Note méthodologique :
Preuve de concept avec DeBERTa

Maodo Fall

Contexte

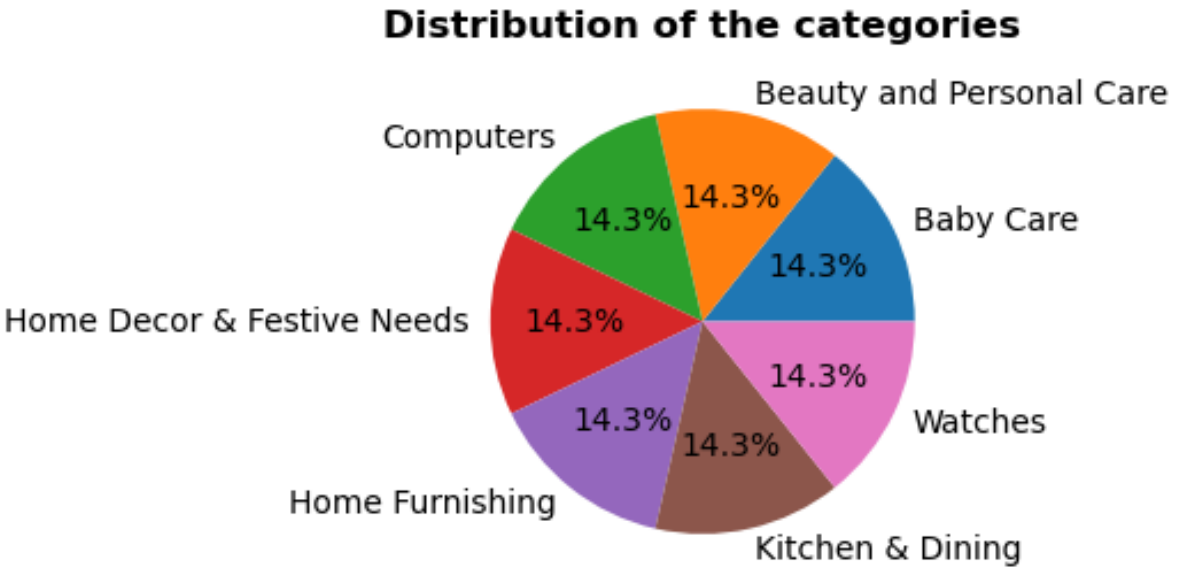
L'objectif de ce rapport est de réaliser, pour le compte de la société financière "Prêt à dépenser", un état de l'art sur une technique récente de modélisation de données, en particulier dans les domaines des données texte (NLP) et des données d'images. Cette demande s'inscrit dans un contexte où la mise en œuvre des dernières avancées en data science et machine learning constitue un axe stratégique pour l'amélioration des performances des modèles utilisés par l'équipe. Les avancées récentes dans ces domaines ont permis d'élargir les possibilités de modélisation de données complexes, notamment grâce à l'usage de modèles plus performants, comme les approches de deep learning tel que le Transformer et ses variantes.
Dans ce document, l'état de l'art se concentre sur DeBERTa (Decoding-enhanced BERT with Disentangled Attention), une technique récente de modélisation du langage naturel (NLP). Développée par Microsoft Research, DeBERTa représente une avancée significative par rapport à BERT et ses dérivés, en introduisant deux principales innovations : un mécanisme d'attention désentrelacé et un décodage amélioré. Ces améliorations permettent au modèle de mieux capter les relations entre les mots, tout en améliorant l'efficacité du traitement contextuel et l'expression des dépendances sémantiques et syntaxiques dans les textes.
L'objectif est d'analyser la technique DeBERTa, de comprendre ses spécificités et ses avantages par rapport à l'approche BERT.

Mots clés : Deep Learnig, NLP, Transformer, BERT, DeBERTa

Sommaire

1 Dataset retenu

Le dataset utilsé provient du projet 6 "Classifier automatiquement des biens de consommation" du parcours Data Science d'OpenClassrooms.
Place de Marché, une entreprise anglophone, souhaite lancer une marketplace e-commerce. Sur leur site, des vendeurs proposent des articles à des acheteurs en postant une photo et une description. Pour l'instant, la catégorisation d'un article est effectuée manuellement par les vendeurs, et est donc peu fiable.
L'entreprise a pour objectif d'automatiser cette tâche d'attribution de la catégorie.
Le jeu de données exploité est composé de 1050 articles labélisées en 07 catégories de 150 articles chacune et il inclut des informations telles que le nom du produit, sa description, et un lien vers l'image de chaque produit.



