CLASSIFIEZ AUTOMATIQUEMENT DES BIENS DE CONSOMMATION

alexandre delaguillaumie

INTRODUCTION

MISSION

- Réaliser une première étude de faisabilité d'un moteur de classification d'articles
- Basé sur une image et une description
- Pour l'automatisation de l'attribution de la catégorie de l'article

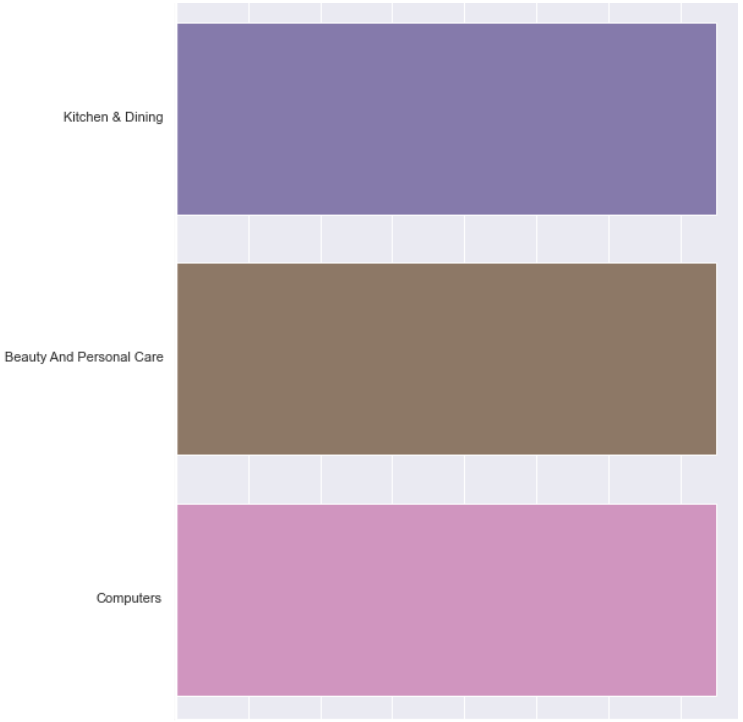
OBJECTIFS

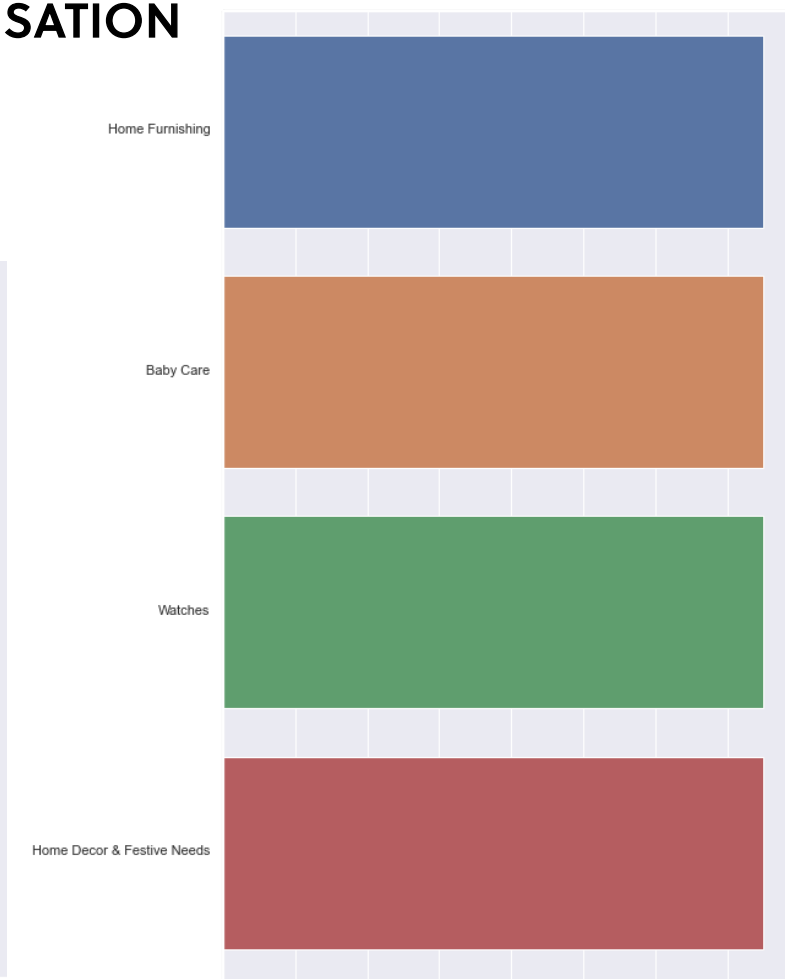
- Rendre plus fiable l'attribution de catégorie
- Faciliter l'expérience utilisateur et vendeur dans la mise en ligne et la recherche de produits

