**Note méthodologique : preuve de concept**

**DeBerta**

Table des matières

[I. Dataset retenu 1](#_Toc186063271)

[II. Les concepts de l’algorithme récent 1](#_Toc186063272)

[III. La modélisation 2](#_Toc186063273)

[IV. Une synthèse des résultats 4](#_Toc186063274)

[V. L’analyse de la feature importance globale et locale du nouveau modèle 5](#_Toc186063275)

[VI. Les limites et les améliorations possibles 6](#_Toc186063276)

# **Dataset retenu**

Le dataset utilisé pour ce projet se concentre sur des données e-commerces liés à des produits de consommation variés. Ce dataset inclut plusieurs caractéristiques pertinentes pour la classification de produits, notamment les colonnes suivantes :

* main\_category : anciennement product\_category\_tree , la catégorie principale a été conservé.
* product\_name : le nom du produit.
* brand : la marque associée au produit.
* Uniq\_id : l’id du produit.
* Image : l’intitulée de l’image
* cleaned\_description : anciennement description, il s’agit de la description qui a été nettoyée et prétraitée pour être utilisée dans les modèles NLP.

# **Les concepts de l’algorithme récent**

Source : <https://arxiv.org/abs/2006.03654v6>

DeBERTa (Decoding-enhanced BERT with disentangled attention) est une amélioration de BERT et RoBERTa. Son architecture introduit deux concepts clés :

Mécanisme d’attention désassemblé : Contrairement à BERT où chaque mot est représenté par un seul vecteur, DeBERTa représente chaque mot avec deux vecteurs distincts : un vecteur de contenu (content embedding) et un vecteur de position (position embedding). Le vecteur de contenu capture la signification sémantique du mot tandis que le vecteur de position encode la position relative du mot dans la phrase. Cette approche permet au modèle d'apprendre des relations plus fines entre les mots et leur contexte, ce qui améliore la compréhension des relations syntaxiques et sémantiques complexes.

La formule mathématique qui permet de calculer l'attention entre les tokens 𝑖 et 𝑗 dans DeBERTa est la suivante :

Une image contenant Police, texte, blanc, écriture manuscrite

Description générée automatiquement

Cela signifie que le poids d'attention entre un couple de mots est calculé en prenant la somme de quatre scores d'attention distincts utilisant des matrices désassemblées sur leur contenu et leurs positions : contenu-à-contenu, contenu-à-position, position-à-contenu, et position-à-position.

Enhanced Mask Decoder (EMD) : DeBERTa introduit un décodeur de masque amélioré (EMD) pour traiter les positions absolues. Contrairement à l’attention relative classique, l’EMD permet d’intégrer des informations de position absolue lors de la décodification des masques, ce qui renforce la capacité à comprendre les relations entre les mots distants dans le texte.

# **La modélisation**

La modélisation s'appuie sur deux approches principales pour l'analyse des descriptions des produits e-commerce : une approche classique basée sur TF-IDF, Word2vec, Bert et une approche avancée utilisant le modèle DeBERTa. Ces deux approches permettent de comparer les performances de méthodes traditionnelles et modernes de traitement du langage naturel (NLP).

**Approche Classique**

L’approche classique s’appuie sur des représentations textuelles comme TF-IDF, Word2Vec et BERT, combinées avec l’algorithme de clustering K-Means. Les étapes principales sont les suivantes :

Les descriptions des produits sont transformées en vecteurs à l’aide des méthodes mentionnées, puis standardisées à l’aide de StandardScaler pour garantir une homogénéité des échelles. Une réduction de dimension via PCA (Analyse en Composantes Principales) est appliquée pour conserver les 50 dimensions les plus significatives, réduisant ainsi la complexité computationnelle tout en préservant l’essentiel de la variance des données. Ensuite, une projection en deux dimensions via T-SNE permet de visualiser les relations entre les produits et d’évaluer la qualité des clusters. L’algorithme K-Means est ensuite utilisé pour regrouper les produits en 7 clusters, correspondant aux catégories réelles.

**Optimisation de l'ARI pour l'approche classique**

Dans l’approche classique, l’optimisation de l’Adjusted Rand Index (ARI) repose sur l’utilisation d’une recherche systématique des hyperparamètres grâce à GridSearchCV. Cette méthode permet d’explorer différentes configurations pour la vectorisation des données textuelles et les étapes de réduction de dimension (PCA, T-SNE). À chaque itération, l’ARI est calculé pour évaluer la cohérence des clusters générés par K-Means par rapport aux catégories réelles. La meilleure combinaison d’hyperparamètres est ensuite retenue pour maximiser l’ARI, garantissant ainsi une séparation optimale entre les clusters.