Dans ce qui suit, nous nous focalisons sur la description et la catégorie des produits pour réaliser l'état de l'art de l'approche NLP DeBERTa à présenter.

2 Les concepts de l'algorithme

Depuis l'essor des modèles Transformer en NLP, plusieurs architectures ont été développées pour améliorer la compréhension du langage naturel (NLU) et la génération de texte (NLG). Le Transformer est un type de réseau de neurone profond qui peut traiter efficacement de longs textes en utilisant le mécanisme d'attention. Une attention est un mécanisme permettant d'apprendre les relations contextuelles entre les mots d'un texte.Des modèles tel que BERT ont permis des avancées significatives dans ce cadre.
DeBERTa (Decoding-enhanced BERT with Disentangled Attention), introduit par Microsoft en 2020, apporte deux améliorations majeures par rapport à BERT.

2.1 Attention Désentrelacée (Disentangled Attention)

- Dans BERT, un mot x_i est représenté par un unique vecteur H_i obtenu par la somme de :
  - son embedding de contenu E_i
  - son embedding de position absolue P_i
- Dans DeBERTa, la représentation est dissociée en deux vecteurs indépendants :
  - Un vecteur contenu C_i pour l'information lexicale
  - Un vecteur positionnel $P_{i|j}$ encodant la position relative entre x_i et x_j

L'attention entre les mots x_i et x_j est alors calculée comme la somme de quatre termes distincts :

$$A_{i,j} = C_i C_j^\top + C_i P_{j|i}^\top + P_{i|j} C_j^\top + P_{i|j} P_{j|i}^\top$$

Chaque interaction entre tokens prend donc en compte le contenu et les positions relatives de manière explicite, au lieu de les combiner dès l'entrée du modèle comme dans BERT.

Avantages :
- Capture plus précise des relations contextuelles
- Réduction du bruit lié à l'entrelacement du contenu et de la position

2.2 Décodeur de masque amélioré (EMD)

Comme BERT, DeBERTa est pré-entraîné en Masked Language Modeling (MLM). Cependant, BERT incorpore l'information de position absolue dès l'entrée, ce qui peut nuire à l'apprentissage des dépendances relatives.
DeBERTa diffère en introduisant l'information absolue uniquement en sortie du modèle (avant le softmax final). Cela permet d'utiliser les positions relatives dans toutes les couches d'attention du modèle, puis de réintroduire l'information absolue uniquement au moment de la prédiction.

Exemple : Dans la phrase "Un nouveau magasin a ouvert à côté du nouveau centre commercial.", si "magasin" et "centre commercial" sont masqués, BERT peut confondre ces deux mots car ils sont tous deux précédés de "nouveau". DeBERTa exploite les relations relatives tout au long du traitement, et réintègre l'information absolue uniquement au moment de la prédiction.

Avantages :
- Meilleure prise en compte de la structure syntaxique
- Prédiction plus fine des tokens masqués
- Meilleure robustesse sur les tâches de complétion de texte

Ces améliorations permettent à DeBERTa d'exceller sur plusieurs benchmarks NLP tout en nécessitant moins de données d'entraînement que RoBERTa(une autre variante de BERT) par exemple.

3 La modélisation

3.1 Représentation des données textuelles

Dans cette étude comparative, les données textuelles sont d'abord pré-traitées puis vectorisées à l'aide de deux modèles de type Transformer : BERT et DeBERTa.

— Pour BERT, le modèle pré-entraîné utilisé est bert-base-uncased accompagné de son tokenizer natif. Chaque phrase est tokenisée et encodée en vecteurs de dimension 768. Afin d'obtenir une représentation unique par phrase, nous effectuons un mean pooling consistant à calculer la moyenne des embeddings des tokens non-masqués de chaque séquence.
— Pour DeBERTa, nous utilisons le modèle microsoft/deberta-large ainsi que son tokenizer dédié. La procédure de vectorisation est identique à celle de BERT : tokenisation, passage dans le modèle pour obtenir les embeddings, puis application du mean pooling pour obtenir une représentation vectorielle unique de chaque phrase.