PRÉSENTATION DES DONNÉES

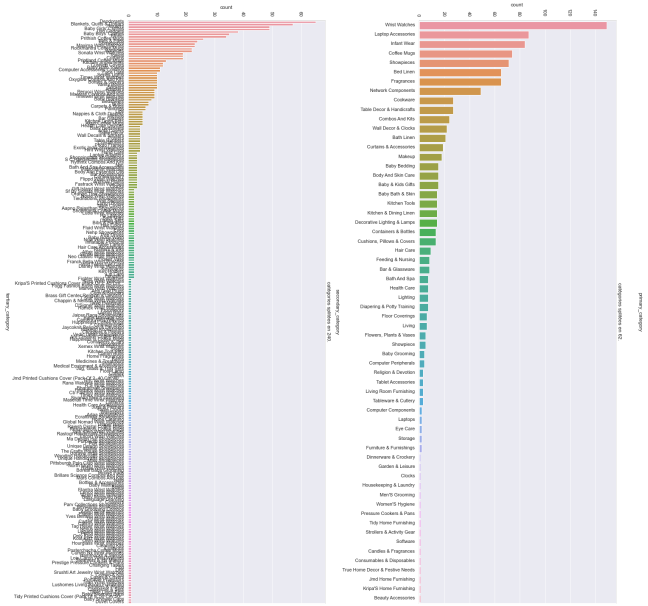
3 NIVEAUX DE CATÉGORISATION

Seul le premier nous permet une lisibilité et interprétabilité des données





Les niveaux 2 et 3 de l'arborescence du site sont trop complexes



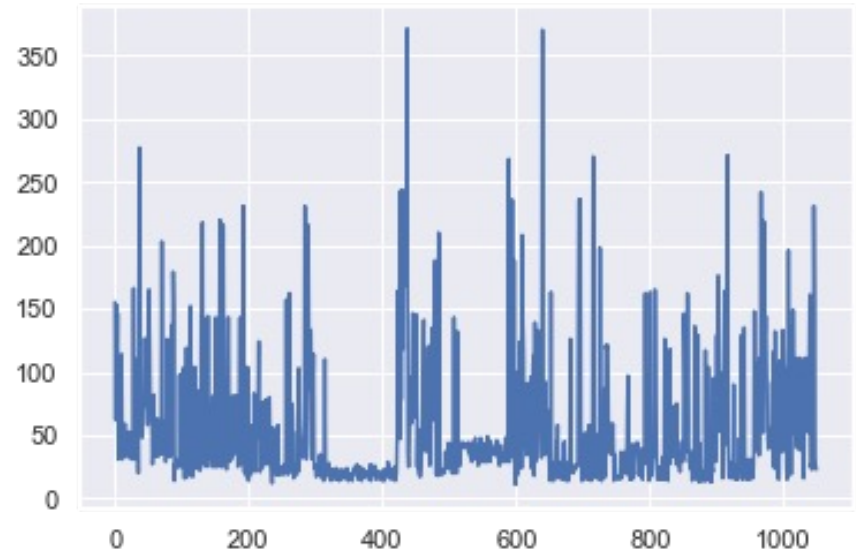
PRÉSENTATION DES DONNÉES

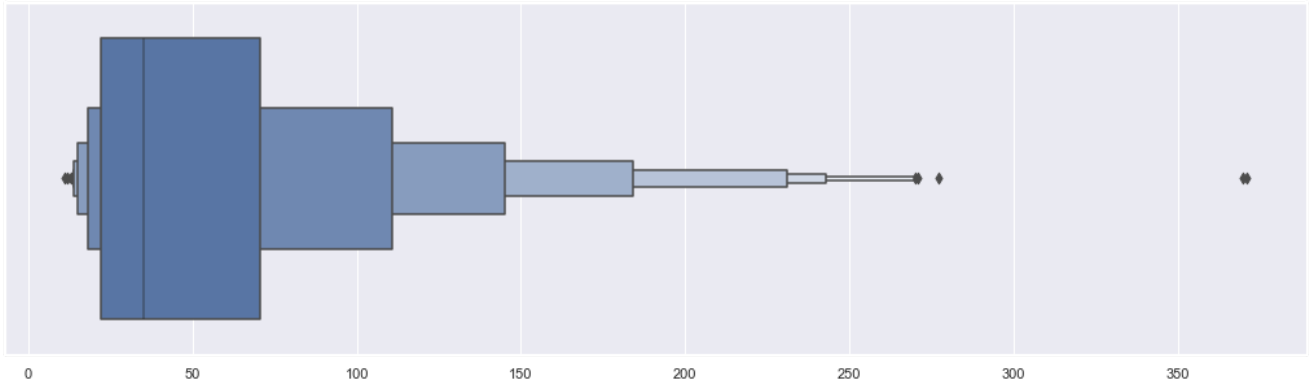
Produits
1050

produits par catégorie
150

Nombre max de mots
371

RÉPARTITION DE LA QUANTITÉ DE MOTS PAR PRODUITS





PARTIE TEXTE

PIPELINE DU TRAITEMENT DES DONNÉES TEXTUELLES

1. FEATURE EXTRACTION
2. t-SNE (Reduction de dimension)
3. CLUSTERING (Partitionnement non supervisé)
4. RAND MEASURE (Analyse de similarité)

BAG-OF-WORDS

PRINCIPE

Algorithme de représentation vectorielle du nombre d'occurrence de mots d'un dictionnaire

ALGORITHMES UTILISÉS :

CountVectorizer

&

TF-IDF

INCONVÉNIANT

toute information sur l'ordre ou la structure des mots dans le document est rejetée.

PREPROCESSING

```
def transform_bow_lem_fct(desc_text) :
    word_tokens = tokenizer_fct(desc_text)
    sw = stop_word_filter_fct(word_tokens)
    lw = lower_start_fct(sw)
    lem_w = lemma_fct(lw)
    transf_desc_text = ' '.join(lem_w)
    return transf_desc_text
```

HYPERPARAMS

stop_words='english'
max_df=0.95
min_df=1

BAG-OF-WORDS

CountVectorizer

PRINCIPE

Simple comptage de mots dans le corpus

EXEMPLES (1050 lignes, 5843 colonnes)

