Projet pour classifier automatiquement des biens de consommation

Cyril REGAN

17 juillet 2020

Table of Contents

Problématique

Besoin :
classifier automatiquement des biens de consommation

Données

- 1050 marchandises [description, image]
- 7 catégories Home Furnishing, Baby Care, Beauty and Personal Care, Computer, Kitchen Dining, Watches, Home Decor Festive Needs

Analyse des :

- descriptions
- images
- des deux ensembles

Table of Contents

Préparation des données

Exemple de préparation des données textuelles :

'Key Features of Elegance Polyester Multicolor Abstract Eyelet Door Curtain Floral Curtain.'

- Racinisation :

=> 'key featur of eleg polyest multicolor abstract eyelet door curtain floral curtain '

- Suppression stop-words (basique + 40 plus fréquents du corpus) + tokenisation :

=> ['abstract', 'curtain', 'door', 'eleg', 'eyelet', 'featur', 'floral', 'key', 'multicolor', 'polyest']

Sac de mots (Bag of word : BOW)

{"Je suis à la maison", "La maison est dans la prairie", "Je suis à la plage"}

	je	suis	à	la	maison	est	dans	prairie	plage
phrase 1	1	1	1	1	1	0	0	0	0
phrase 2	0	0	0	2	1	1	1	1	0
phrase 3	1	1	1	1	0	0	0	0	1

Term-Frequency - Inverse Document Frequency (TF-IDF)

TF = nombre de fois où le mot est dans le document / nombres de mots dans le document

IDF = nombre de document / nombre de documents où apparaît le mot

	...	la	...
phrase 1	...	$\frac{1}{5} \times \frac{3}{3} = 0.2$	...
phrase 2	...	$\frac{2}{6} \times \frac{3}{3} = 0.3$	...
phrase 3	...	$\frac{1}{5} \times \frac{3}{3} = 0.2$	...

Table of Contents

Plongements de mot (word embeddings) : **Word2vec**

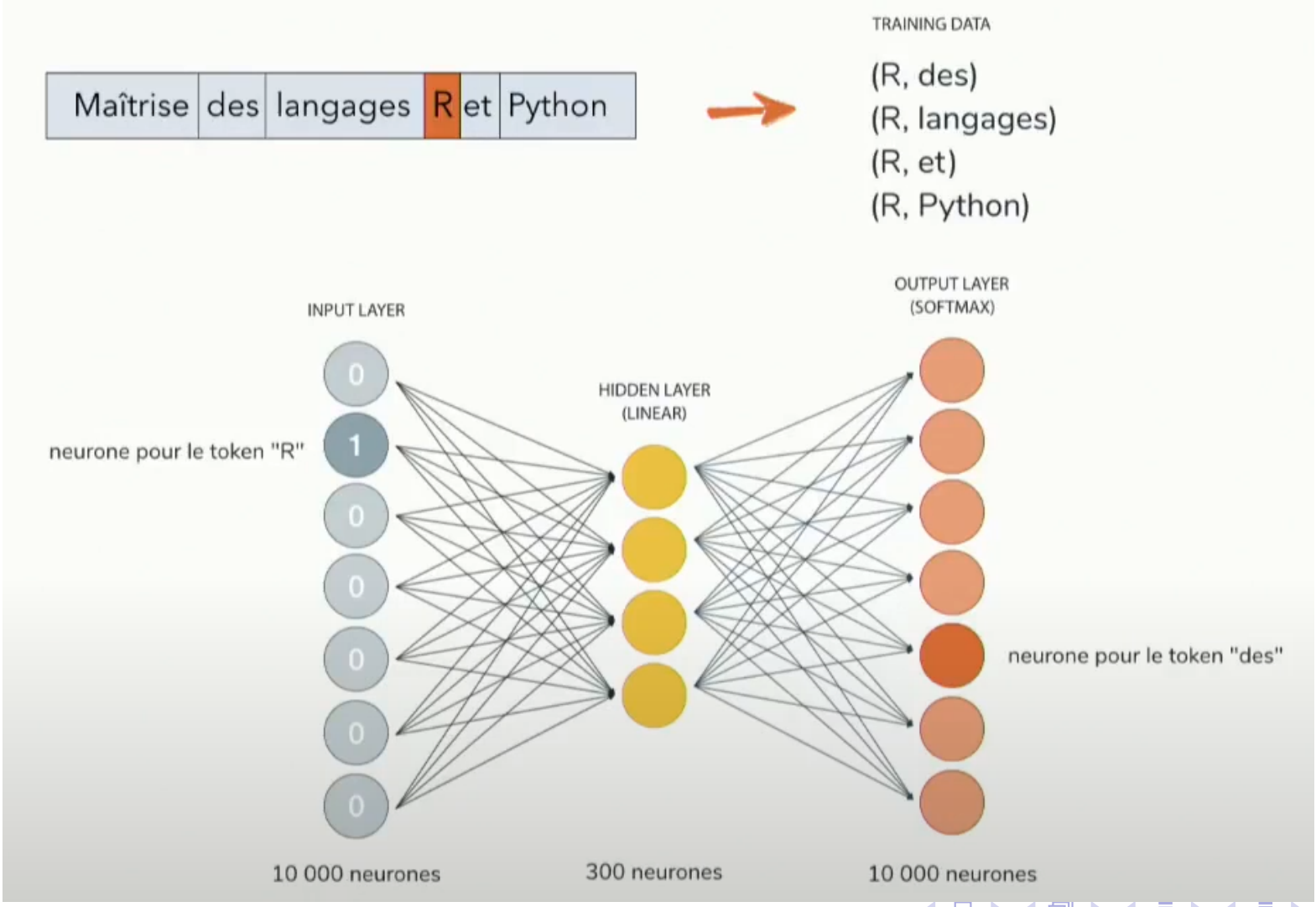


Table of Contents

Plongements de mot (word embeddings) : **Bidirectional Encoder Representations from Transformers (Bert)**

BERT est :

- Entièrement bidirectionnel
- Fonctionne avec : transformers (mécanisme de l'attention)
- Exécute 2 taches automatiquement :
  - Masqued Langage Modeling (MLM) : BERT masque 15 % des mots input aléatoirement
  - Prédiction prochaine phrase
- Pré-entrainné sur corpus de livre (800M de mots anglais) et Wikipédia anglais.
  CamemBERT pré-entraîné sur 138Go de texte français.

