Projet de data-science DataScientest

Promotion DEC23_BOOTCAMP_DS

Projet

Thibaud BENOIST, Julien CHANSON, Julien FOURNIER, Alexandre MANGWA

Mentor: Yanniv

Résumé

Dans ce projet, l'objectif était d'évaluer la capacité de diverses méthodes de classification à catégoriser sur la base de textes et d'images des produits issus d'un catalogue de marketplace. Les modèles spécialisés traitant séparément le texte et les images ont d'abord été benchmarkés avant d'être fusionnés dans des classificateurs hybrides. La combinaison par simple vote majoritaire du modèle hybride (un transformer fusionnant BERT et ViT) avec plusieurs modèles spécialisés (BERT et XGBoost pour le texte; ViT et ResNet pour les images) a permis d'atteindre un score *weighted-F1* de 0.911 après entraînement sur 80% des données. Ce score est très satisfaisant et figure parmi les scores les plus élevés des classements publics et privés du site du challenge dont est issu ce jeu de données.

Repository: https://github.com/DataScientest-Studio/dec23_bds_rakuten

Introduction

L'objectif d'une marketplace est de mettre en relation vendeurs et acheteurs par le biais d'une plateforme en ligne unique, simplifiant ainsi le processus d'achat et de vente. Pour qu'elle soit efficace, il est crucial que les produits soient aisément identifiables, que les utilisateurs bénéficient d'une navigation fluide tout au long de leur parcours d'achat et que la plateforme offre des recommandations personnalisées. Un aspect essentiel du bon fonctionnement d'une marketplace est donc l'organisation méthodique du catalogue en catégories spécifiques, facilitant la recherche et la découvrabilité des produits.

Ce processus de classification des produits requiert l'application de techniques de machine learning (ML) pour plusieurs raisons. Premièrement, les vendeurs pourraient ne pas assigner les nouveaux produits aux catégories pertinentes, introduisant des erreurs dans l'organisation du catalogue. Par ailleurs, le classement manuel des produits peut s'avérer fastidieux et inefficace lors de l'ajout d'articles en batch. Enfin, les catégories peuvent être amenées à être modifiées, impliquant une mise à jour sur l'ensemble du catalogue. L'utilisation de technique de machine learning permet de surmonter ces problèmes en automatisant la catégorisation des produits sur la base des descriptions et images fournies par les vendeurs.

Le dataset utilisé dans ce projet est issu d'un challenge proposé par le groupe Rakuten, un acteur majeur du marketplace B2B2C. Il se compose d'un catalogue d'environ 80.000 produits, répartis selon 27 catégories distinctes, accompagnés de leurs descriptions textuelles et images correspondantes.

L'objectif de ce projet est de développer un modèle prédictif capable de classifier précisément (sur la base du score *weighted-F1*) chacun des produits en se basant sur les descriptions et images fournies.

Résultats

Le jeu de données consiste en 84.916 entrées du catalogue de la marketplace Rakuten. Chaque produit est caractérisé par les informations suivantes (**Figure 1**):

- ***"désignation"***: un titre court (<250 caractères)
- ***"description"***: description plus longue (sans limite de taille apparente) et facultative (absente pour 35.3% des produits)
- **"productid"** : identifiant unique du produit (clé primaire du dataset)
- ***"imageid"***: identifiant renvoyant à l'image correspondante (résolution 500 x 500).
- ***"prdtypecode"***: la catégorie à laquelle le produit appartient .

Les textes, principalement en français (plus de 80%) incluent également des entrées en anglais et en allemand.

Chaque entrée du dataset est associée à un identifiant "*productid*" unique. Cependant, cet identifiant ne distingue pas nécessairement des produits différents: 2392 entrées (2.8%) sont en réalité des réplicats (répliqués de 2 à 17 fois) avec *désignation* et *description* identiques, qui ne diffèrent que par l'image associée. La majorité de ces réplicats correspondent au même produit, listé plusieurs fois avec différentes images.

La présence de ces produits répliqués est intéressante car elle nous permet d'estimer la fiabilité des labels attribués par Rakuten dans ce jeu de données. Parmi les 2392 réplicats identifiés, 43 produits se trouvent dans des catégories différentes bien qu'ayant les mêmes désignation et description. On peut

donc en déduire que notre jeu de données contient un certain nombre d'erreurs: au maximum, 1.8% des produits (43 / 2392) pourraient être catégorisés de manière inexacte.

	prdtypecode	designation	description	productid	imageid
0	10	Olivia: Personalisiertes Notizbuch / 150 Seite...	NaN	3804725264	1263597046
1	2280	Journal Des Arts (Le) N° 133 Du 28/09/2001 - L...	NaN	436067568	1008141237
2	50	Grand Stylet Ergonomique Bleu Gamepad Nintendo...	PILOT STYLE Touch Pen de marque Speedlink est ...	201115110	938777978
3	1280	Peluche Donald - Europe - Disneyland 2000 (Mar...	NaN	50418756	457047496

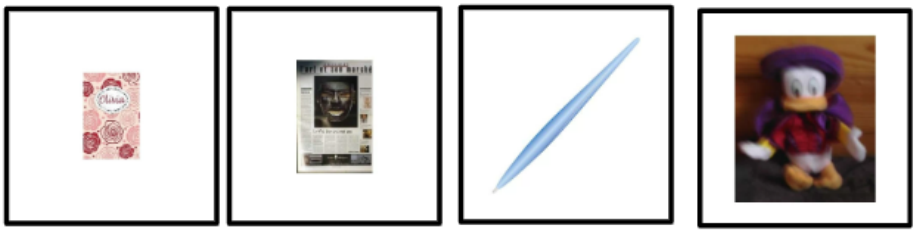


Figure 1: Extrait des premières lignes du dataset Rakuten et images correspondantes

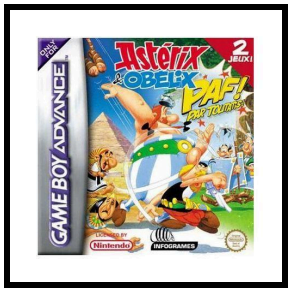
Pre-processing

Afin d'améliorer la qualité du texte, nous avons procédé aux corrections suivantes (Figure 2):

- Suppression des balises HTMLs.
- Suppression des caractères spéciaux.
- Correction des erreurs d'encodage.
- Traduction des textes anglais et allemand en français.
- Fusion des champs *désignation* et *description*

Les images (500 x 500) n'ont été prétraitées que pour uniformiser le padding:

- Cropping de l'image non-padded
- Redimensionnement à 500 pixels en conservant l'aspect ratio
- Padding constant (255, 255, 255) pour atteindre 500 x 500

Avant preprocessing	Après preprocessing
	

Avant preprocessing	Après preprocessing
Horloge murale en bois antique de style vintage pour le bureau de cuisine **¿¿** domicile **Caract¿¿ristique**: 100% neuf et de haute **qualit¿¿.?Quantit¿¿**: 1pc.**?Mat¿¿riau**: […] Fonctionne parfaitement en silence et conserve **l'**heure exacte Cette horloge murale vintage peut **¿°tre** un bon cadeau pour les pendaisons de **cr¿¿maill¿¿re** les mariages et les **r¿¿unions** sociales Emballage inclus: Horloge murale en bois antique style 1xVintage	Horloge murale en bois antique de style vintage pour le bureau de cuisine **a** domicile **caractéristique**: 100% neuf et de haute **qualité**.**quantité**: 1pc.matériau: [...] Fonctionne parfaitement en silence et conserve **l'**heure exacte Cette horloge murale vintage peut **être** un bon cadeau pour les pendaisons de **crémaillère** les mariages et les **réunions** sociales Emballage inclus: Horloge murale en bois antique style 1x Vintage

Figure 2: Exemple d'entrée avant et après preprocessing du texte et de l'image. Les mots mal encodés et leur correction sont surlignés

Analyse descriptive

Déséquilibre de classes (Figure 3). Les catégories de produit du jeu de données affichent un déséquilibre notable, allant de moins de 100 articles pour certaines catégories telles que figurines, confiserie ou vêtements pour enfants, jusqu'à plusieurs milliers d'articles pour des catégories comme le mobilier ou les accessoires de piscine.

Déséquilibre de textes (Figure 4 et 5). La longueur des descriptions textuelles montrent une variabilité importante. Les descriptions des livres d'occasion ou des cartes de jeux sont généralement brèves (quelques dizaines de mots, champ description absent), tandis que celles des jeux vidéo pour PC s'étendent souvent sur plusieurs centaines de mots.

Variabilité des langues (Figure 6). Bien que nous ayons fait le choix de traduire l'ensemble du jeu de données vers une langue unique (francais), on remarque que la langue varie significativement selon la catégorie de produit.

Séparabilité des catégories (Figure 7 et 8). Certaines catégories ont un chevauchement lexical notable (par exemple, les consoles de jeu et les jeux vidéo), comme on peut le remarquer dans les wordclouds (Figure 7) ou dans la matrice de corrélation entre vecteurs de fréquence des mots (Figure 8).

Classification du texte

Approches classiques

Dans le contexte de la classification de produits sur la base du texte seul, nous avons commencé par examiner différentes techniques de vectorisation (Bag-of-Words avec *TF-IDF* et Word2Vec avec *Skipgram* et *CBOW*) associées à des méthodes de classification classiques (SVM, régression logistique, arbres de décision, etc).

L'entraînement s'est effectué sur 80% des données, avec une évaluation des performances sur les 20% restants. Nous avons optimisé les hyper-paramètres (e.g. taille du vecteur d'embedding pour Word2Vec ou paramètres de régularisation pour SVM, etc) via une recherche exhaustive avec validation croisée à 5 folds sur l'ensemble d'entraînement.