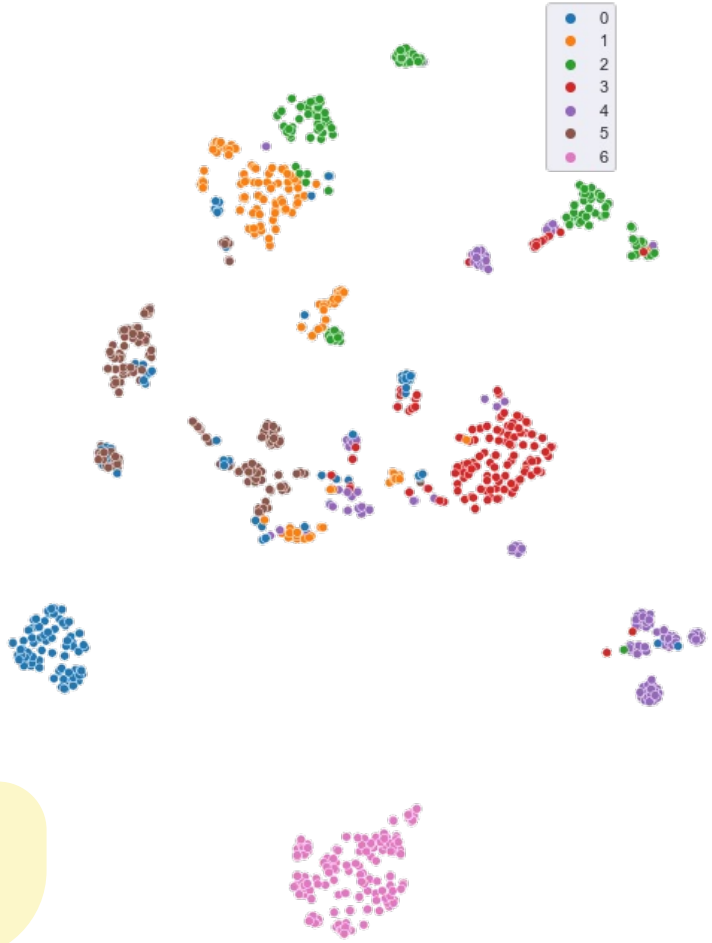
	abroad	absolut	absorb	abstract	abstrct	ac	accent	access	accessori	accid
0	0	0	0	4	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0

BAG-OF-WORDS

CountVectorizer

t-SNE des catégories prédites

t-SNE des catégories réelles



ARI

0.3784

BAG-OF-WORDS

TF-IDF

PRINCIPE

Calcul de la fréquence d'apparition des mots, ainsi que pénalisation des fréquences élevée dans le corpus

EXEMPLES (1050 lignes, 5843 colonnes)

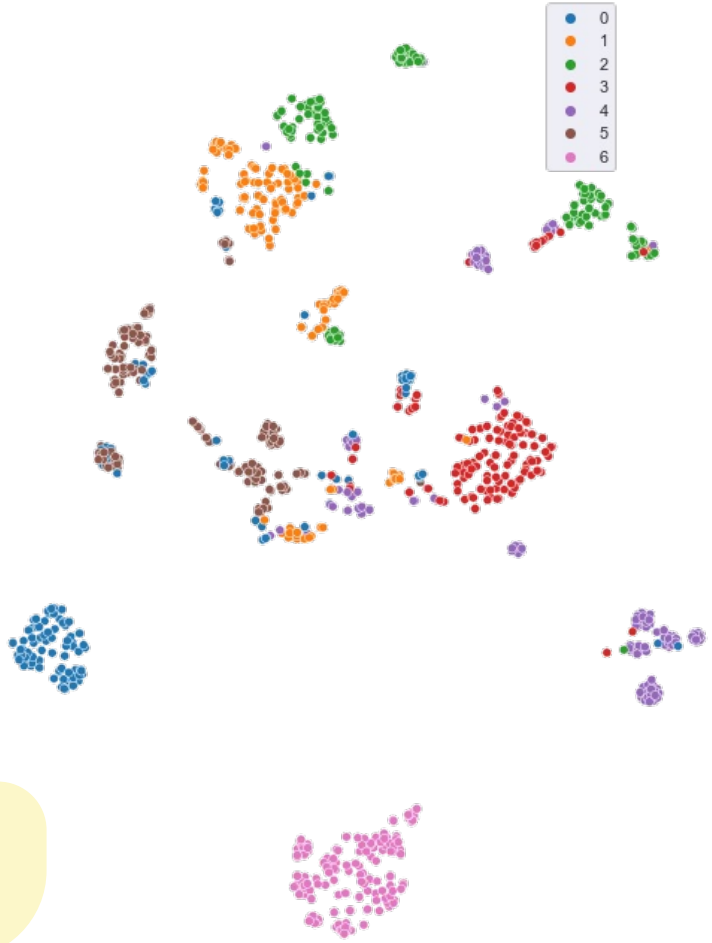
	abroad	absolut	absorb	abstract	abstrct	ac	accent	access	accessori	accid
0	0.0	0.0	0.000000	0.171482	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.052647	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0

BAG-OF-WORDS

TF-IDF

t-SNE des catégories prédites

t-SNE des catégories réelles



ARI

0.5567

WORD EMBEDDING

PRINCIPE

Les mots qui partagent des contextes similaires (par réduction de dimension) sont représentés par des vecteurs numériques proches.

ALGORITHME UTILISÉ :
Word2Vec, USE, BERT

PREPROCESSING

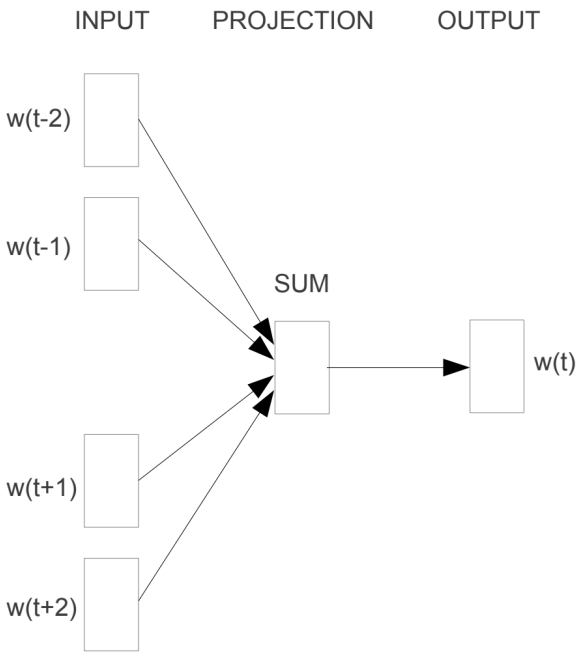
```
def transform_dl_fct(desc_text) :
    word_tokens = tokenizer_fct(desc_text)
#     sw = stop_word_filter_fct(word_tokens)
    lw = lower_start_fct(word_tokens)
    # lem_w = lemma_fct(lw)
    transf_desc_text = ' '.join(lw)
    return transf_desc_text
```