Classification

Données stratifiées à 80% d'entraînement et 20% test.

Deux temps :

- Évaluation modèles sur TF-IDF.
- Évaluation encodages (TF-IDF, BERT) sur best modèle.

Table of Contents

Évaluation modèles de classification :

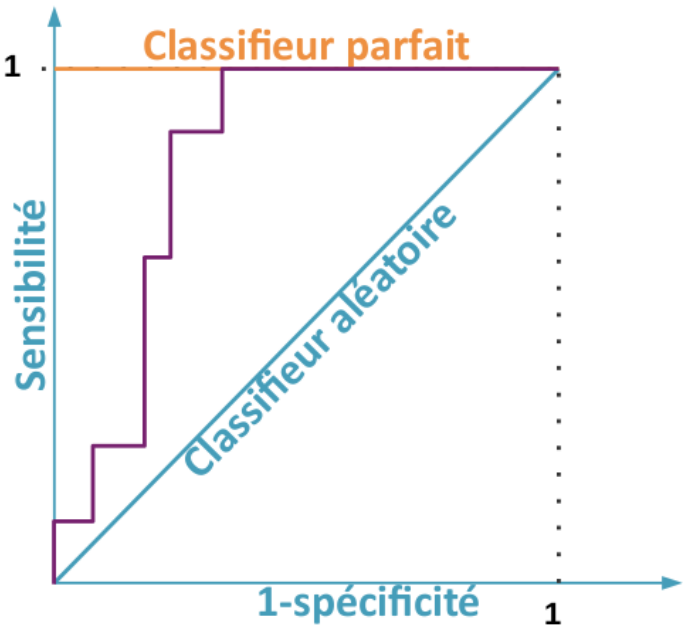
- Justesse (accuracy).
- F-mesure (f1).
- Précision (precision).
- Rappel ou sensibilité (recall) .
- L'AUROC (aire sous la courbe ROC).

Matrice de confusion :

		Classe réelle	
		-	+
Classe prédite	-	True Negatives *(vrais négatifs)*	False Negatives *(faux négatifs)*
	+	False Positives *(faux positifs)*	True Positives *(vrais positifs)*

- $Rappel = \frac{TP}{TP+FN}$: taux de vrais positifs. Proportion de positifs que l'on a correctement identifiés.
- $Precision = \frac{TP}{TP+FP}$: proportion de prédictions correctes parmi les points que l'on a prédits positifs.
- $F - mesure = 2 \times \frac{Precision \times Rappel}{Precision + Rappel}$: moyenne harmonique entre rappel et précision.
- $Specificite = \frac{TN}{FP+TN}$: taux de vrais négatifs.

- *Justesse*(*accuracy*) : proportion de prédiction exacte.
- Receiver-Operator Characteristic (*ROC*, pour sorties en probabilité que les points soient positif) : Courbe Sensibilité (Rappel) / Spécificité en faisant varier le seuil.



L'AUROC est "l'aire sous la courbe ROC"

Table of Contents

Présentation modèles

Modèles :

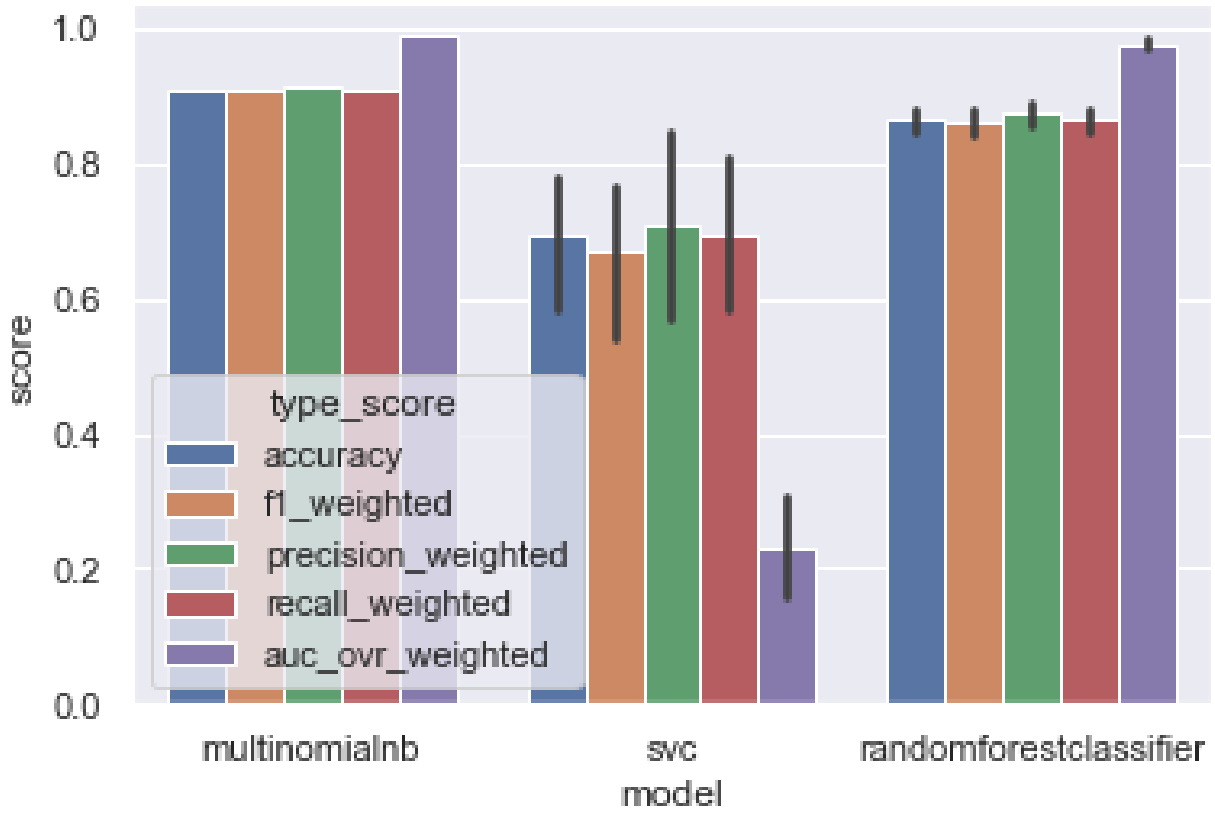
- Naive Bayes : Probabilité d'appartenance à une classe.
- Support Vector Machine : Séparation par hyperplans à marge souple maximale (régularisation l_2)
-  aléatoire : Arbres de décision tirés avec remise (bootstrap).
  Coupure : minimise l'impureté des noeuds fils avec fraction de features ($\sqrt{p}$) tirés aléatoirement.