Ce premier benchmark indique que la vectorisation Bag-of-Words (*TF-IDF*) combinée à LinearSVC ou xgBoost surpasse les méthodes Word2Vec, avec un **F1-score de 0.824** pour LinearSVC basée sur *TF-IDF* (Figure 9, Table 1).
Les matrices de confusion révèlent la difficulté de ces modèles à différencier des catégories sémantiquement proches, telles que (Figure 10):

- "Maison Décoration", "Mobilier de jardin", "Mobilier", "Outillage de jardin", "Puériculture"
- "Figurines et jeux de rôle", "Figurines et objets pop culture", "Jouets enfants", "Jeux de société pour enfants"
- "Livres d'occasion", "Livres neufs", "Magazines d'occasion", "Bandes dessinées et magazines"
- "Jeux vidéo d'occasion", "CDs et équipements de jeux vidéo", "Accessoires gaming"

Algorithme	F1 score	durée fit (s)[1]
LinearSVC	**0.824**	6.1
XgBoost	**0.819**	3840
Logistic Regression	**0.813**	179
SVC	**0.784**	2993
RandomForest	**0.776**	2344
MultinomialNB	**0.771**	0.45

Table 1: Synthèse des benchmarks de machine-learning basé sur une vectorisation TF-IDF

Transformers

L'emploi de modèles basés sur les transformers dans la résolution de problèmes de classification de texte est devenu incontournable. Nous avons donc poursuivi notre stratégie de classification textuelle en entraînant des transformers de type **BERT** (Bidirectional Encoder Representations from Transformers).

Plusieurs versions de transformers pré-entraînés sur divers corpus français ont été comparées: **CamemBERT-base**, **CamemBERT-ccnet** et **FlauBERT**. Chaque modèle a été complété par une tête de classification comprenant une couche dense de 128 unités suivie d'un dropout de 20%, avant d'arriver à la couche finale de classification. Les modèles ont été entraînés sur 80% des données et testés sur les 20% restants (même partition que mentionnée précédemment). Nous avons affiné les poids de ces transformers sur 8 époques d'entraînement, avec un taux d'apprentissage initial de 5e-5 réduit de 20% à chaque époque.

Les résultats montrent une nette supériorité de ces modèles transformer (**camemBERT**, **F1-score = 0.886,** Table 2), augmentant le f1-score de 7.5% par rapport au meilleur modèle de notre précédent benchmark (linearSVC sur TF-IDF, Figure 11). Les performances sont relativement équivalentes entre modèles BERT pré-entraînés sur différents corpus, avec néanmoins une légère avance pour les modèles camemBERT.

[1] benchmark réalisés sur un google colab CPU standard

L'examen des matrices de confusion révèle que l'utilisation de modèles transformers réduit le taux d'erreur sur les catégories sémantiquement proches (Figure 12). Néanmoins, ce sont les mêmes catégories qui posent toujours problème. L'analyse de cas spécifiques de classifications incorrectes met en lumière la complexité inhérente à cette tâche: il est difficile de déterminer à la simple lecture du texte la catégorie associée à ces produits (Table 3).

Modèle	F1 score	vitesse fit (s)
CamemBERT CCNET	**0.886**	16955
CamemBERT	**0.885**	17225
FlauBERT	**0.878**	15138

Table 2: Synthèse des benchmarks pour les transformer de type BERT

Classification des images

CNN et Vision transformer

Dans le domaine de la classification d'images, l'adoption de réseaux de deep learning est incontournable. Les réseaux de neurones convolutifs (CNN) sont particulièrement performant mais plus récemment, les modèles basés sur des architectures de transformer, comme le modèle Vision Transformer (ViT) se sont aussi révélés efficaces.
Pour classifier les produits sur la base des images associées, nous avons donc utilisé différents réseaux convolutifs (**ResNet**, **EfficientNet** et **VGG**) ainsi que le transformer **ViT** (Vision Transformer), tous pré-entraînés sur la base de données ImageNet.
Tout comme les modèles BERT pour le traitement de texte, chaque modèle de classification d'images a été doté d'une tête de classification comprenant une couche dense de 128 unités suivie d'un dropout de 20%, menant à la couche de classification finale. L'entraînement a suivi une démarche similaire à celle employée pour les modèles BERT: entraînement sur 80% des données, évaluation sur les 20% restants (même partition que pour le texte), avec un fine-tuning des poids sur 8 époques d'entraînement et un taux d'apprentissage initial de 5e-5, réduit de 20% à chaque époque.
Les F1-scores mesurés sur l'ensemble de test révèlent une supériorité marquée du modèle Vision Transformer (**ViT, F1-score = 0.675**) comparativement au meilleur modèle CNN testé (**ResNet152, F1-score = 0.658,** Table 4). Ces modèles image restent cependant beaucoup moins performant que les modèles texte, illustrant la complexité inhérente à la classification de produits sur la base exclusive d'images. Néanmoins, il est intéressant de noter que les catégories les plus fréquemment confondues par les modèles dédiés aux images correspondent presque exactement à celles posant des difficultés dans la classification de texte (Figure 13).

Modèle	F1 score	vitesse fit (s)
ViT_b16	**0.675**	10572
ResNet152	**0.658**	6894
ResNet101	**0.656**	6754
EfficientNetB1	**0.655**	6657
ResNet50	**0.653**	6720
VGG16	**0.620**	6054

Table 4: Synthèse des benchmarks pour les convNet et transformer dans la classification des images

Classification multimodale simple

Dans un contexte de classification multimodale, il paraît raisonnable d'anticiper une amélioration des performances en exploitant conjointement les données texte et image. Toutefois, compte tenu de la disparité observée entre les performances des modèles texte et image, ainsi que la similitude des catégories souvent confondues, des questions se posent quant à l'amélioration de performance à laquelle on peut s'attendre avec un modèle hybride.

Pour cette partie, nous avons retenu les deux meilleurs architectures obtenues sur le texte et les images (i.e. camemBERT et ViT). Plusieurs architectures de modèles effectuant la fusion entre ces deux transformers ont ensuite été testées et comparées:

- **Approches d'ensemble**: classifiers de type **voting** ou **stacking** (par régression logistique) opérant sur les logits de sortie des modèles spécialisés pré-entraînés. Le poids attribué à chaque modèle dans le voting classifier est défini par le rapport des F1-scores des modèles spécialisés (par exemple: **camemBERT**: $\text{F1-score}^{\text{BERT}}$ / ($\text{F1-score}^{\text{BERT}}$ + $\text{F1-score}^{\text{ViT}}$) = **0.57**; **ViT**: $\text{F1-score}^{\text{ViT}}$ / ($\text{F1-score}^{\text{BERT}}$ + $\text{F1-score}^{\text{ViT}}$) = **0.43**). Pour éviter tout leakage des F1-scores utilisés comme poids du modèle, la performance du voting classifier est estimée par validation croisée à 5 folds sur l'ensemble de test.
- **Approche transformer (TF)**: fusion des sorties des derniers blocs de transformer de camemBERT et ViT par l'intermédiaire d'un bloc transformer cross-attentionnel, suivi d'un nombre variable de blocs de transformer classiques avec self-attention (TF: 1, 3 ou 6 blocs).

Pour l'approche transformer (TF), nous avons adopté la même démarche d'entraînement que pour les modèles BERT et ViT seuls: fine-tuning des poids sur 8 époques d'entraînement avec un taux d'apprentissage initial de 5e-5, réduit de 20% à chaque époque.

Les performances obtenues à partir des méthodes d'ensemble simples (voting ou stacking) montrent que la fusion des modèles spécialisés pré-entraînés n'apporte qu'une amélioration modeste (0.5%) à la précision des prédictions (0.891 vs 0,886). Comme mentionné précédemment, on pouvait peut-être s'y attendre étant donné que les modèles texte et images rencontrent des difficultés sur les mêmes catégories. Il est important de souligner que le score obtenu pour l'approche de stacking est probablement sous-évalué. En effet, par manque de temps, le fit de la régression logistique du modèle de

stacking a été réalisé sur le même ensemble de données que celui utilisé pour entraîner les modèles spécialisés, plutôt que par cross-validation. Une estimation par cross-validation aurait limité le biais induit par l'overfitting des estimateurs de base sur l'estimateur final.

L'approche transformer s'avère être la plus performante de toutes les stratégies hybrides testées, permettant une amélioration de 1.5% (**TF6**, **F1-score = 0.899**) par rapport au modèle texte seul (Table 5). On note néanmoins que le nombre de blocs de transformer après fusion ne semble pas contribuer à une nette amélioration des performances.

Modèle	F1 score	vitesse fit (s)
TF6	**0.899**	28874
TF3	**0.897**	24375
TF1	**0.899**	20773
Voting	**0.892**	NA
Stacking	**0.891**	1311

Table 5: Synthèse des benchmarks pour les modèles hybrides

Classification multimodale combinatoire

L'examen des matrices de confusion pour les modèles multimodaux "simples" indique que les mêmes catégories demeurent une source d'erreur (Figure 14). Cependant, une comparaison plus approfondie des F1-scores entre modèle hybride (Transformer) et spécialisés (camemBERT et ViT) suggère qu'une synergie entre ces modèles pourrait optimiser notre approche.

Pour tester cette hypothèse, nous avons établi un modèle de Voting, intégrant le modèle hybride et différents modèles spécialisés. Les poids attribués à chaque modèle ont été établis selon la même approche que celle utilisée pour le voting "simple" precedent, i.e. à partir des F1-scores de chaque modèle avec évaluation par cross-validation sur l'ensemble de test.

Dans un premier temps, nous avons combiné le modèle multimodal transformer (TF6) avec les deux modèles spécialisés correspondants (camemBERT-ccnet et ViT). Cette approche permet de gagner presque un point de F1-score par rapport au meilleur modèle hybride (TF6 + camemBERT + ViT, f1-score = 0.909).

Fort de cette avancée, nous avons envisagé d'élargir notre modèle de vote en y intégrant d'autres modèles précédemment évalués, explorant ainsi la potentialité d'une synergie plus large pour optimiser davantage les performances.

Le meilleur de ces modèles hybrides combine les classifiers suivants: TF6 (hybride), camembert-base-ccnet (texte), flaubert-base-uncased (texte), xgboost_tfidf (texte), vit_b16 (image), ResNet152 (image). Ce modèle permet de gagner plus d'un point de F1-score par rapport au modèle multimodale simple de type transformer (**F1-score = 0.911**, Table 6).
Comparativement au modèle optimal obtenu, nous avons évalué les performances d'alternatives (Table 6): 1) Un vote combinant TF6, CamemBERT et ViT s'approche très étroitement en termes de

performances tout en utilisant un modèle de moins; 2) L'association de TF6 avec les alternatives FlauBERT et ResNet152, par rapport aux modèles intégrés à TF6, permet de gagner presque 0.7% en comparaison avec le TF6; 3) Le meilleur vote sans TF6 atteint un plateau à 0.902; 4) Le meilleur vote sans modèle transformateur pour le texte atteint un plateau à 0.852.