&

```
def transform_bow_lem_fct(desc_text) :
    word_tokens = tokenizer_fct(desc_text)
    sw = stop_word_filter_fct(word_tokens)
    lw = lower_start_fct(sw)
    lem_w = lemma_fct(lw)
    transf_desc_text = ' '.join(lem_w)
    return transf_desc_text
```

WORD EMBEDDING

Word2Vec



CBOW

PRINCIPE

- Basé sur les Google news
- Embedding statique
- Réseau de neurone avec une couche caché
- Vecteur retenu représente les poids de la couche cachée

HYPERPARAMS

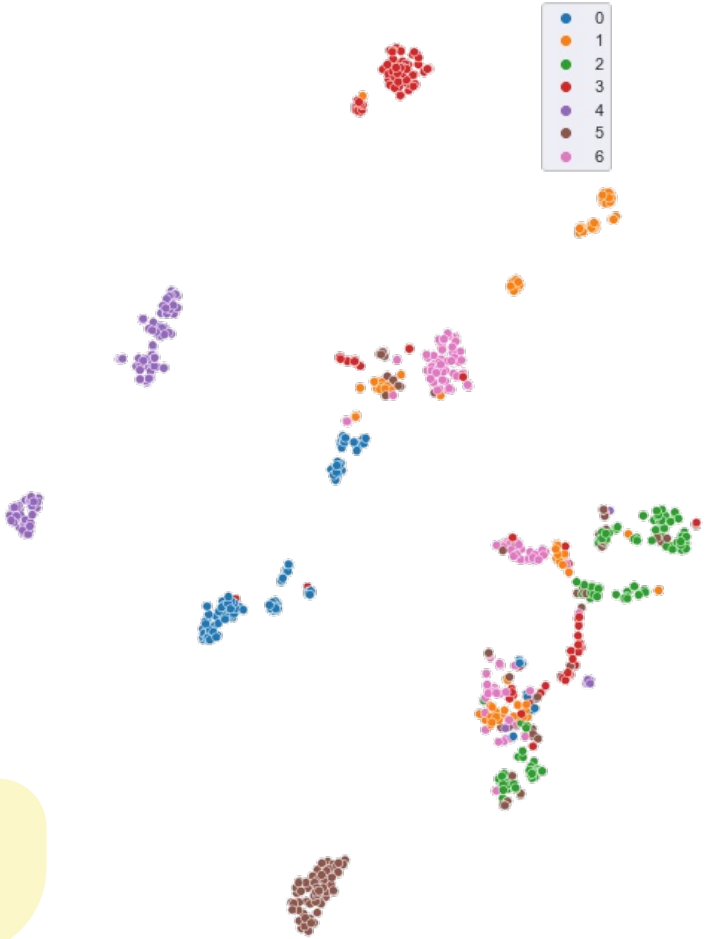
w2v_window=**5**
w2v_min_count=**1**
w2v_epochs=**485**
w2v_size=**550**

WORD EMBEDDING

Word2Vec

t-SNE des catégories prédites

t-SNE des catégories réelles

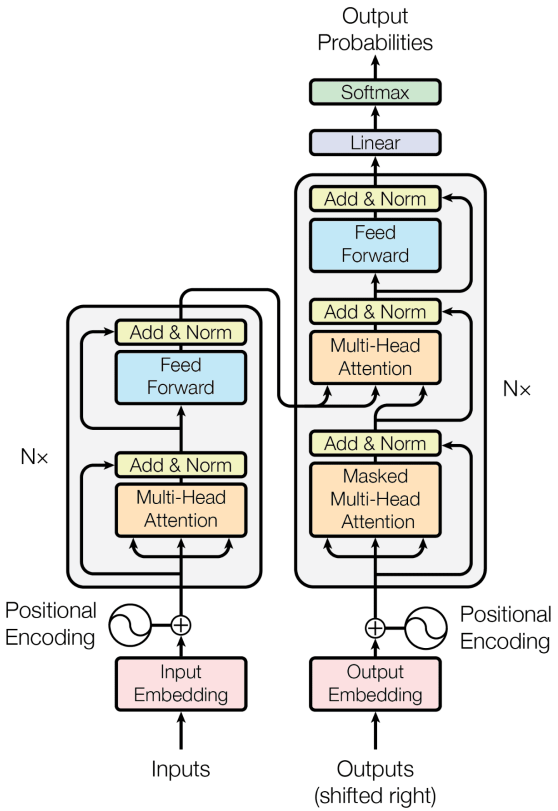


ARI

0.3613

WORD EMBEDDING

USE



PRINCIPE

- Universal Sentence Encoder
- Basé sur Wikipédia
- Réseau de neurone basé sur l'architecture Transformers
- Sentence embedding