Évaluation modèles l'encodage TF-IDF

Jeu d'entraînement stratifié en 5 parties.
GRIDSearch :

- SVM :
  - Paramètre régularisation : $C = [10^{-6}...10^{-2}]/5$,
  - tolérance pour critère d'arrêt : $tol = [10^{-5}...10^{-2}]/5$.

- Forêt aléatoire :
  - Nbr d'arbre = 100 ($> 100 \not\Rightarrow$ meilleurs résultats),
  - Profondeur max : $(max\_depth) = [5, 25, 50]$,
  - Minimum d'observations pour coupure : $(min\_samples\_split) = [2, 5, 10]$.



=> Très bon résultats **Naive Bayes** et **Forêt aléatoire**

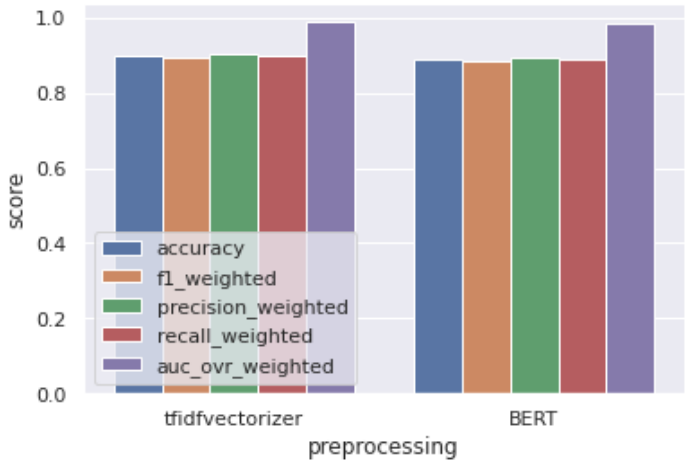
Table of Contents

Évaluation encodages

GRIDSearch :

- Nbr de Visual Word sur TF-IDF : [100, 1000, 2500, 4370 (max)]
- Foret aléatoire (Nbr d'arbre 100 ($> 100 \not\Rightarrow$ meilleurs résultats), $max\_depth = [5, 25, 50]$, $min\_samples\_split = [2, 5, 10]$.)
- ~~Naive Bayes~~ : n'accepte pas valeurs négatives (BERT)

	BERT - TF-IDF
accuracy	+0,71 %
f1	+0,80 %
precision	-1,33 %
recall	+0,71 %
auc_ovr	+0,11 %



=> Très bon résultats **BERT** et **TFIDF**

BERT : Erreur sur jeu TEST

=> Bons résultats conservés sur jeu TEST.