**Approche Avancée (DeBERTa)**

La modélisation avec DeBERTa commence par la préparation des données. La colonne cleaned\_description est nettoyée et transformée pour créer un texte d'entrée sous la forme classify: [texte]. Cette étape formate les descriptions pour qu'elles soient compatibles avec le modèle DeBERTa. Chaque description est ensuite tokenisée à l'aide du tokenizer DeBERTa, générant les tenseurs input\_ids, attention\_mask et labels, qui servent d'entrée pour l'entraînement. L’entraînement s’effectue sur un jeu d’entraînement représentant 80 % des données, et les performances sont évaluées sur un jeu de validation couvrant les 20 % restants.

Le modèle DeBERTa, chargé depuis la bibliothèque HuggingFace Transformers, est configuré pour une classification multi-classe avec 7 labels correspondant aux catégories de produits identifiées. Une optimisation des hyperparamètres est réalisée à l’aide de GridSearch, en testant différentes valeurs pour le taux d’apprentissage (2e-5, 3e-5), le nombre d’époques (3, 5) et la taille des batchs (16). Chaque configuration est évaluée en fonction de l’ARI obtenu sur le jeu de validation. Le modèle offrant le meilleur ARI est sélectionné et sauvegardé. Après l’entraînement, les embeddings sont extraits de la dernière couche du modèle et réduits en dimension via T-SNE. Ces représentations bidimensionnelles sont ensuite regroupées avec K-Means, et les clusters sont comparés aux catégories réelles pour évaluer leur cohérence.

**Métriques utilisées**

L’ARI est utilisé comme métrique principale pour évaluer la qualité des regroupements, en mesurant leur alignement avec les catégories réelles tout en ajustant les regroupements aléatoires. En complément de l’ARI, des métriques comme la précision, le rappel et le score F1 sont utilisées pour analyser les performances globales des modèles et évaluer leur capacité à classifier correctement les descriptions.

# **Une synthèse des résultats**

Une image contenant texte, Police, reçu, blanc

Description générée automatiquement

DeBERTa surpasse de loin l’ensemble des approches classique en termes de performance globale, comme le démontrent les métriques ARI, précision, rappel, et F1-score, où DeBERTa atteint des scores proches de 1, indiquant une très haute précision dans la classification. Par rapport à TF-IDF, Word2Vec et BERT, DeBERTa se distingue non seulement par sa précision moyenne (0.990) mais aussi par sa robustesse mesurée par un ARI élevé (0.876), montrant qu'il capte mieux la structure des données.

Cependant, cette supériorité se fait au prix d'un temps d'exécution beaucoup plus élevé pour l'entraînement. En revanche, pour des tâches nécessitant une réponse rapide et légère, l'approche Word2vec ou Bert peut être privilégiée.

# **L’analyse de la feature importance globale et locale du nouveau modèle**

**Importance globale des tokens du modèle DeBERTa**

L’analyse globale des tokens montre les mots ayant les poids les plus élevés dans l'ensemble des données. Ces poids représentent la contribution relative de chaque token à la classification globale selon le modèle DeBERTa.

Parmi les tokens les plus influents, on retrouve les 10 plus influent :

Une image contenant texte, reçu, Police, algèbre

Description générée automatiquement

Les tokens comme Ġapparel, Ġdenim, Ġprimary, et Ġoccasion reflètent des termes qui apparaissent fréquemment dans les descriptions de produits et sont importants pour différencier les catégories.

**Importance locale des tokens du modèle DeBERTa**

Dans l’analyse locale, les tokens les plus influents pour un produit donné sont identifiés. Cela permet de comprendre quels mots spécifiques ont le plus contribué à la prédiction pour ce produit.

Parmi les tokens les plus influents pour l’index 1, on retrouve les 10 plus influent :

Une image contenant texte, Police

Description générée automatiquement

Cela suggère que le modèle s'appuie sur des descripteurs clés comme "cotton", "material" ou « bath », qui semblent particulièrement importants pour déterminer la catégorie de ce produit.

# **Les limites et les améliorations possibles**

L'approche DeBERTa présente des défis liés à son temps de calcul élevé (14 696.51 secondes pour ce projet).

Des améliorations sont possibles, notamment en limitant le nombre de tokens à 128 ou en utilisant des versions allégées comme DeBERTa-small pour réduire les coûts computationnels. L’adoption de modèles distillés, tels que TinyBERT ou DistilBERT, permettrait également de conserver des performances élevées tout en diminuant le temps d’entraînement.

Enfin, des approches complémentaires, comme l’augmentation du dataset via des techniques textuelles, l’évaluation sur des jeux de données supplémentaires, ou l’hybridation des scores TF-IDF avec les embeddings de DeBERTa, pourraient maximiser la robustesse et la généralisation du modèle.