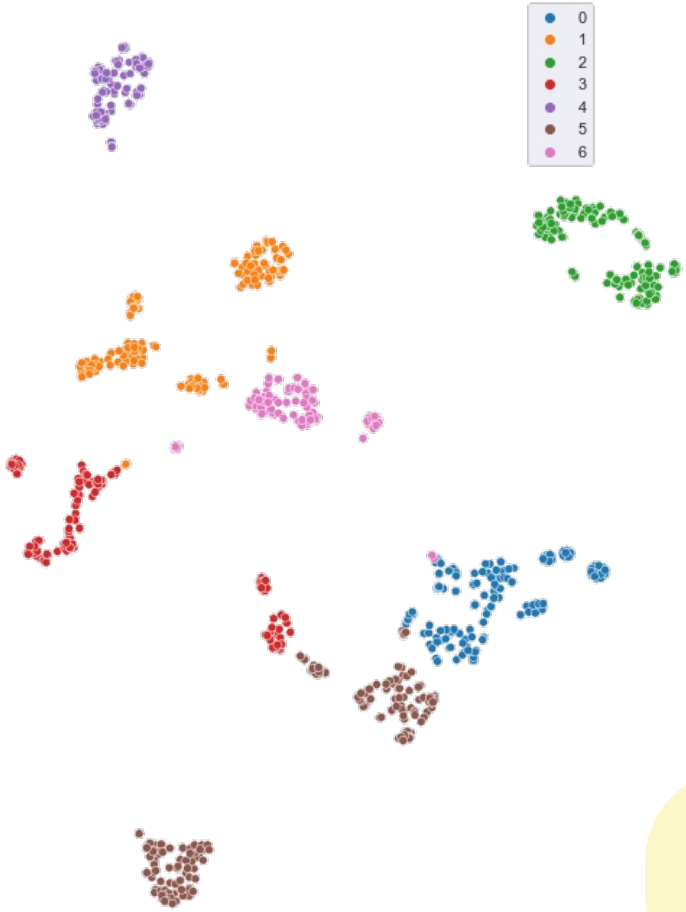
HYPERPARAM

batch_size=**10**

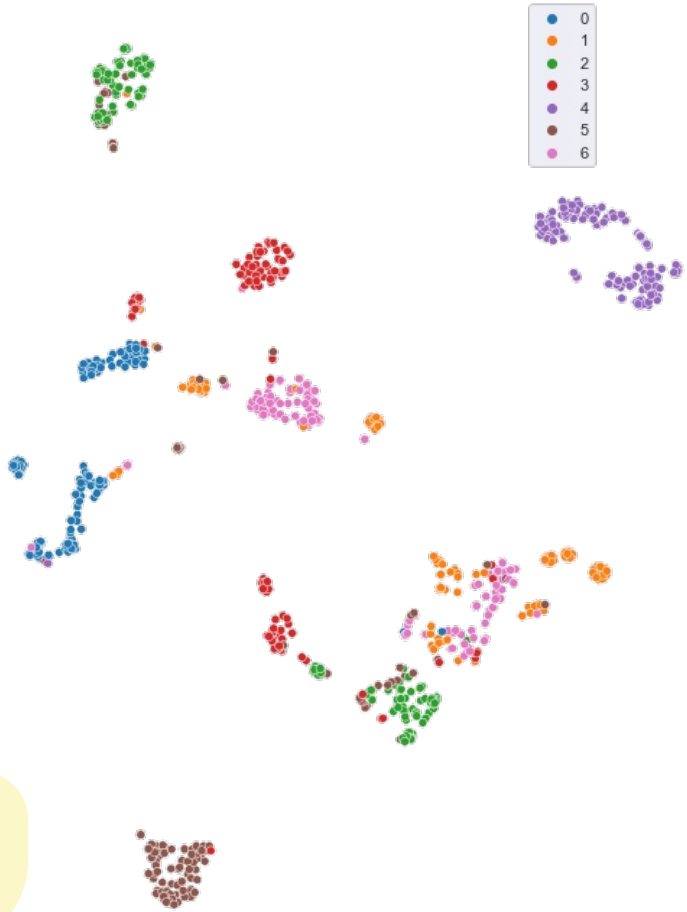
WORD EMBEDDING

USE

t-SNE des catégories prédites



t-SNE des catégories réelles

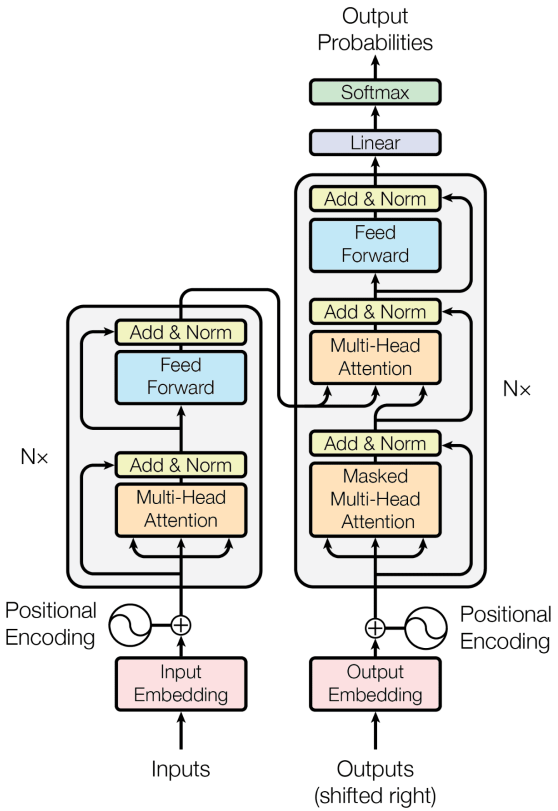


ARI

0.454

WORD EMBEDDING

BERT



PRINCIPE

- Bidirectional Encoder Representations from Transformers
- Basé sur Wikipédia
- Réseau de neurone basé sur l'architecture Transformers
- Embedding dynamique (se base sur les phrases)
- Modèle bidirectionnel (mots avant et après la cible)