Matrice de confusion texte

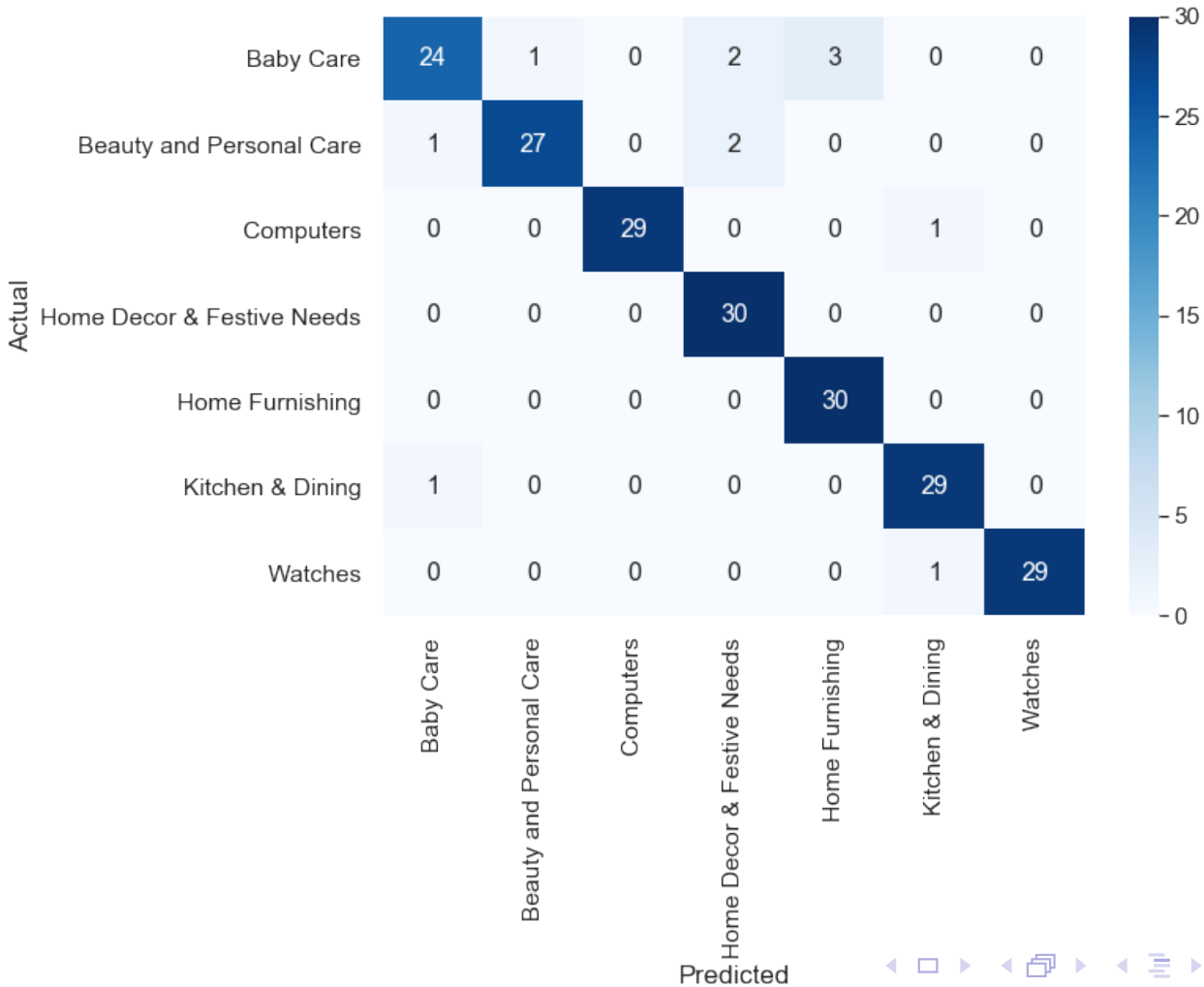


Table of Contents

Latent Dirichlet Allocation (LDA)

Méthode non-supervisée générative. Hypothèses :

- Chaque document du corpus est un bag-of-words
- Chaque document m aborde un certain nombre de thèmes dans différentes proportions qui lui sont propres $p(\theta_m)$;
- Chaque mot possède une distribution associée à chaque thème $p(\phi_k)$.

Thème 0 :	babi detail girl fabric dress sleev cotton neck boy shirt
Thème 1 :	mug coffe bring perfect gift design broadcast cupcak sip forget
Thème 2 :	kadhai thi bowl cushion white cover bless kd2 sport macbook
Thème 3 :	girl detail babi onli short fabric ship neck dress guarante
Thème 4 :	mug ceram coffe rockmantra one thi perfect love safe gift

=> Reconnaissance approximative des classes

Table of Contents

Segmentation BERT

Bon coefficient de silhouette de **0.17** (0.06 pour tf-ifd) pour 8 clusters sur KMeans.

=> Identification **claire** de groupes.

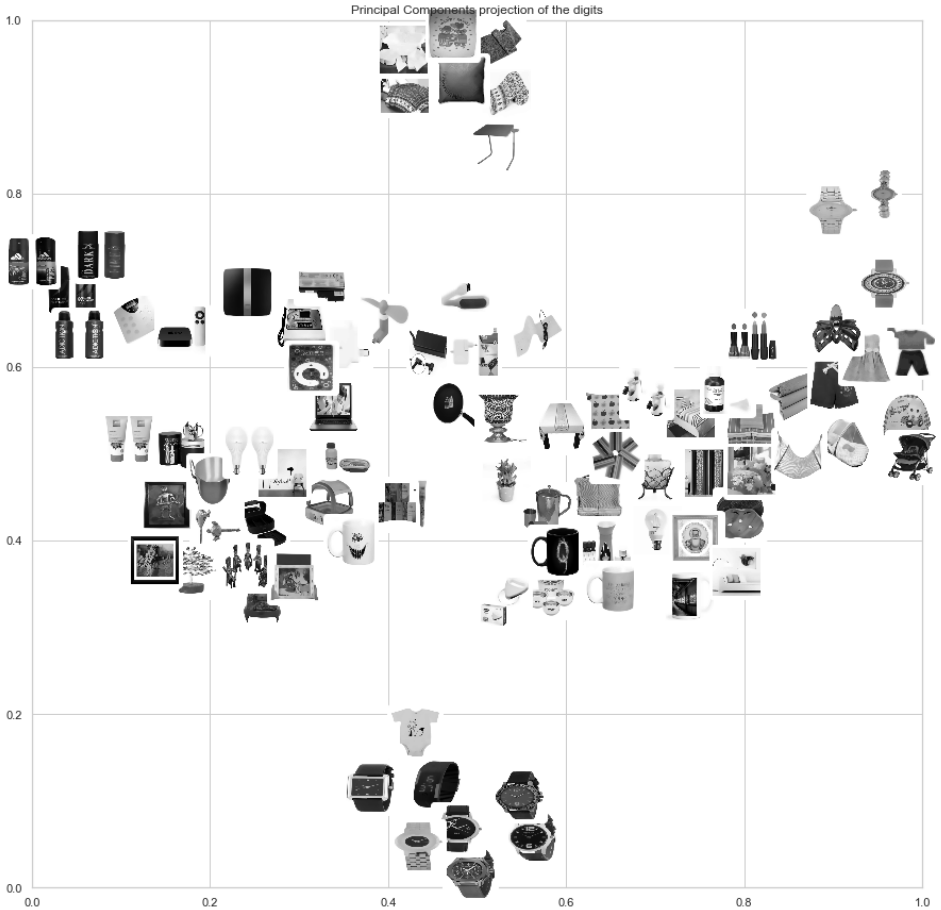
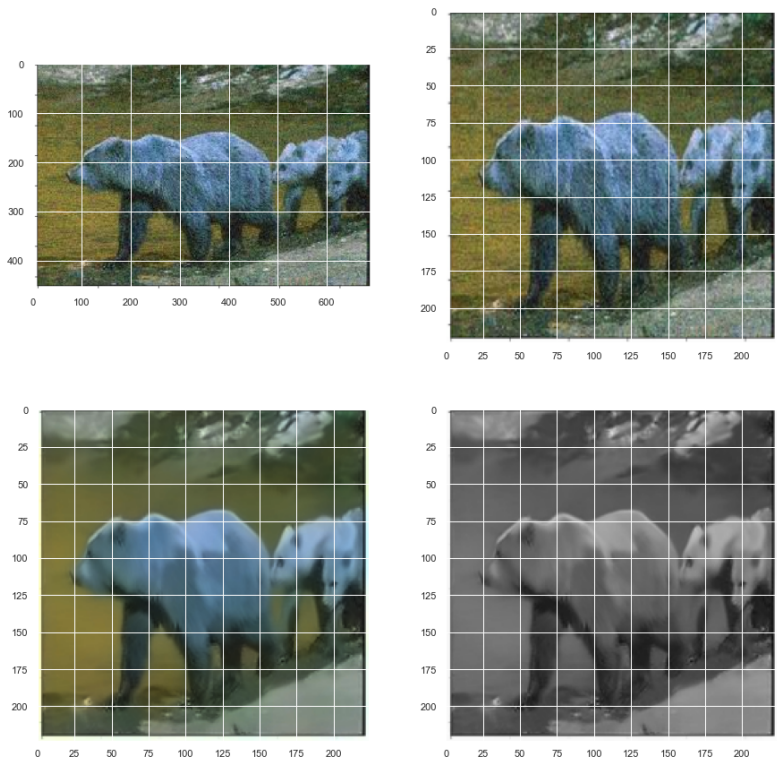