3.2 Evaluation non supervisée et visualisation

Dans les deux cas, nous avons cherché à analyser la structure des données dans l'espace latent généré par les modèles. Après l'extraction des embeddings via BERT ou DeBERTa, un algorithme de clustering K-Means est d'abord appliqué afin de détecter les regroupements au sein des représentations vectorielles. Afin de visualiser les clusters dans un espace 2D, une réduction de dimension est ensuite réalisée à l'aide de t-SNE. Cette projection 2D permet d'interpréter visuellement la qualité de la séparation entre les groupes identifiés.
L'évaluation des clusters repose sur :

— Le Adjusted Rand Index (ARI), mesurant la cohérence entre les clusters trouvés et la distribution réelle des classes.
— La matrice de confusion entre les labels réels et les clusters obtenus.

3.3 Optimisation du modèle DeBERTa

Une étape d'optimisation supplémentaire a été intégrée pour le modèle DeBERTa, visant à ajuster la longueur maximale des séquences (max_length). Initialement, la moitié de la valeur par défaut de BERT(512 tokens) était utilisée. Cependant, cette valeur était surdimensionnée et entraînait un gaspillage de ressources. Une analyse empirique a été menée :

1. Calcul de la distribution des longueurs réelles des séquences tokenisées.
2. Fixation du nouveau max_length au 95^{e} percentile de cette distribution afin de capturer l'essentiel de l'information tout en limitant les séquences inutiles.

Cette démarche a permis de ré-entraîner les représentations avec un max_length optimisé, améliorant ainsi légèrement le temps de calcul et la qualité des représentations obtenues.
Les étapes de visualisation et d'évaluation (t-SNE, K-Means, ARI, matrice de confusion) sont ensuite ré-appliquées sur ces nouvelles représentations vectorielles et donnent une légère amélioration. Ce qui en resulte reste néanmoins toujours moins performant que l'approche BERT sur nos données.

4 Sythèse des résultats

4.1 Résultats obtenus avec BERT

L'utilisation du modèle bert-base-uncased a permis d'extraire des représentations vectorielles pertinentes pour le clustering. L'algorithme K-Means appliqué sur les embeddings a produit des clusters relativement cohérents, visualisés en 2D via t-SNE (figure 1).

— La matrice de confusion révèle une séparation partielle mais acceptable des classes.
— L'Adjusted Rand Index (ARI) atteint un score de 0.301, ce qui constitue un résultat satisfaisant pour cette tâche non supervisée.
— Le temps de calcul est resté raisonnable, ce qui rend cette approche adaptée pour un usage efficace et rapide.

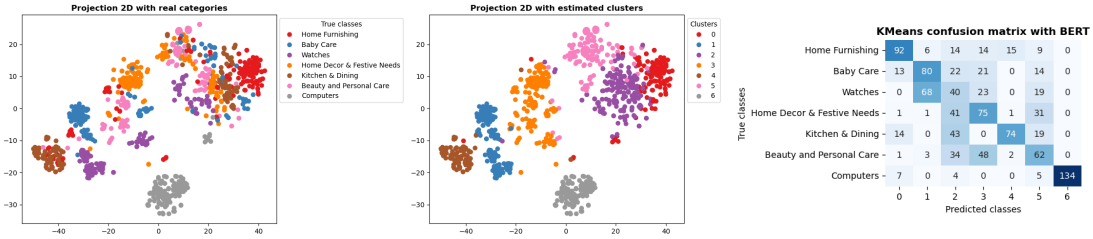


FIGURE 1 – Résultats avec BERT

4.2 Résultats obtenus avec DeBERTa-large (sans optimisation)

Avec le modèle microsoft/deberta-large, nous nous attendions à une amélioration des performances grâce à sa capacité de modélisation plus importante. Toutefois, les résultats obtenus sont nuancés :

— La visualisation t-SNE n'a pas montré d'amélioration claire des clusters par rapport à BERT.
— La matrice de confusion présente des performances similaires, sans gain significatif.
— L'ARI est inférieur à celui de BERT, avec une valeur de 0.296.
— En revanche, le temps d'exécution a fortement augmenté en raison du traitement de séquences longues (*max_length* = 256).