Ensemble de modèle	F1 score
hybride: **TF6** texte: **camemBERT**, **FlauBERT**, **xgboost-tfidf** image: **ViT**, **ResNet152**	**0.911**
hybride: **TF6** texte: **camemBERT** image: **ViT**	**0.909**
hybride: **TF6** texte: **FlauBERT** image: **ResNet152**	**0.907**
texte: **camemBERT**, **FlauBERT** image: **ViT, ResNet152**	**0.902**
texte: **camemBERT**, **FlauBERT**, **xgboost-tfidf** image: **ViT**	**0.900**
texte: **text/SVC_skipgram, text/LinearSVC_tfidf, text/xgboost_tfidf** image: **ViT**	**0.852**

Table 6: Synthèse des benchmarks pour les modèles de voting incluant différentes combinaisons de modèles texte, image et hybride.

Discussion

Dans ce projet, nous avons évalué la capacité de diverses méthodes de classification à prédire les catégories de produits à partir de données textuelles et visuelles. Notre stratégie a impliqué l'établissement de benchmarks pour des modèles spécialisés traitant séparément les données textes et images, avant de fusionner les modèles les plus performants au sein de classificateurs hybrides. Parmi l'éventail de paramètres explorés, les meilleurs résultats ont été obtenus en combinant plusieurs modèles spécialisés (BERT pour le texte, XGBoost, ViT et ResNet pour les images) et un modèle hybride (un transformateur fusionnant BERT et ViT) via une méthode de vote majoritaire. Cette approche a permis d'atteindre un score de 0.911 après un entraînement sur 80 % des données.

Plusieurs aspects inexplorés de notre pipeline d'analyses auraient pu potentiellement améliorer nos résultats. Bien que l'importance de conserver les stop words et certains symboles ait été examinée pour les approches de vectorisation comme TF-IDF ou Word2Vec, nous n'avons pas évalué leur impact sur les performances de transformers. De même, certains hyperparamètres des réseaux de neurones tels que le learning rate, le drop-out ou la régularisation, n'ont été que partiellement examinés, avec une approche heuristique plutôt que systématique. Le travail sur le déséquilibre initial des classes pourrait aussi être approfondi : augmentation du jeu de données par scrapping de données complémentaires ou par génération de texte complémentaires (synonymes, …) ou undersampling.

L'exploration de différentes architectures de réseaux sur les performances de modèle hybride s'est limitée à la combinaison des sorties encodées de modèles profonds spécialisés. D'autres approches auraient pu être envisagées, telle que la fusion des features textes et images dès les premières couches du modèle ou l'incorporation de connexions résiduelles pour permettre une fusion à plusieurs niveaux tout en limitant le problème de vanishing gradient.

Plusieurs aspects de notre exploration étaient inattendus. En particulier le fait qu'une approche de type TF-IDF soit plus performante que des approches Word2Vec ou que la fusion de nombreux modèles par simple vote majoritaire puissent conduire à une améliorations aussi notable des performances par rapport à des modèles de type transformer qui se sont pourtant avérés être les plus performants pris indépendamment.

Le score maximal de 0.911 est très satisfaisant. Ce résultat dépasse les meilleurs scores affichés sur les leaderboard public et privé du site du challenge. Cependant, il ne parvient pas à égaler les performances présentées lors du workshop SIGIR dédié à l'e commerce, ou le score macro-F1 maximal était de 0.919.

Remerciements

Merci à Yanniv pour son soutien et ses conseils; à Aida pour sa disponibilité et son organisation; et à Axalia, pour nous avoir offert l'opportunité de collaborer sur ce projet passionnant.

Méthodes

Pre-processing

Les opérations de preprocessing suivantes sont effectuées séparément sur les colonnes désignation et description, avant leur fusion ultérieure qui ne s'effectue qu'après traduction. Cette approche est dictée par le fait que les champs "désignation" et "description" d'un même article n'ont parfois pas la même langue, rendant difficile l'étape de correction des exceptions d'encodage qui dépend de la langue détectée.

Transformations réalisées sur le texte

Opération	Description	Librairies utilisées
Parsing HTML	Retrait des balises HTML Conversion de caractères spéciaux HTML en UTF-8	BeautifulSoup / html.parser
Parsing XML	Retrait de balises XML résiduelles	BeautifulSoup / lxml
Cleaning regex	Suppression d'urls, de noms de fichiers. Ajout d'espaces manquants avant les majuscules	re
Language detection	Détection de la langue afin de cibler la traduction sur les lignes concernées. Nous avions aussi envisagé initialement d'ajouter une feature 'language' au dataset.	langdetect
Fixing encoding exceptions	Certains textes contiennent des anomalies d'encoding (apparition de caractères type ? en plein milieu d'un mot). Ajout d'un script spellchecker tentant de récupérer les mots erronés. A noter que la traduction permettait aussi de corriger certaines de ces anomalies	SpellChecker
Translating	Traduction en français de tous les textes identifiés comme anglais ou allemand.	Google translate

Toutes ces opérations ont fait l'objet de multiples tests :

- différents essais de regex pour affiner au mieux la correction d'anomalies
- comparaison de deux librairies de détection de langage : langid et langdetect
- essais de différents services de traduction (la plupart étant limités en mode gratuit)

Transformations réalisées sur les images

Opération	Description	Librairies utilisées
Cropping	Modification de la zone de padding originale pour retirer un maximum de blanc dans le format carré cible.	opencv
Sizing	Redimensionnement de l'image en conservant l'aspect ratio	opencv
Padding	Padding de l'image pour harmoniser les dimensions a une dimension finale de 224 x 224.	opencv

Création des ensembles d'entraînement et de test

Le jeu de données a finalement était partitionné en un **ensemble d'entraînement de 80%** des données et un **ensemble de test de 20%**. Cette partition a été faite de manière aléatoire, en respectant la proportion des classes dans le jeu de données complet. Les labels des classes ont également été encodés pour garantir leur compatibilité avec diverses méthodes de classification.

Vectorisation du texte

La vectorisation du texte transforme le texte en séquences de tokens (mots ou sous-mots) et associe à chaque token un vecteur compréhensible par l'algorithme de classification. Tokenisation et vectorisation varient selon la méthode employée:

- **Vectorisation TF-IDF**: La tokenisation se fait généralement par mots (avec ou sans racinisation ou lemmatisation). Le vocabulaire construit transforme chaque texte en vecteur où chaque dimension représente un token, avec des valeurs basées sur la fréquence du token dans le corpus, pondérée par l'inverse de sa fréquence dans l'ensemble de document (TF-IDF). **Dans notre projet nous avons utilisé les paramètres par défaut du TfidfVectorizer de sklearn (pas de limite à la taille du vocabulaire, conservation des stop words, sans racinisation).**

- **Word2Vec**: la segmentation est le plus souvent faite par mot. Une fois que le vocabulaire est établi, un index est attribue a chaque token. La vectorisation s'effectue par l'entraînement d'un réseau dense visant à prédire les tokens voisins (Skipgram) ou un token spécifique à partir de ses voisins (CBOW). **Dans notre projet, la taille optimale du vecteur d'embedding pour les méthodes skipgram et cbow a été estimée par cross-validation a 5 folds.**

- **Transformers**: la tokenisation est effectuée par un tokenizer spécifique, disposant d'un vocabulaire déjà établi et segmentant le texte en mots ou sous-mots. Chaque token est associé à un index spécifique puis transformé par la couche d'embedding pour donner une séquence de vecteurs. **Dans le cas des transformer utilisés dans notre projet, cet embedding correspond à des vecteurs de dimension 768.**

Vectorisation	Description	Librairies utilisées
Bag of words (TF-IDF)	Conversion des textes en vecteur de valeurs TF-IDF a partir de l'ensemble d'entraînement. Pas de limite de taille de vecteur. Valeur de TF-IDF normalisés par la norme euclidienne pour chaque entrée.	sklearn
Skip-gram	Ajustement du vocabulaire et des vecteurs d'embedding sur l'ensemble d'entraînement. Taille d'embedding cross-validée.	gensim
C-BOW	Ajustement du vocabulaire et des vecteurs d'embedding sur l'ensemble d'entraînement. Taille d'embedding cross-validée.	gensim
BERT	Tokenizer pré-entraîné, chargé depuis Hugging Face	transformer

Augmentation des images

Dans le cadre de l'entraînement des réseaux de neurones dédiés à la classification d'images, nous avons systématiquement augmenté notre jeu de données d'images en appliquant les mêmes paramètres d'augmentation : rotation aléatoire de +/- 20 degrés, translation aléatoire de 10%, et flip horizontal aléatoire. De plus, les images ont été re-scalées spécifiquement pour chaque réseau, conformément aux paramètres utilisés lors du pré-entraînement sur la base de données ImageNet.

Classification

Approches classiques

Tous les algorithmes de classification classiques sélectionnés proviennent de la bibliothèque sklearn, à l'exception de XGBoost. Les hyperparamètres de ces méthodes ont pour la plupart été optimisés grâce à une validation croisée à 5 folds sur l'ensemble d'entraînement.

Méthodes	Paramètres optimaux	Librairies utilisées
LinearSVC	C = 0.5	sklearn
LogisticRegression	TF-IDF: C = 2 CBOW, skipgram: C = 10	sklearn
SVC	TF-IDF: C = 0.1 CBOW, skipgram: C = 10	sklearn
RandomForest	n_estimators = 200, max_depth = 500	sklearn
MultinomialNB	TF-IDF: alpha = 0.02, fit_prior = True	sklearn
xgboost	n_estimators = 200, max_depth = 6	xgboost

Approches Transformer BERT et ViT

Pour le texte, nous avons utilisé des transformers de type **BERT**, pre-entrainés sur divers corpus francais (camemBERT-base, camemBERT-base-ccnet, FlauBERT-base_uncased).Ces modèles intègrent douze blocs de transformer. Chaque bloc est doté d'une multi-tête d'attention composée de douze couches, suivi d'une couche dense feed-forward, calibrée selon la dimension de l'embedding (768) (**Figure 15**).