Table of Contents

Pré-traitement images



- Re-dimensionner en 224×224 pixels.
- Filtre "non-local means" : Moyenne tous les pixels pondérés par similarité avec pixel cible.
- Conversion niveau de gris.

Extraction caractéristiques

- Sac de mots visuels (BOVW : Bag of Visual Words)
- VGG16 : Réseau de Neurones par Convolution (CNN : Convolutional Neural Network) pré-entraîné

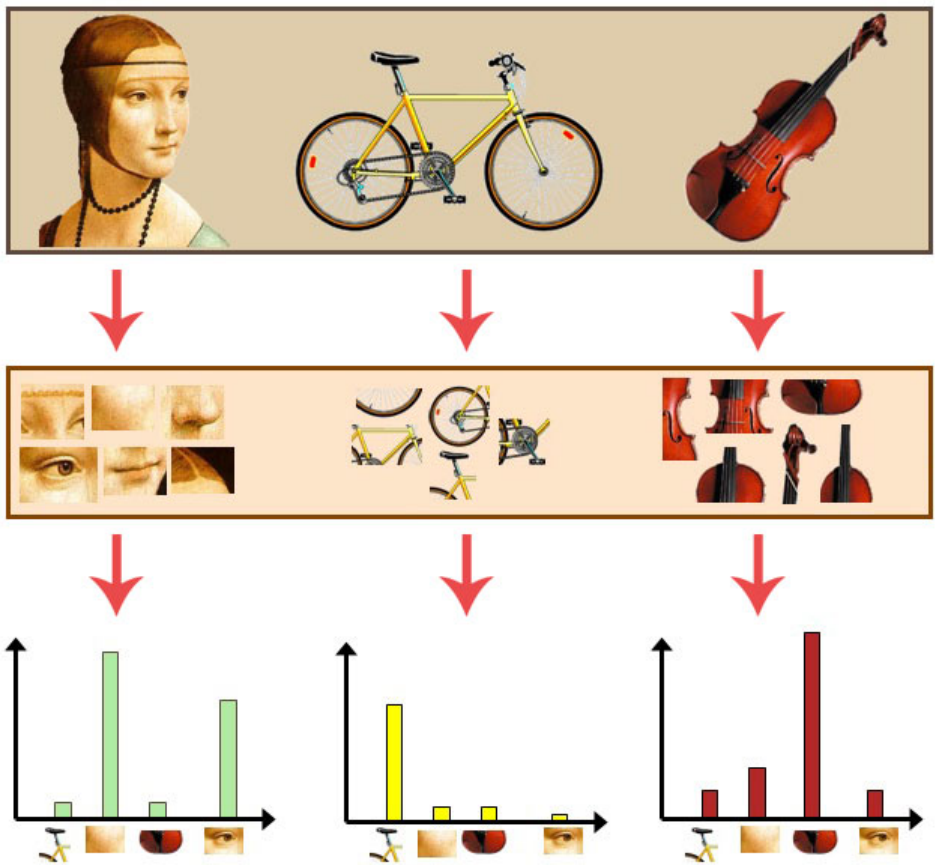
Table of Contents

Bag of Visual Words (BOVW)

Visual Words : segmentation des couples [point d'intérêt,descripteur].

* *Point d'intérêt (feature)* : centre d'ellipses + ou - grandes
* *Descripteur* : caractérise la zone du point d'intérêt (invariant par rotation, échelle, illumination).
* *SIFT* ou *ORB (open source)* : détection et description automatique des [points d'intérêt, descripteurs].

Dernière étape : Segmentation des descripteurs (KMeans) = *bag of Visual Words*



Histogramme de fréquence de *visual word* (*Term-Frequency (TF) du BOW*)

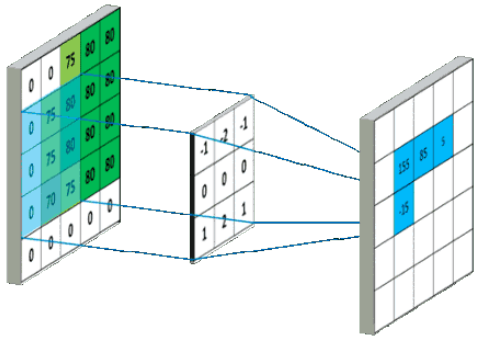
=> Classification ou segmentation avec l'histogramme

Table of Contents

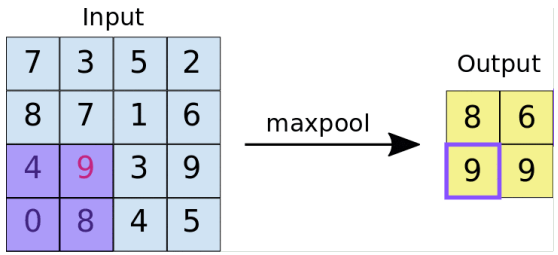
VGG16 : CNN pré-entraîné

Principales couches du CNN : *convolution*, *Pooling*, *RELU* et *Fully-connected*.