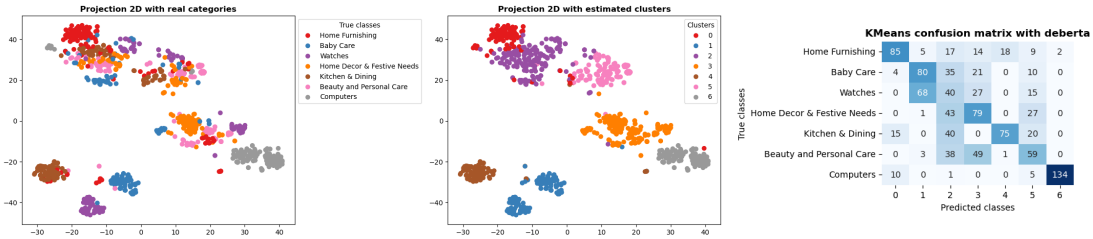


FIGURE 2 – Résultats avec DeBERTa

4.3 Résultats obtenus avec DeBERTa-large optimisé

L'optimisation du max_length en l'ajustant au 95^{e} percentile de la distribution des longueurs de séquences n'a pas permis de réduire le coût computationnel par rapport au DeBERTa sans optimisation, et n'a pas non plus permis de compenser la baisse de performance :

— Le score ARI, bien qu'il soit amélioré, reste inférieur à celui de BERT et la qualité des clusters demeure toujours moins performante
— La matrice de confusion et la visualisation t-SNE confirment cette observation.

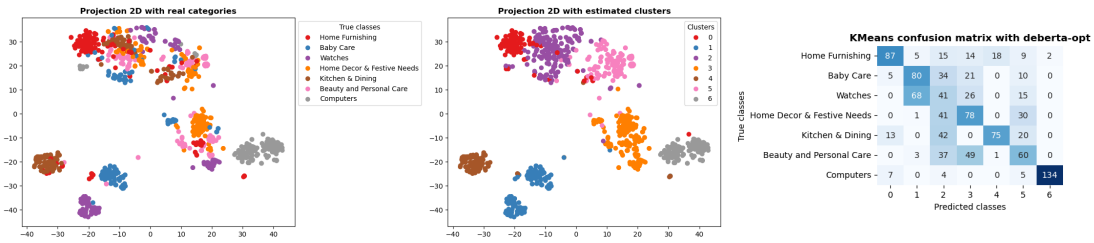


FIGURE 3 – Résultats avec DeBERTa optimisé

4.4 Analyse comparative et conclusion

Cette étude met en évidence que, dans notre cas de figure :

- DeBERTa-large n'améliore pas les performances par rapport à BERT malgré sa plus grande capacité théorique.
- Son temps d'exécution est nettement supérieur, ce qui pose problème en pratique.
- BERT obtient un ARI supérieur tout en étant plus léger et plus rapide.

Conclusion : Dans ce contexte spécifique, l'approche BERT apparaît plus adaptée que DeBERTa, grâce à un compromis plus favorable entre performance et coût computationnel. Cette analyse souligne l'importance d'évaluer l'apport des modèles récents, qui ne garantissent pas toujours de meilleures performances, notamment lorsque le jeu de données ou la tâche ciblée ne justifient pas forcément son usage.

5 Feature importance globale/locale

Nous entrainons un modèle de classification proxy, un RandomForestClassifier, basé sur les clusters obtenus avec KMeans puis analysons l'importance des features avec SHAP et LIME.

5.1 Feature importance globale avec SHAP

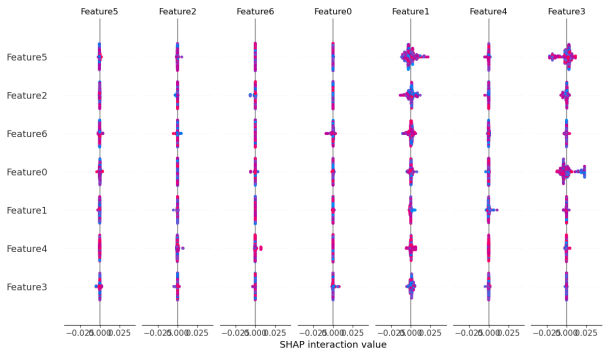


FIGURE 4 – Feature importance globale

- Globalement, ce sont les caractéristiques *Feature*0, *Feature*1, *Feature*2, *Feature*3, *Feature*4, *Feature*5 et *Feature*6 qui impactent le plus sur la classification basée sur les clusters obtenues avec KMeans.
- Cependant, dans notre cas de figure, il nous est difficile de dégager une analyse fiable de l'apport de chaque caractéristique sur la performance globale du modèle au vu des résultats obtenus avec SHAP.