Pour les images, nous avons utilisé le transformer **ViT** (vit-keras), dont l'architecture s'aligne sur celle de BERT, avec douze blocs de transformer et une taille d'embedding de 768. Les images sont préalablement segmentées en patchs (16 x 16 pixels pour le modèle utilisé dans ce projet) puis intégrées via une couche d'embedding, avant d'être transmises à travers la série de douze blocs de transformers.

Pour la tête de classification, nous avons ajouté une couche dense de 128 unités (activation de type relu) prenant la sortie associée au premier token de la séquence encodée (token CLS). La classification est finalement effectuée par une couche dense de 27 unités, avec activation softmax.

ConvNets

Différents réseaux convolutifs ont été utilisés pour la classification d'image: **VGG16**, **resnet50**, **resnet101**, **resnet152** et **EfficientNet**. Ces modèles, pré-entraînés sur ImageNet, sont aisément accessibles via TensorFlow.keras. Tout comme pour les transformers, la tête de classification utilisée est une couche

dense de 128 unités prenant la sortie de la derniere couche moyennée sur l'ensemble des positions spatiales (GlobalAveragePooling). La classification est ensuite effectuée par une couche dense de 27 unites avec activation softmax.

Modèle Hybride Transformer

Pour la classification multimodale, nous avons développé une architecture hybride fusionnant les transformers camemBERT et ViT. Cette fusion est réalisée via un bloc de transformer avec une tête cross-attentionnelle à 12 couches, dans laquelle la sortie encodée de camemBERT sert de requête (*query*), tandis que la sortie de ViT est utilisée comme clé et valeur (*key* et *value*). Ce bloc cross-attentionnel est suivi de un ou plusieurs blocs de transformers, similaires aux blocs de transformer utilisés dans BERT et ViT. La tête de classification reste identique à celle utilisée pour les modèles camembert et ViT spécialisés (**Figure 16**).

Entraînement.

L'entraînement de l'ensemble de nos réseaux de neurones, qu'ils soient basés sur une architecture Transformer ou ConvNet, a été uniformisé autour d'une procédure commune. L'ensemble des poids du modèle ont été ajustés à notre jeu de données au cours de 8 époques, avec drop de 20% précédent la couche de classification finale et un learning rate de 5e-5 décroissant de 20% à chaque époque.

Meta Voting.

Pour intégrer des modèles de différentes natures, nous avons adopté une stratégie de soft-Voting. Cette méthode consiste a pondérer les logits de sortie des différents modèles pour estimer la classe la plus probable pour chaque entrée. Les poids attribués à chaque modèle ont été définis sur la base de leur f1-scores (e.g. pour le modele *m*, $\text{weight}^m = \text{F1-score}^m / \text{sum}_i(\text{F1-score}^i)$), en cross-validant la performance du modèle par vote sur l'ensemble de test.

Analyse des performances

Les performances de nos modèles ont ete evaluees par le calcul du f1-score pondéré (*weighted* F1-score), i.e. le F1-score calculé par catégorie puis moyenné selon la proportion de chaque classe dans le jeu de données. Ce score a été systématiquement calculé grâce à la fonction *sklearn.metrics.f1_score* de sklearn afin d'éviter toute ambiguïté. Pour analyser plus en détails les performances, nous avons également calculé les matrices de confusion, exprimées en pourcentage du nombre d'articles par catégorie (normalisation par colonne).

Figures

	prdtypecode	designation	description	productid	imageid
0	10	Olivia: Personalisiertes Notizbuch / 150 Seite...	NaN	3804725264	1263597046
1	2280	Journal Des Arts (Le) N° 133 Du 28/09/2001 - L...	NaN	436067568	1008141237
2	50	Grand Stylet Ergonomique Bleu Gamepad Nintendo...	PILOT STYLE Touch Pen de marque Speedlink est ...	201115110	938777978
3	1280	Peluche Donald - Europe - Disneyland 2000 (Mar...	NaN	50418756	457047496

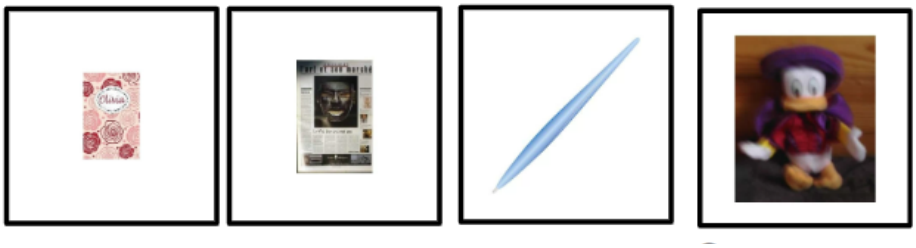


Figure1

Extrait des premières lignes du dataset Rakuten et images correspondantes (Retour au texte)

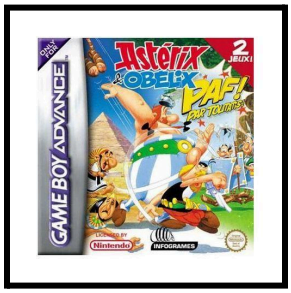
Avant preprocessing	Après preprocessing
	
Horloge murale en bois antique de style vintage pour le bureau de cuisine ¿¿ domicile **Caract¿¿ristique**: 100% neuf et de haute **qualit¿¿.?Quantit¿¿**: 1pc.**?Mat¿¿riau**: [...] Fonctionne parfaitement en silence et conserve **l'**heure exacte Cette horloge murale vintage peut **¿°tre** un bon cadeau pour les pendaisons de **cr¿¿maill¿¿re** les mariages et les **r¿¿unions** sociales Emballage inclus: Horloge murale en bois antique style 1xVintage	Horloge murale en bois antique de style vintage pour le bureau de cuisine **a** domicile **caractéristique**: 100% neuf et de haute **qualité**.**quantité**: 1pc.matériau: [...] Fonctionne parfaitement en silence et conserve **l'**heure exacte Cette horloge murale vintage peut **être** un bon cadeau pour les pendaisons de **crémaillère** les mariages et les **réunions** sociales Emballage inclus: Horloge murale en bois antique style 1x Vintage

Figure 2

Exemple d'entrée avant et après preprocessing du texte et de l'image. Les mots mal encodés et leur correction sont surlignés (Retour au texte).

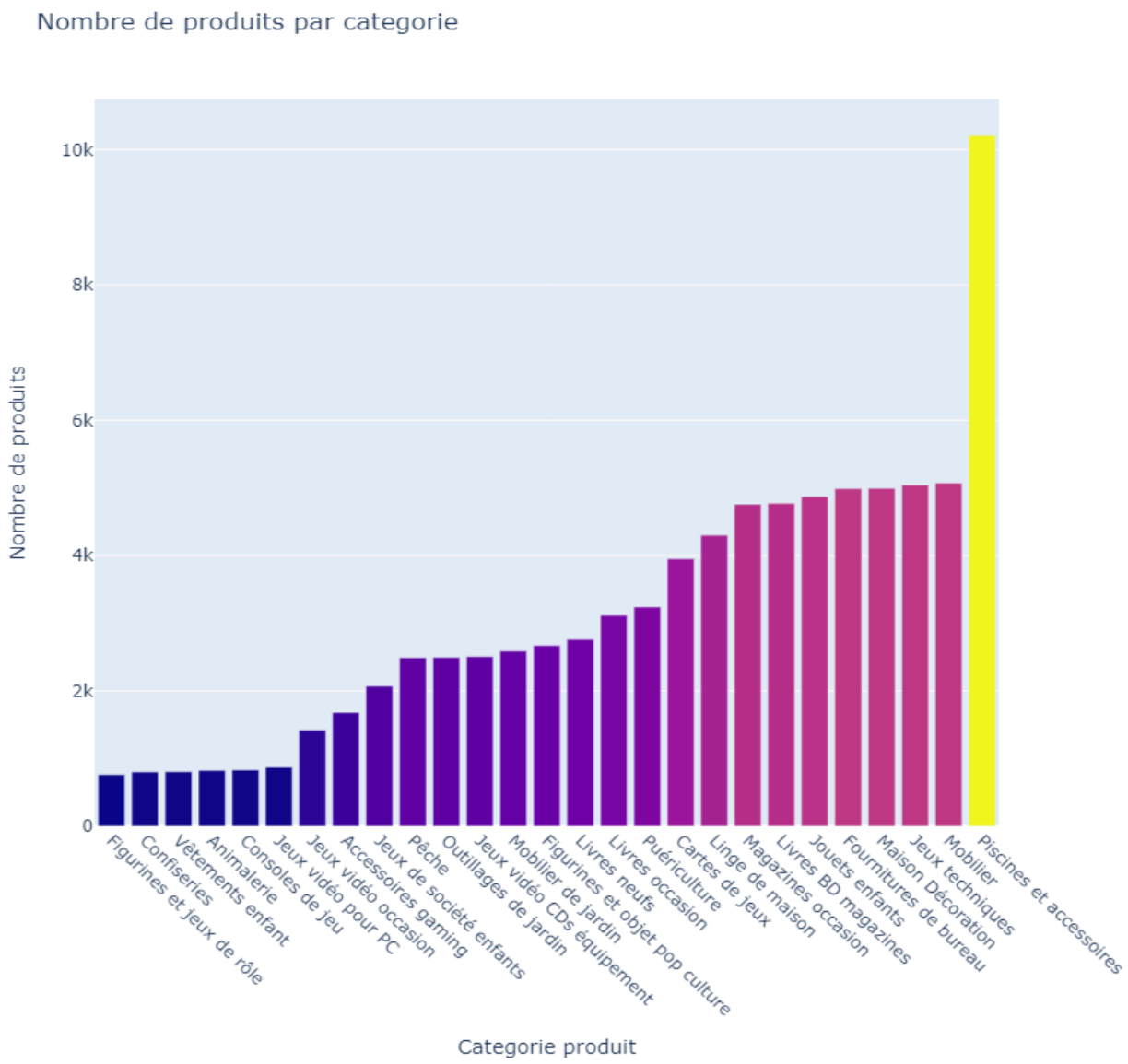


Figure 3

Distribution des produits par catégories (**Retour au texte**)

