, dans des tâches de clustering ou de classification, le modèle dépasse ses concurrents en termes de **précision** et de

rappel, tout en nécessitant moins de temps de calcul pour produire des embeddings de haute qualité.

Chapitre 4 : Outils de déploiement

4.1 Ollama vs SentenceTransformers

Le modèle **mxbai-embed-large-v1** est disponible via plusieurs plateformes de déploiement, notamment **Ollama** et **SentenceTransformers**. Chaque plateforme a ses avantages et ses inconvénients en termes de facilité d'utilisation et de personnalisation des paramètres du modèle.

- **Ollama** : Cette plateforme fournit une solution prête à l'emploi pour générer des embeddings avec **mxbai-embed-large-v1**. Elle est simple d'utilisation et permet d'obtenir rapidement des résultats sans avoir à manipuler le modèle en profondeur.
- **SentenceTransformers** : Cette bibliothèque offre plus de flexibilité pour les utilisateurs avancés, permettant de fine-tuner le modèle et de l'intégrer dans des pipelines complexes. Cependant, elle demande une meilleure maîtrise des outils NLP.

4.2 Avantages/inconvénients des plateformes

Ollama offre un environnement facile d'accès pour les utilisateurs qui recherchent une solution rapide pour générer des embeddings sans avoir à configurer eux-mêmes le modèle. Cependant, les possibilités de personnalisation sont limitées par rapport à **SentenceTransformers**, qui permet d'ajuster plus finement le modèle en fonction des besoins spécifiques de l'utilisateur.

4.3 API et compatibilité avec d'autres outils

Les APIs fournies par ces plateformes permettent d'intégrer facilement le modèle dans des systèmes plus larges, que ce soit pour des applications de **chatbots**, de **moteurs de recherche** ou de **systèmes de recommandation**. Le modèle est également compatible avec des outils de visualisation comme **SHAP** pour expliquer ses décisions et représentations.

Chapitre 5 : Résultats

5.1 Synthèse des résultats

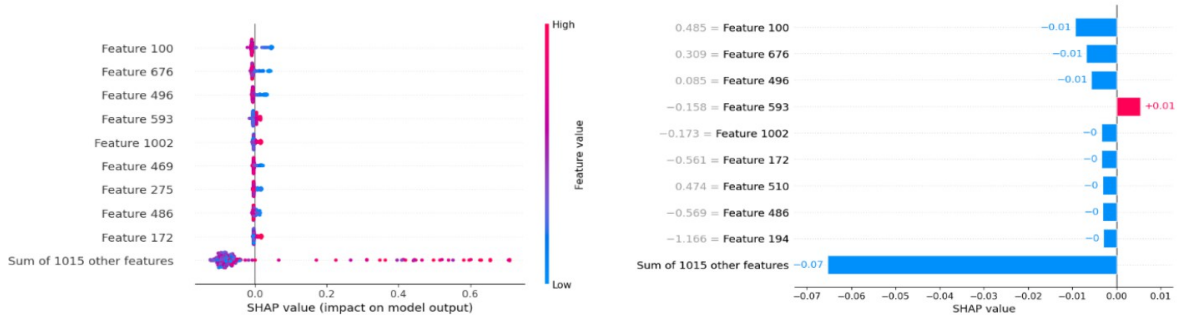
	Ollama	ST (512)	ST (1024)	BERT	USE
Score ARI	**0,709**	0,686	0,582	0,322	0,443
Accuracy	0,952	0,943	**0,962**	**X**	**X**

Au niveau des résultats obtenus, l'embedding via Ollama semble plus performant que celui avec Sentence Transformers.

Cet écart de performance est probablement dû aux (sous-)versions du modèle, qui semble avoir été mis à jour plus récemment sur Ollama. Cependant, selon des besoins précis, SentenceTransformers ou Transformers peuvent permettre une meilleure manipulation/modification du modèle, et donc être plus adaptés, malgré des performances légèrement inférieures. Dans tous les cas, ces solutions montrent des résultats significativement supérieurs aux modèles utilisés lors des tests précédents. En termes de résultats externes, le modèle se place en vingt-quatrième position dans le leaderboard MTEB, et affiche souvent des résultats supérieurs aux autres modèles testés dans les articles que j'ai pu consulter.