5.2 Feature importance locale avec LIME

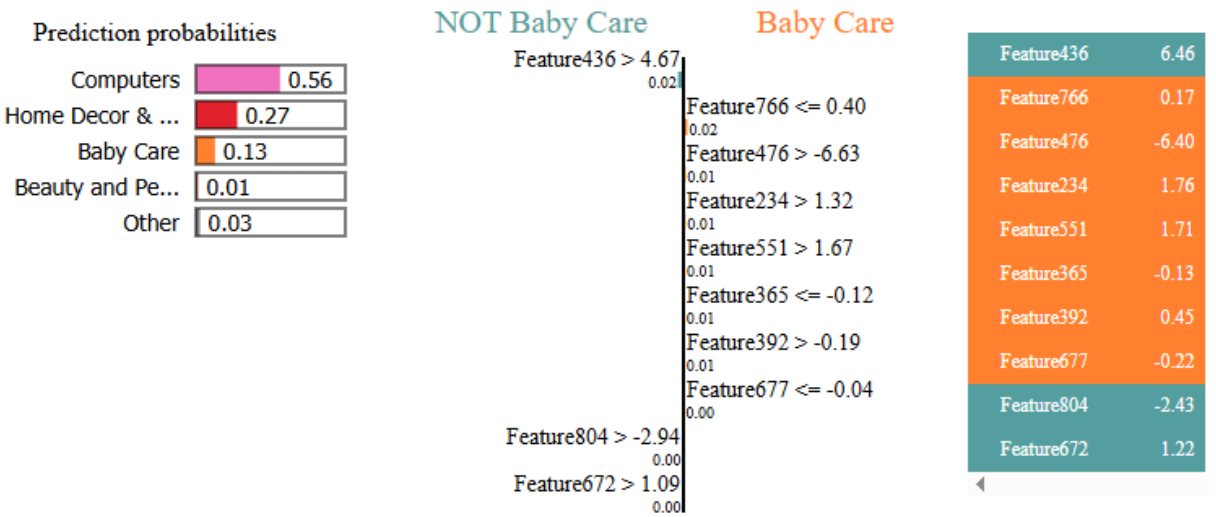


FIGURE 5 – Feature importance globale

Ce graphique montre une prédiction locale pour un article donné, ainsi que les probabilités de classification. Il présente l'impacte local des caractéristiques sur la prédiction.

- Le modèle prédit avec une probabilité de 0.56 que l'article appartient à la catégorie "Computers".
- Lorsque $Feature436 > 4.67$, $Feature804 > -2.94$ et $Feature672 > 1.09$, il y a de fortes chance que cet article soit prédit dans le cluster "Computers".

6 Limites et améliorations

6.1 Limites de l'approche actuelle

L'approche DeBERTa présente plusieurs limites identifiées au cours de cette expérimentation :

1. Temps de calcul élevé : Le modèle DeBERTa-large étant particulièrement volumineux, le temps de traitement est très long lors l'extraction des embeddings.
2. Pas d'amélioration des performances par rapport à BERT : Malgré sa complexité, le modèle n'a pas surpassé BERT sur les métriques étudiées (ARI et matrice de confusion). Cette observation peut s'expliquer par la taille du dataset, la nature de la tâche ou un besoin de réglage plus fin des hyper-paramètres.
3. Méthode de pooling basique : L'utilisation du mean pooling (moyenne des embeddings de tokens) pourrait ne pas exploiter pleinement la richesse des représentations contextuelles de DeBERTa.

6.2 Pistes d'amélioration envisageables

Pour renforcer la qualité des résultats et mieux exploiter les capacités de DeBERTa, plusieurs améliorations peuvent être envisagées :

1. Optimisation avancée du modèle :
   - Tester d'autres stratégies de pooling, comme le CLS pooling.
   - Explorer des modèles plus légers de la famille DeBERTa, par exemple deberta-base, afin de réduire le temps de calcul tout en conservant la capacité du modèle.
2. Augmentation de données et prétraitement : Enrichir ou nettoyer davantage le corpus pour favoriser l'apprentissage de représentations plus discriminantes.

Références

Pre-training of deep bidirectional transformers for language understanding. *BERT's original paper, Arxiv*. URL https://arxiv.org/pdf/1810.04805.

Huggingface documentation of deberta. URL https://huggingface.co/docs/transformers/en/model_doc/deberta.

All released versions of deberta. *Github releases*. URL https://github.com/microsoft/deberta.

Deberta : Machine learning paper explained. URL https://www.youtube.com/watch?v=_c6A33Fg5Ns.

Decoding-enhanced bert with disentangled attention. *DeBERTa's original paper, Arxiv*, 2021. URL https://arxiv.org/pdf/2006.03654.