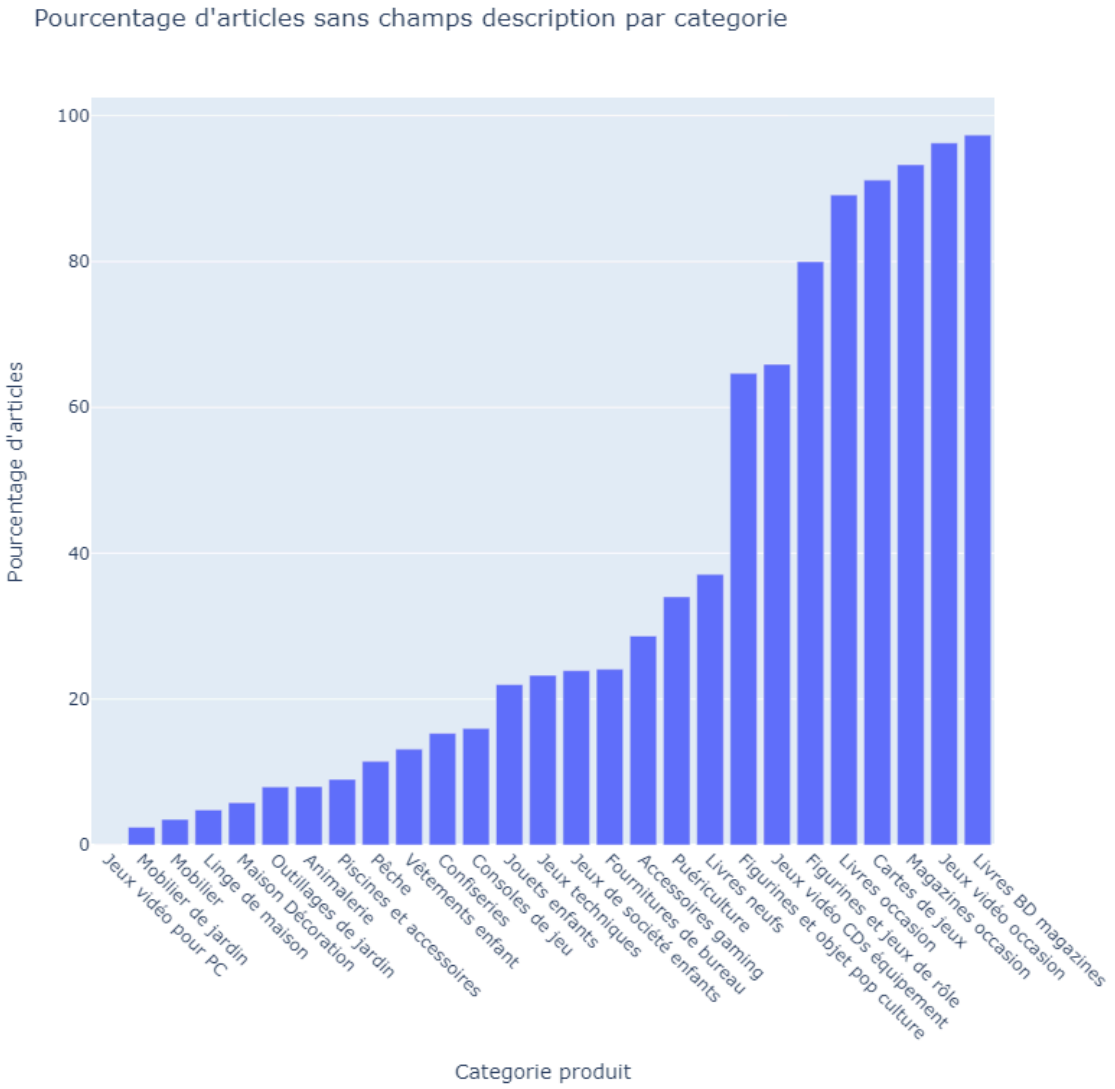


Figure 4

Pourcentage d'articles sans champ description par catégorie de produit (Retour au texte).

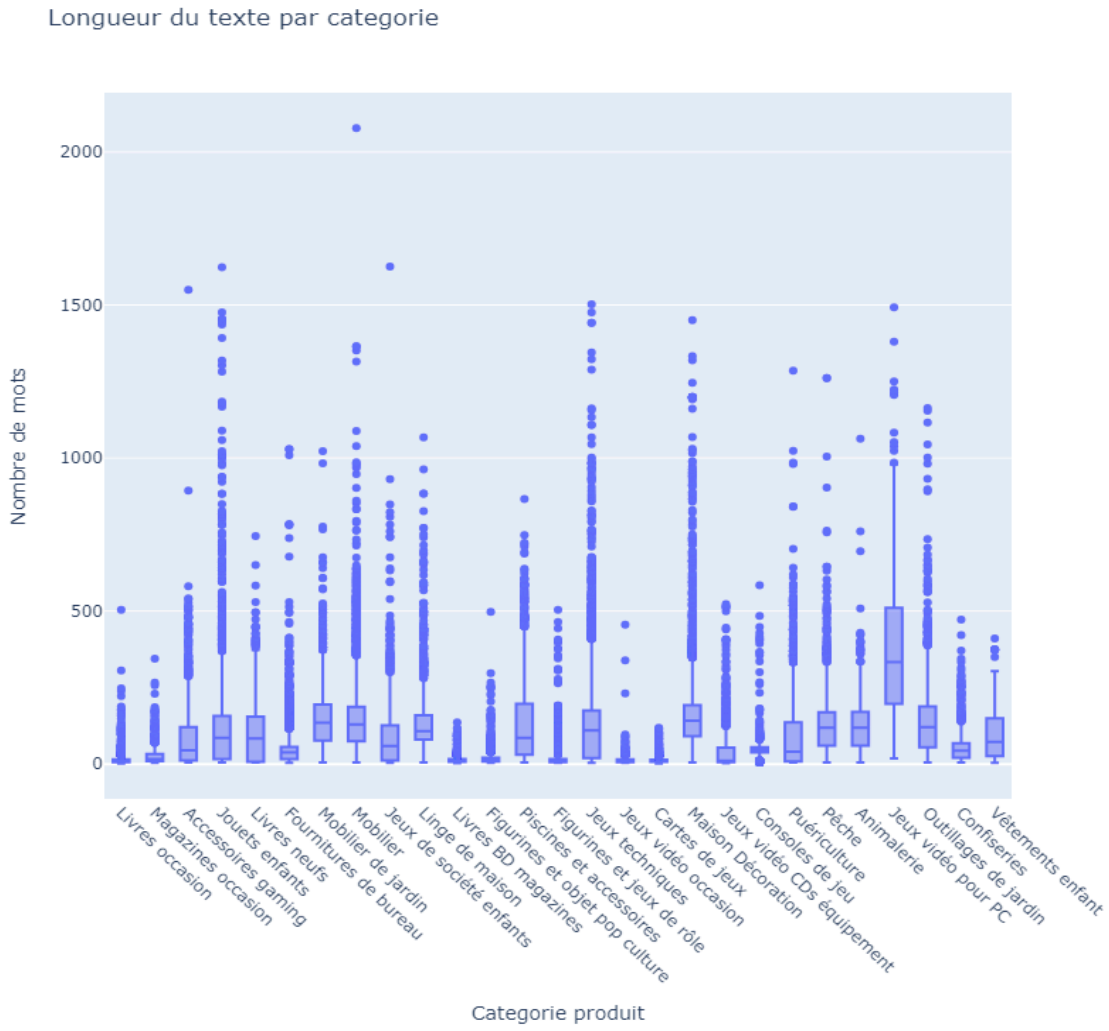


Figure 5

Longueur du texte totale par catégorie de produit (**Retour au texte**)

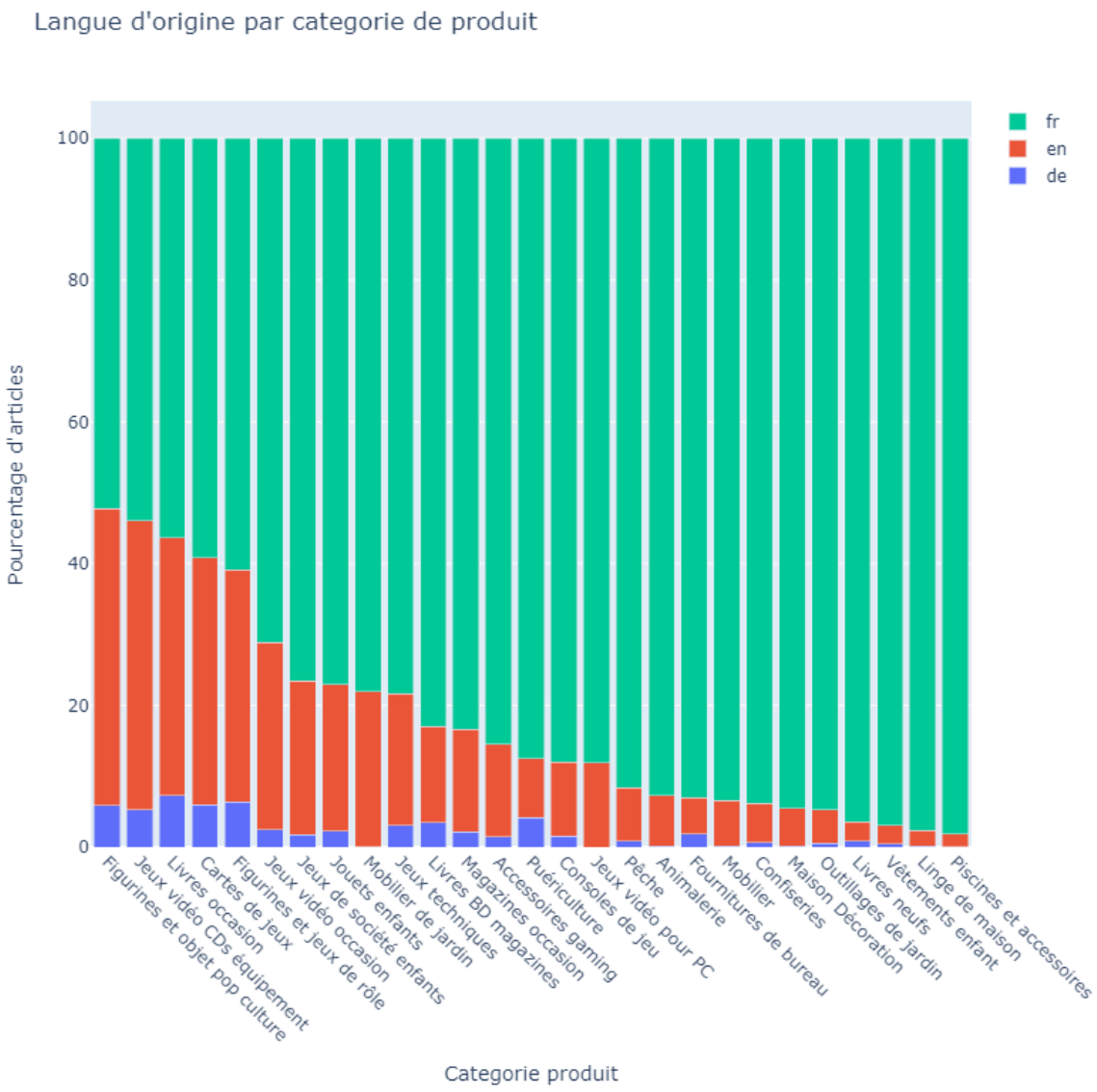


Figure 6

Pourcentage d'articles en français, anglais et allemand par catégorie (**Retour au texte**).