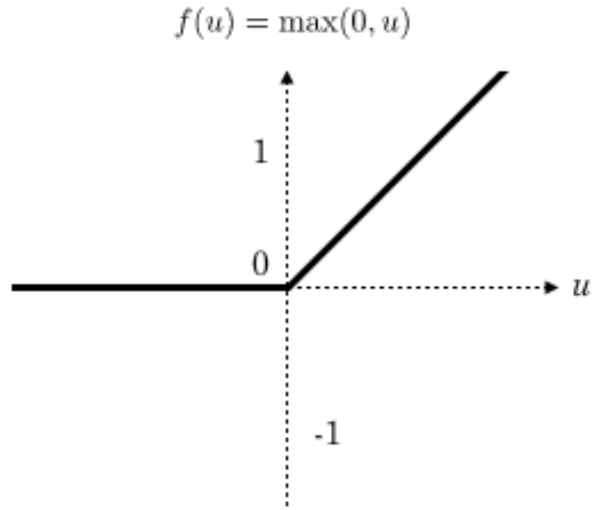
- Convolution :



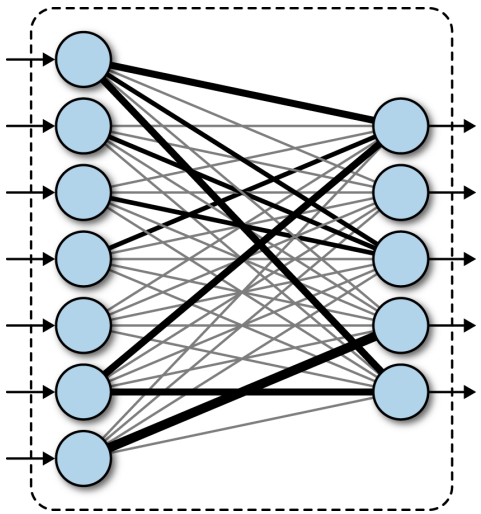
- Pooling : Réduction la taille des images



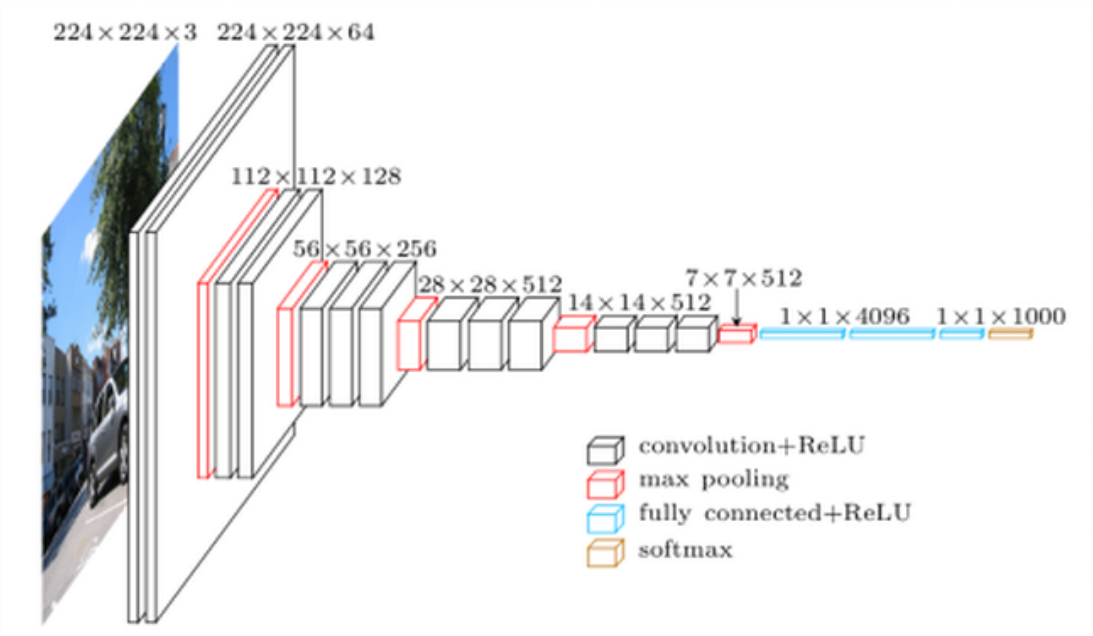
- RELU :



- Fully-connected :



Méthode : **Transfer learning** sur VGG 16 : CNN pré-entraîné de 1000 classes sur *ImageNet* (BDD d'images d'objets).



Utilisation de VGG16 pré-entraîné avec remplacement du classifieur (dernière couche) par le notre (7 classes).