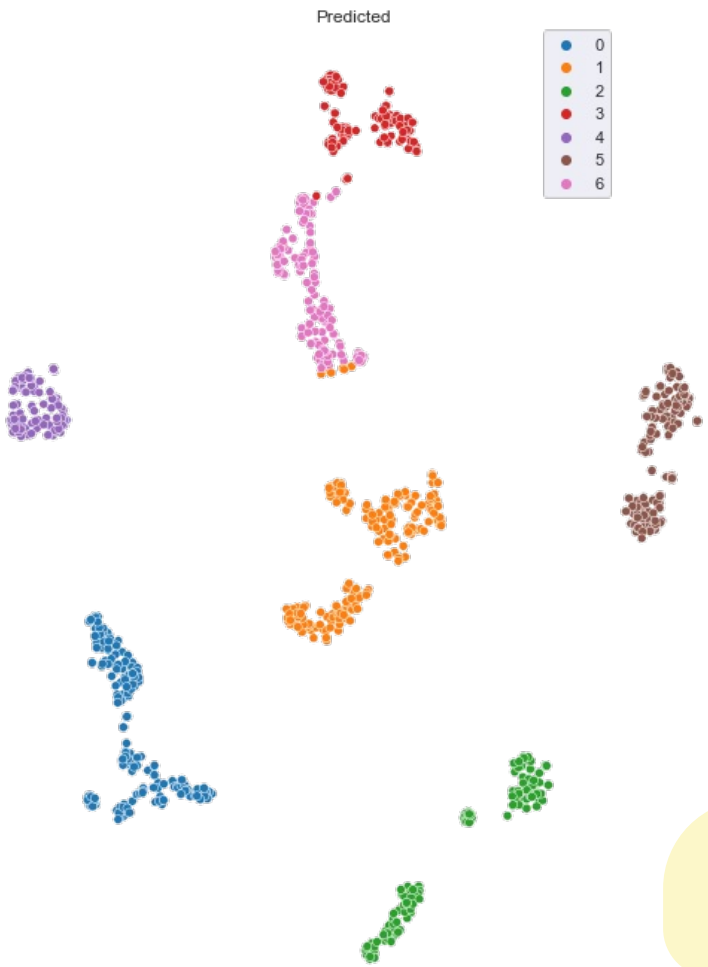
HYPERPARAM

batch_size=**10**

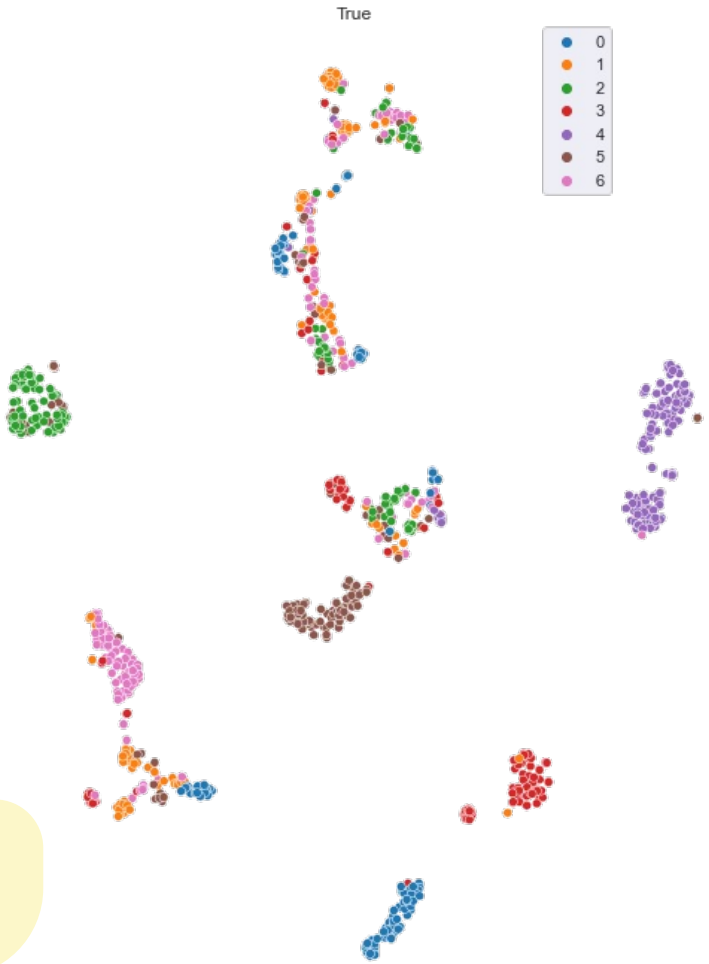
WORD EMBEDDING

BERT

t-SNE des catégories prédites



t-SNE des catégories réelles



ARI

0.2887

PARTIE IMAGE

PIPELINE DU TRAITEMENT DES IMAGES

1. FEATURE EXTRACTION
2. CLUSTERING DESCRIPTEURS
3. PCA & t-SNE (Reduction de dimension)
4. RAND MEASURE (Analyse de similarité)