de haut en bas et de gauche à droite (livres d’occasion, livres BD magazines, magazines d’occasion et livres neufs

Figure 7

Word clouds pour quatre exemples de catégories proches (Retour au texte)

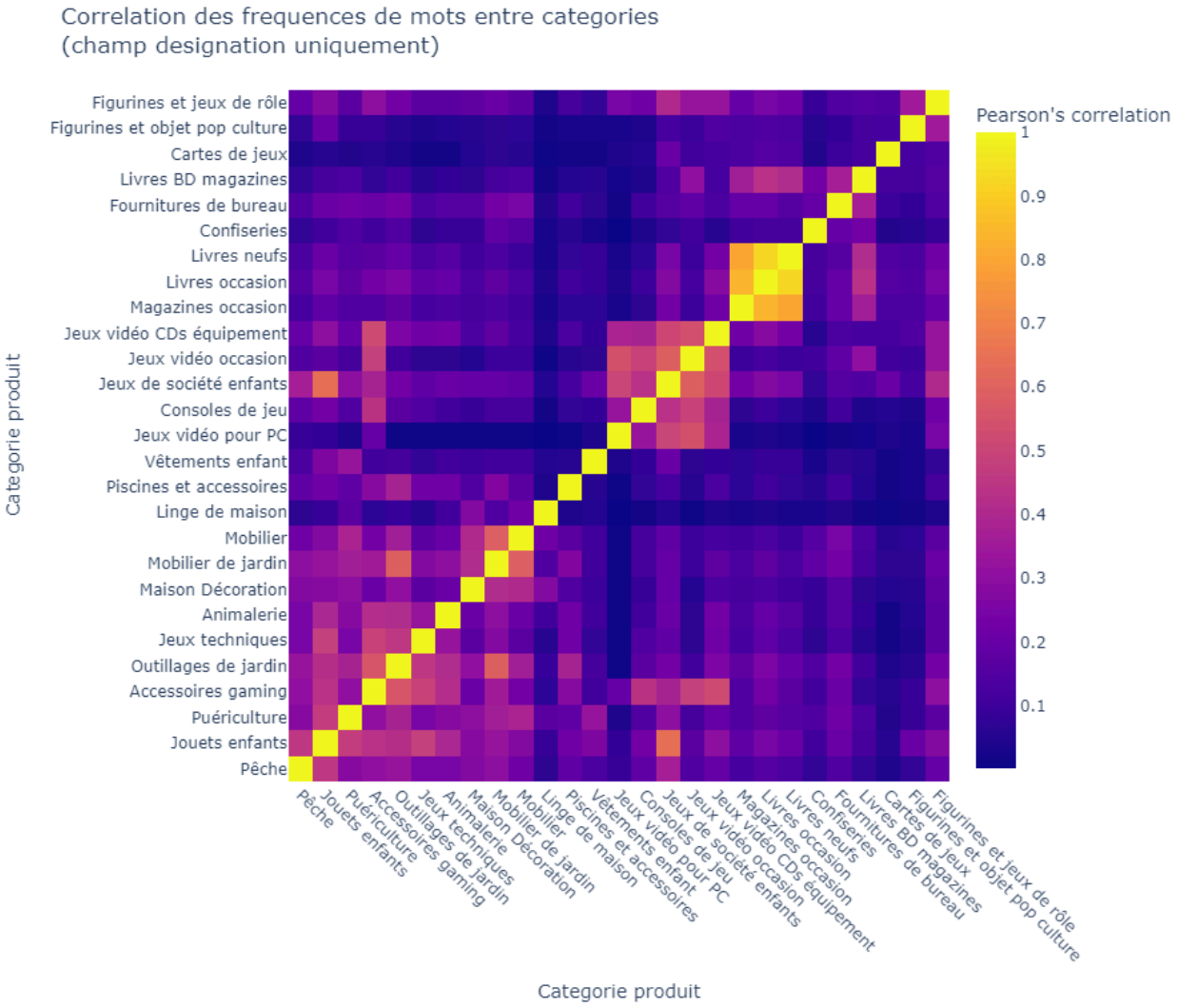


Figure 8

Matrice de corrélation entre vecteurs de fréquence de mot calculés pour chaque catégorie (**retour au texte**).

Figure 9

Benchmarks des F1 scores (weighted-F1) pour les modèles de machine learning classique, avec vectorisation de type Bag of word (TF-IDF) ou Word2Vec (skipgram et C-BOW) (**retour au texte**)

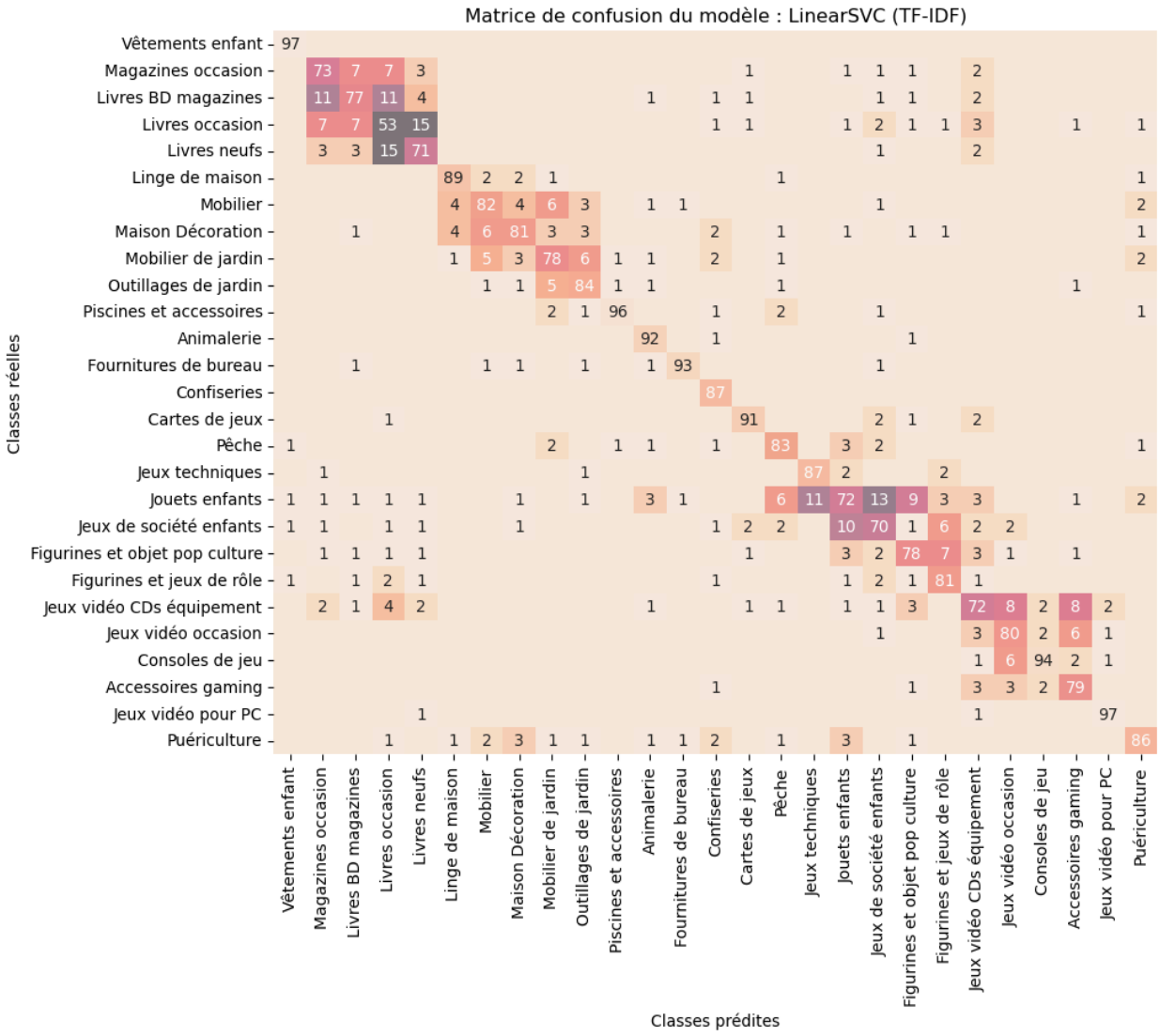


Figure 10

Matrice de confusion entre classes réelles (lignes) et classes prédites (colonne) pour le modèle LinearSVC appliqué au texte vectorisé par TF-IDF. Les valeurs sont exprimées en pourcentage d'articles par catégorie (normalisation par colonne) (**retour au texte**).