Table of Contents

Classification BOVW

Forêt aléatoire (arbres=100,$max\_depth = None$, $min\_samples = 2$) dans GRIDSearch avec nbr de VW = [2000, 4000, 6000]

=> Résultats BOVW décevants

Table of Contents

VGG16 en validation croisée avec comme paramètres :

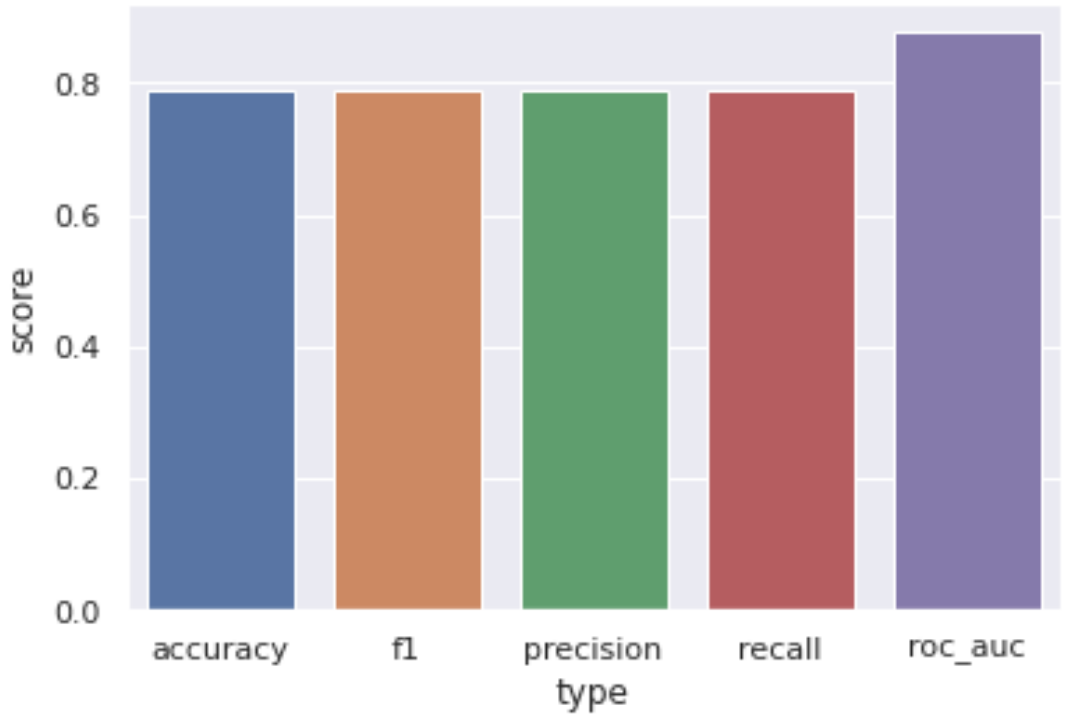
- Fonction de perte multi-classe :
  $loss = categorical\_crossentropy$
- Retro-propagation du gradient : $optimizer = adam$:
  correction poids avec estimation premier (moyenne) et second (variance) moment du gradient.

=> Moins sensible à la vitesse d'apprentissage

- Métrique pour fonction de perte $metric = accuracy$
- Époques : $epochs = 9$: nbr de passage du jeu d'entraînement dans le réseau
- EarlyStopping avec $patience = 2$: retient l'époque (et les poids associés) où l'erreur est minimale sur le **jeu de validation**.
  Le CNN *patiente* jusqu'à 2 époques une fois le minimum d'erreur trouvé.

=> Réduction du sur-apprentissage.

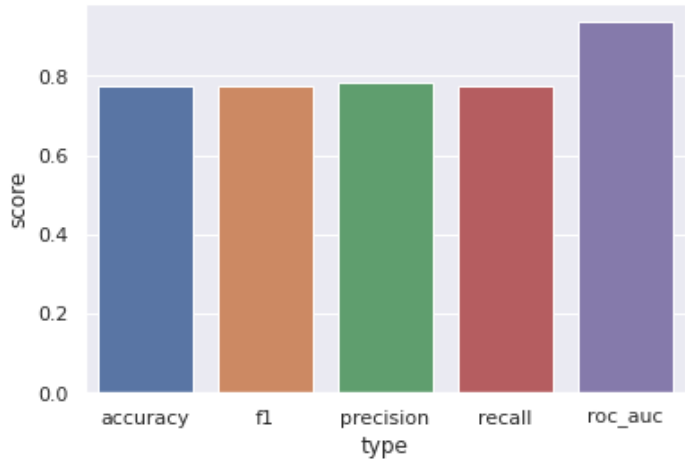
Nbr d'époques (EarlyStopping) sur les 5 plis : [4,3,3,7,3] soit 4 en moyenne.



=> Bon Résultats sur validation croisée VGG16

Ré-entrainement sur tout le jeu d'entraînement avec nbr époques = 4. Évaluation sur jeu **test** :