5.2 Analyse de la feature importance globale et locale du nouveau modèle

Par leur nature, ce type de modèles d'embedding sont dit "boites noires", c'est à dire qu'il est difficile d'interpréter le fonctionnement du modèle et de connaitre ce qui influence la génération des vecteurs. La seule possibilité d'analyse de "feature importance" serait de passer le classifieur dans SHAP par exemple, comme ci-dessous pour la classe 'Beauty and Personal Care' et le quarante-deuxième produit. 5 Mais cela n'est pas vraiment pertinent, étant donné que notre seule "feature" est notre vecteur, dont il est impossible d'identifier le sens des 1024 éléments qui le composent.



Chapitre 6 : Limites et perspectives

6.1 Limites

Bien que **mxbai-embed-large-v1** soit un modèle performant, il présente quelques limites. Tout d'abord, sa capacité à traiter des textes dans plusieurs langues est limitée aux langues pour lesquelles il a été formé, principalement l'anglais. De plus, le nombre de **tokens** qu'il peut gérer est limité, ce qui peut poser problème pour l'analyse de longs documents.

6.2 Perspectives d'améliorations

Plusieurs pistes d'amélioration sont envisageables pour ce modèle :

- **RAG (Retrieval-Augmented Generation)** : En combinant le modèle avec une approche **RAG**, il serait possible d'améliorer la qualité des réponses pour des tâches de génération de texte ou de résumés.
- **Couches supplémentaires** : L'ajout de **couches supplémentaires** au modèle pourrait permettre une meilleure capture des relations à long terme dans les textes, notamment pour des tâches plus complexes.

Conclusion

Le modèle **mxbai-embed-large-v1** est un exemple clair de l'évolution des modèles d'embeddings dans le domaine du NLP. Avec son architecture basée sur les **Transformers** et son **entraînement contrastif**, il parvient à dépasser les modèles comme **BERT** et **USE** dans plusieurs benchmarks, tout en étant plus rapide et plus flexible. Ses applications sont vastes et touchent de nombreux secteurs, de l'e-commerce à la santé en passant par la finance. Cependant, des améliorations restent possibles, notamment en termes de traitement multilingue et de gestion de longs documents.

Bibliographie

Caspari, L., Ghosh Dastidar, K., Zerhoudi, S., Mitrovic, J., & Granitzer, M. (2024). Beyond Benchmarks: Evaluating Embedding Model Similarity for. Passau, Germany: University of Passau.

Google Developers. (2024, Août 7). Classify text with BERT. Récupéré sur tensorflow.org: https://www.tensorflow.org/text/tutorials/classify_text_with_bert

Guochao, J., Zepeng, D., Yuchen, S., & Deqing, Y. (2024). P-ICL: Point In-Context Learning for Named Entity Recognition. Shanghai, China: School of Data Science, Shanghai Key Laboratory of Data Science.

Huertas-García, Á., Martí-González, C., Muñoz, J., & Ambite, E. (2024). Small Language Models and Large Language Models in Oppositional Thinking Analysis: Capabilities, Biases and Challenges. CLEF 2024 (p. 11). Grenoble, France: PAN.

Kriesch, L., & Losacker, S. (2024). A global patent dataset for the bioeconomy. Innovation Studies, 4, 8, 11.

Li, X., & Li, J. (2024). AngLE-optimised text embeddings. Hong Kong: Department of Computing, The Hong Kong Polytechnic University.

Mixedbread AI. (2024, Août 7). Embeddings - Overview - Docs. Récupéré sur mixedbread.ai: https://www.mixedbread.ai/docs/embeddings/overview

Mixedbread AI. (2024, Août 7). mixedbread-ai/mxbai-embed-large-v1. Récupéré sur huggingface.co: https://huggingface.co/mixedbread-ai/mxbai-embed-large-v1

Mixedbread AI. (2024, Août 7). mxbai-embed-large-v1 - Docs. Récupéré sur mixedbread.ai: https://www.mixedbread.ai/docs/embeddings/mxbai-embed-large-v1

MTEB. (2024, Août 9). MTEB Leaderboard - a Hugging Face Space by mteb. Récupéré sur huggingface.co: https://huggingface.co/spaces/mteb/leaderboard

Ollama. (2024, Août 7). Embedding models. Récupéré sur ollama.com: https://ollama.com/blog/embedding-models