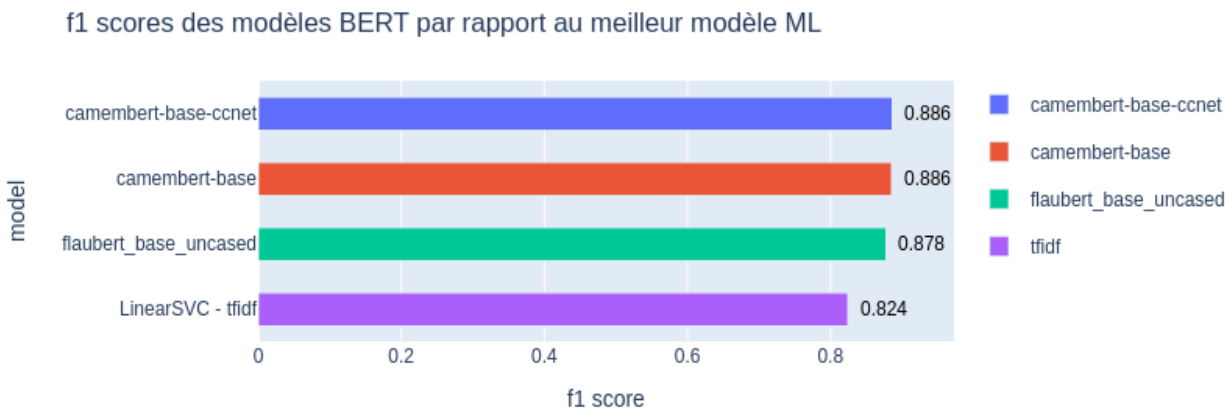


Figure 11

Benchmarks des F1 scores (weighted-F1) pour les modèles transformer de type BERT pour la classification de texte (**retour au texte**)

Figure 12

Matrice de confusion entre classes réelles (lignes) et classes prédites (colonne) pour l'architecture BERT pré-entraîné sur un corpus français (camemBERT-base-ccnet). Les valeurs sont exprimées en pourcentage d'articles par catégorie (normalisation par colonne) (**retour au texte**).

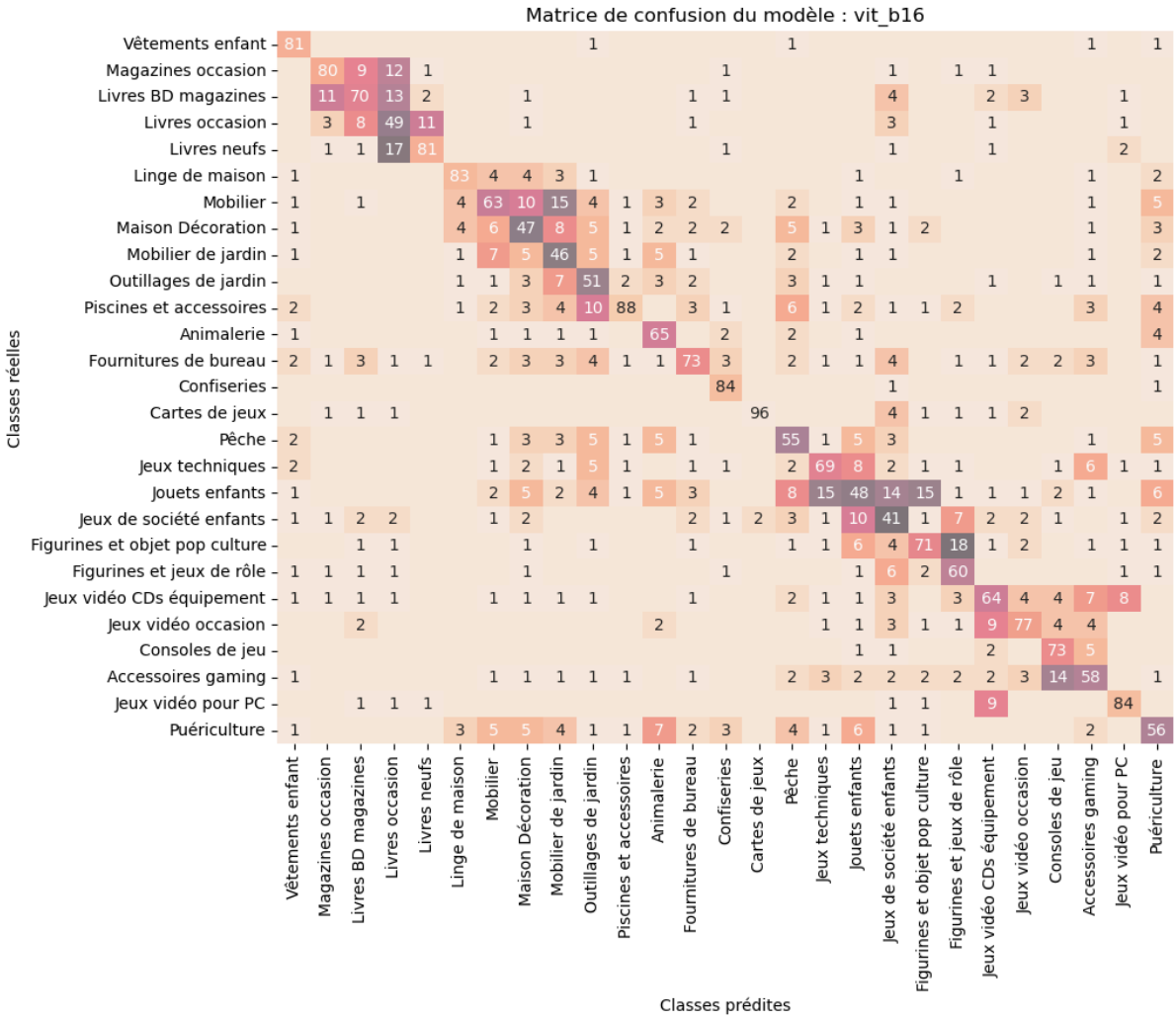


Figure 13

Matrice de confusion entre classes réelles (lignes) et classes prédites (colonne) pour le Vision Transformer (ViT), pré-entraîné sur imageNet. Les valeurs sont exprimées en pourcentage d'articles par catégorie (normalisation par colonne) (**retour au texte**).

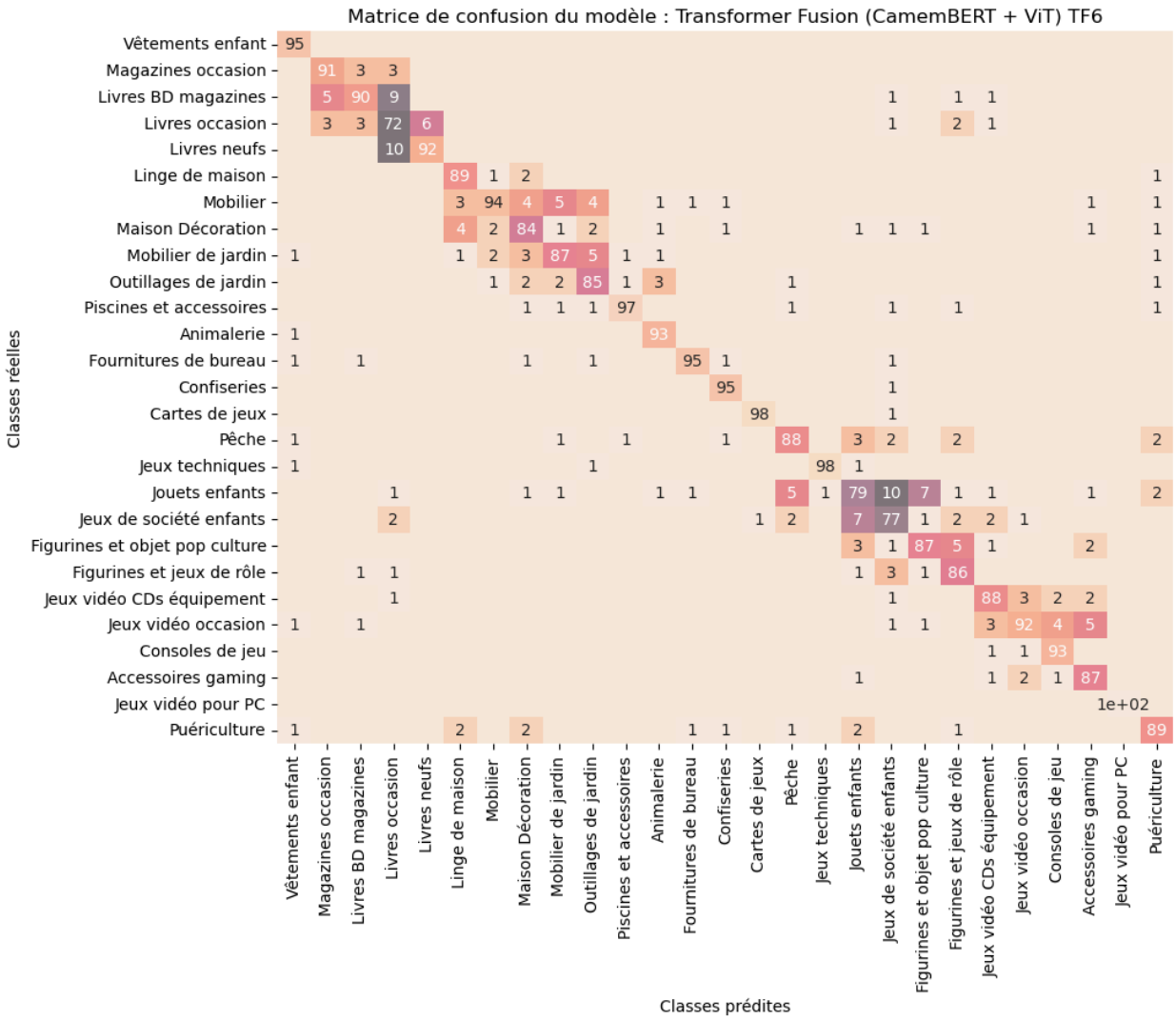


Figure 14

Matrice de confusion entre classes réelles (lignes) et classes prédites (colonne) pour le Fusion Transformer (TF6: CamemBERT + ViT). Les valeurs sont exprimées en pourcentage d'articles par catégorie (normalisation par colonne) (retour au texte).