PRÉALABLE AU TRAITEMENT DES IMAGES

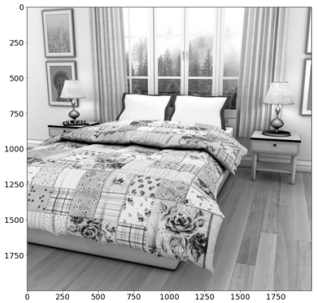
PREPROCESSING

1

IMAGE ORIGINALE

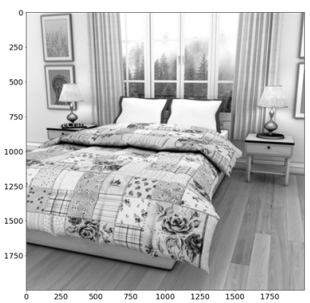
2

NUANCES DE GRIS



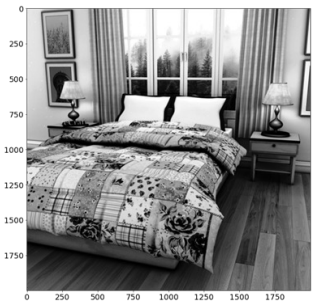
3

SUPPRESSION DU BRUIT



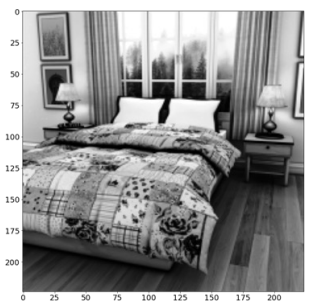
4

EGALISATION DE L'HISTOGRAMME



5

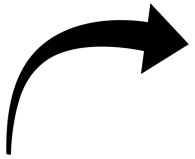
REDIMENSION 255 X 255



SIFT (Scale-Invariant feature transform)

PRINCIPE

- Détecte des points d'intérêt à différentes échelle de zoom de l'image et par comparaison de filtrage gaussien successif
- Les descripteurs ou points d'intérêt sont identifiés grâce aux fortes variations d'intensité ou de couleur des pixels
- Ils sont invariants par rotation, changement d'échelle et exposition



PROBLÈME

Le nombre de descripteurs varie pour chaque image, ce qui rend impossible l'utilisation des descripteurs comme feature

SOLUTION

Faire un Bag Of Visual Words via un MiniBatchKMeans pour déterminer le nombre de descripteurs par cluster pour chaque image (principe de récurrence)

SIFT (Scale-Invariant feature transform)

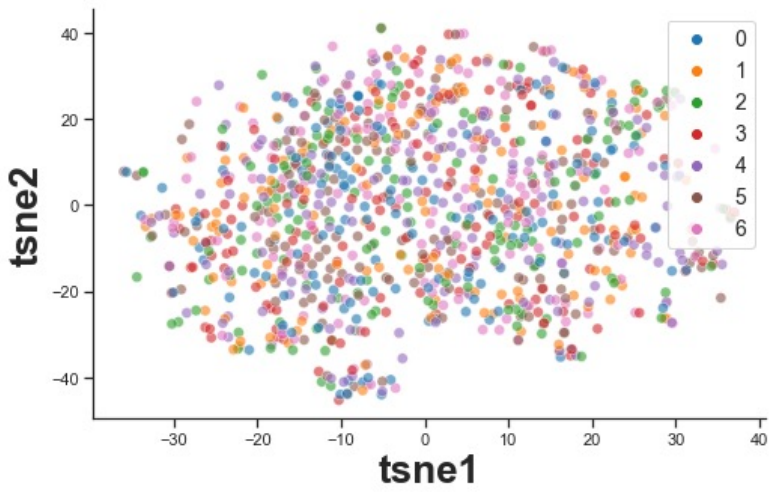
NOMBRE DE COLONNES

Descripteur
516 533

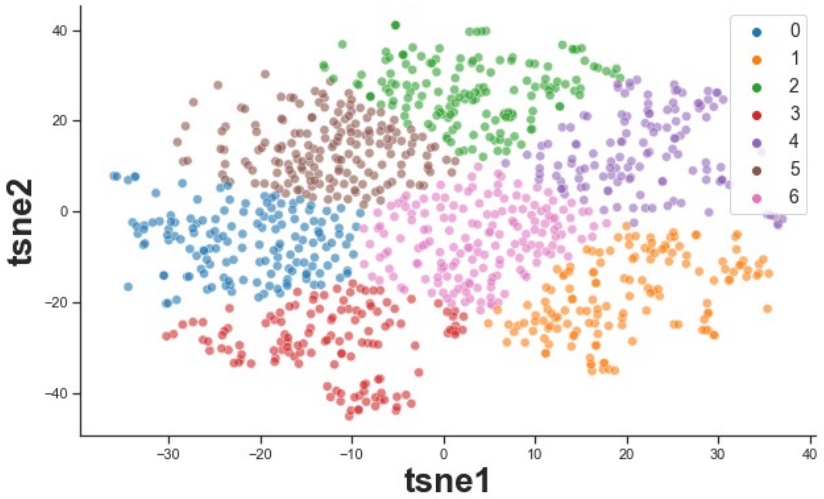
KMeans
719

PCA (99% variance)
501

TSNE selon les vraies classes



TSNE selon les clusters



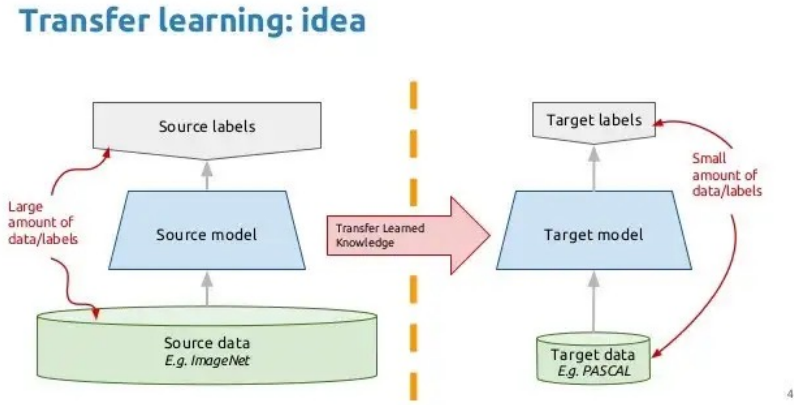
ARI
$\simeq 0$

TRANSFER LEARNING

CNN

PRINCIPE

- Convolutional Neural Networks (Deep Learning)
- Utiliser un modèle de Deep Learning pre-entrainé sur des millions d'images
- Enlever la dernière couche du modèle et faire un predict pour créer des features
- Transférer ses larges connaissance au dataset actuel