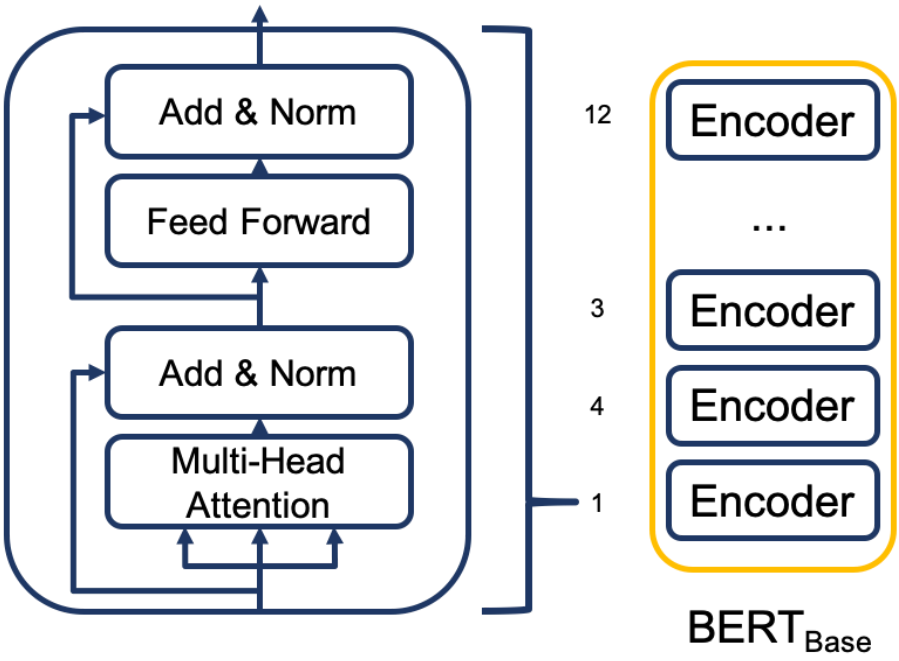


Figure 15

Architecture BERT (Bidirectional Encoder Representation from Transformers). L'architecture BERT de base (droite) est composée d'une succession de 12 blocs de transformers, chacun constitué d'une tête multi-attentionnelle a 12 couches, suivi d'une couche dense feed-forward avec connexions résiduelles (**retour au texte**).

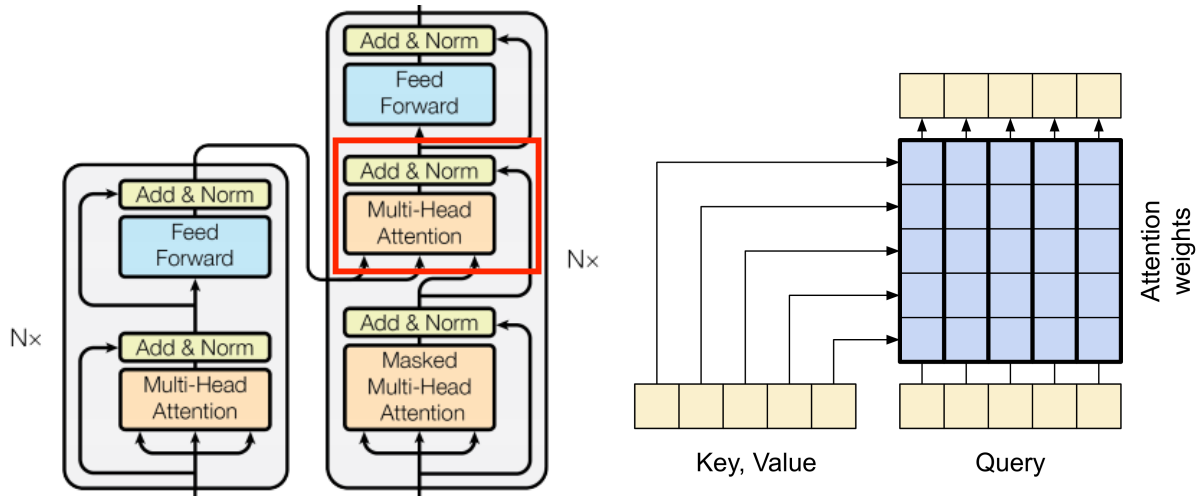


Figure 16

Représentation schématique de la fusion de BERT et ViT dans le modèle hybride TFx. Les sorties encodées de camemBERT et ViT sont combinées par l'intermédiaire d'un bloc de transformer avec une tête cross-attentionnelle (*gauche*) prenant comme *Key* et *Value* la sortie encodée de ViT et comme *Query* la sortie encodée de BERT (*droite*). Ce bloc de fusion est suivi de un ou plusieurs blocs de transformers classiques avec self-attention (**retour au texte**).

designation	description	true	pred
Bas De Noël 18 Pouces Avec Grand Flocon De Neige Bas Plaid Pour Candy Hu6014	Bas de Noël 18 __g Virt_NP_NNS_NNPS<__ pouces avec grand flocon de neige Bas Plaid pour Candy Description du produit Grand décor flocon de toile de jute. Il vaut mieux correspondre à Jute Plaid Snowflake Christmas Tree Skirt. FEEL RUSTIQUE. 100% flocon de neige de jute de qualité à carreaux classique fausse fourrure en peluche. Ajouter une ambiance chaleureuse de luxe à votre décor de vacances. Prêt à être farcie avec des cadeaux de goodies pour toute la famille. bas unique famille pour Noël. FORTE DURABLE. Triple couches conception avec du coton pp intérieur. Super épais solide. Dernière pour les années à venir. .. GRAND 18" SIZE Chaque bas de Noël a un ruban boucle pour accrocher affichage facile Style:. Rustique luxe Taille spacieux prêt à être bourrés de goodies liste: 1 x Bas Candy Noël Plaid.	Jouets enfants	Mobilier
Catalogue Phildar N°607 Modèles D'accessoires Femme À Tricoter		Livres BD magazines	Magazines occasion
Dragon Ball Z Ichiban Kuji Songoku Super Instinct		Figurines et objet pop culture	Cartes de jeux
Squishies Gâteau Kawaii Arc-En-Jumbo Rising Lent Squishies De Fromage Parfumé	Squishies Gâteau arc-Jumbo Kawaii lente hausse Squishies de fromage parfumé Fonction: Fonction: Stress Relief Pillow main Jouets décorat d'intéri Vent Emotions Finger Rehabilitation formation Simulations Jouets Matériel: PU matériaux de protection de l'environnement en mousse non toxiques. Squishies Caractéristique: Kawaii mignon doux Squishies Ralentir La hausse parfumée flexible. Stress Relief Oreiller main Jouets décorat d'intéri Vent Emotions Finger Rehabilitation formation Simulations Jouets Avis: Lorsque vous recevez les Squishies et après ventilation pendant 1-2 jours le parfum sera de retour à la normale. contenu du paquet: 1 x Gâteau arc	Pêche	Jouets enfants
Taie Canapé Voiture Taille Throw Coussin Décoration Pillow Case 24098	Taie Sofa taille voiture Throw Coussin Accueil Décoration Description: 100% tout neuf mode camouflage parfait pour place sur le canapé un café une bibliothèque un magasin de livres partie club. fleur Oiseaux feuilles imprimé modèle apporte un monde magnifique et coloré. Apporte Rechercher chaleureux et moderne à votre maison décorative Lavage en machine à froid séparément doucement cycle seulement sans javellisant Séchage en bas Repassage basse température Si nécessaire Parfait pour être une décoration pour la maison bureau voiture canapé etc. Taille : Forfait 45x45cm inclus: 1pc x Taie d'oreiller	Linge de maison	Maison Décoration

Table 3

Exemple de description de produits mal classifiés par le camemBERT (**retour au texte**).

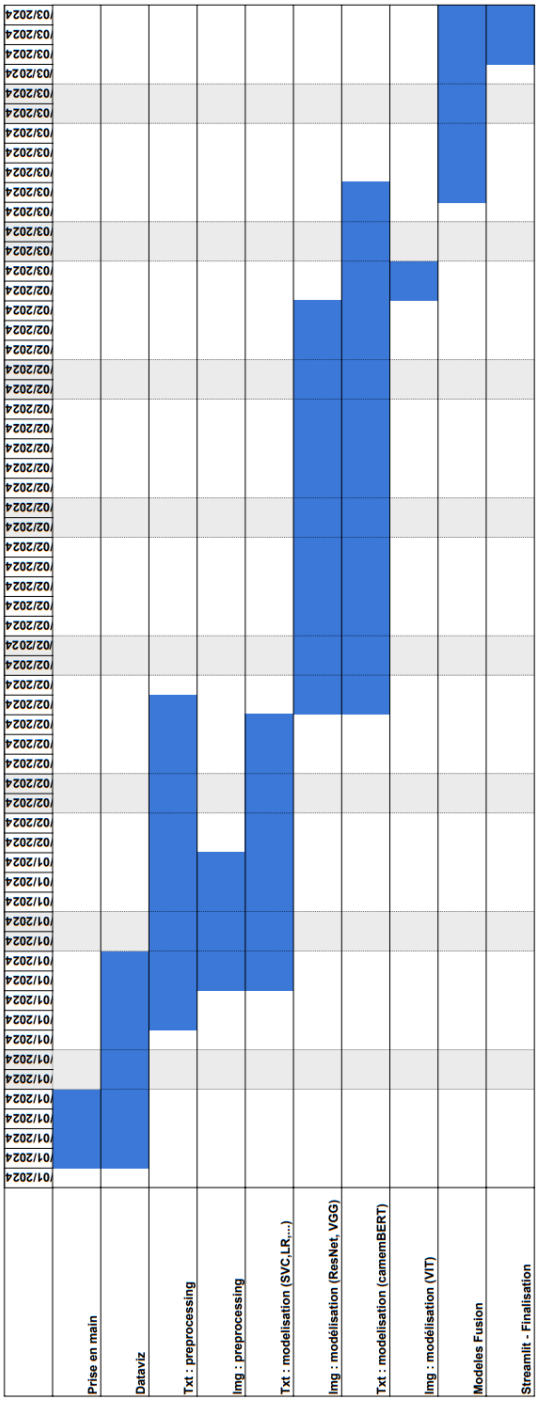


Table 7

Diagramme de Gantt.