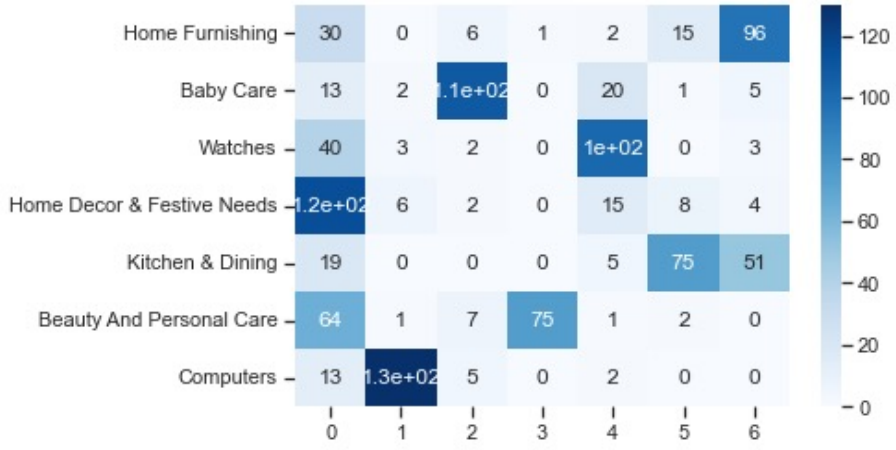
TRANSFER LEARNING

CNN

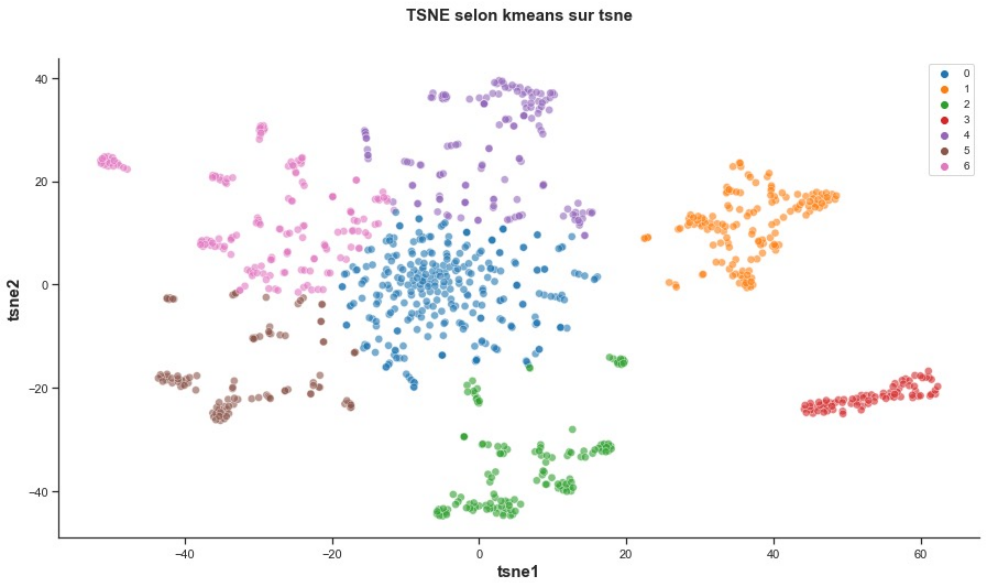
NOMBRE DE COLONNES

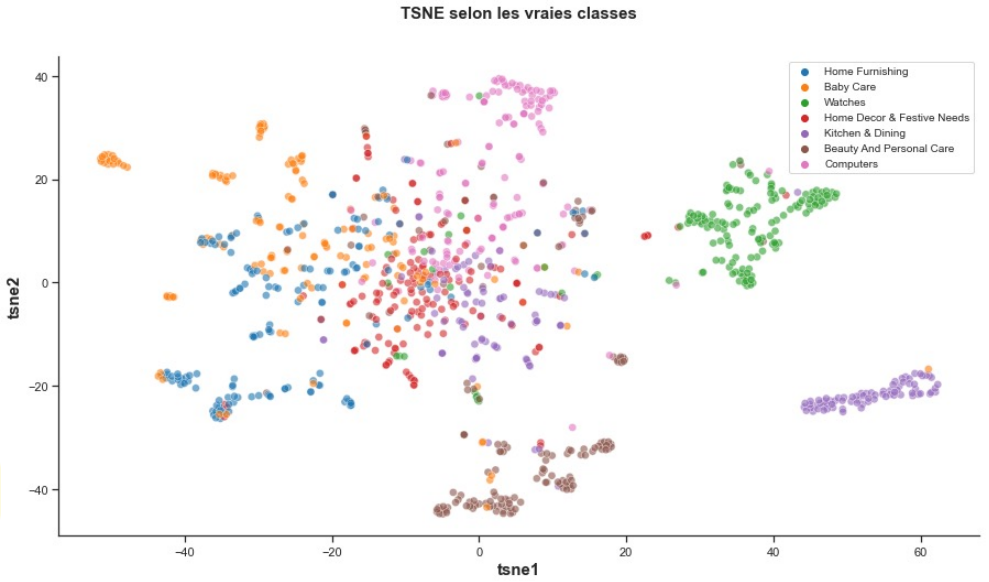
KMeans
1000

PCA (99% variance)
202



ARI
0,3908





CONCLUSION

- La segmentation est automatisable malgré des scores perfectibles (ARI à 0,39 et 0,55)
- Il serait préférable de clusteriser via les données images et textuelles conjointement
- Il est impératif d'utiliser des catégories larges de produit
- Cette mise en place sera certainement plus efficace que le système actuel d'attribution manuel

MERCI POUR VOTRE ÉCOUTE

alexandre delaguillaumie