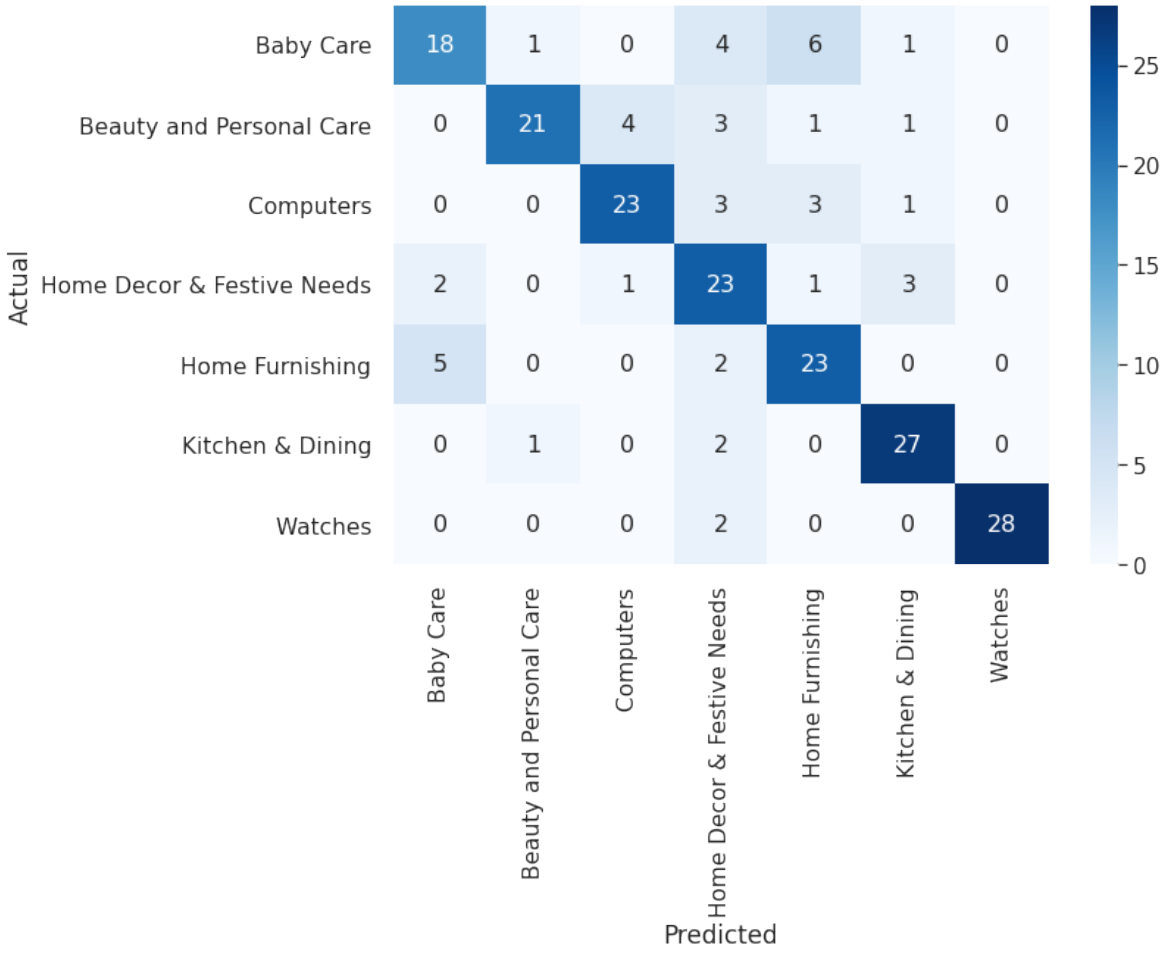
	VGG16 - BERT
accuracy	-6 %
f1	-6 %
precision	-6 %
recall	-6 %
auc_ovr	-5 %



=> Résultats VGG16 restent bons sur jeu test.

=> Mais moins bons que BERT.

Matrice de confusion VGG16 :



Segmentation VGG16

Coefficient de silhouette de **0.06** (moyen / 0.17 pour BERT) sur 12 clusters avec KMeans.

=> Identification de quelques groupes.

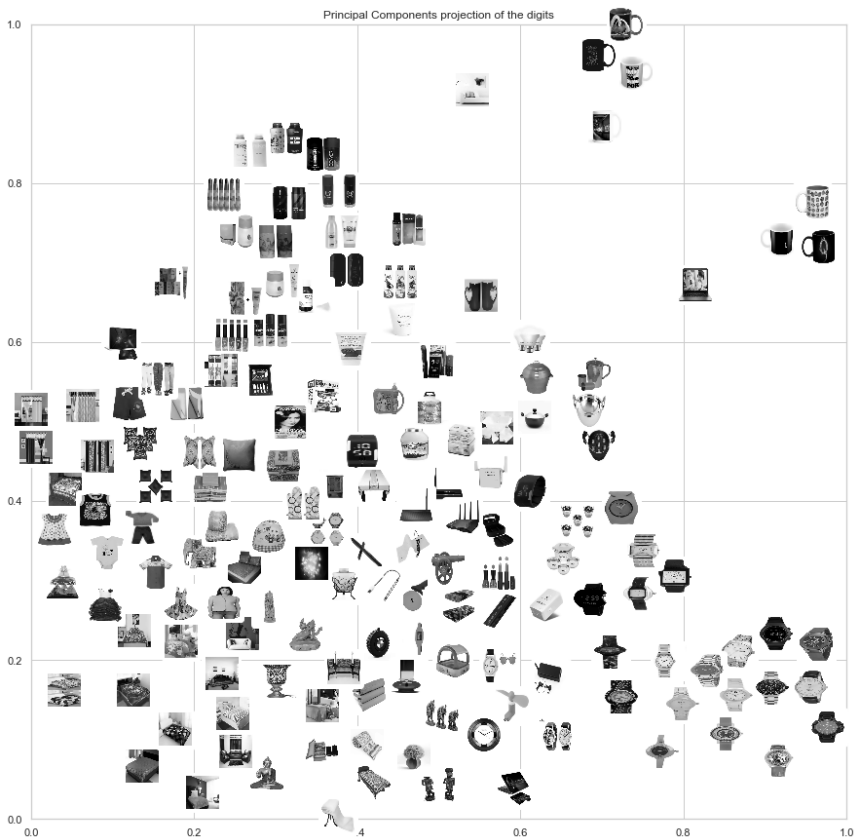
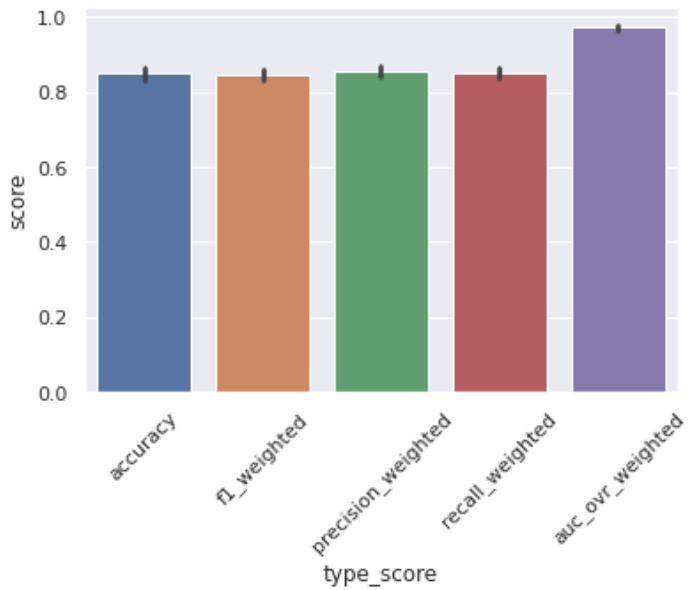


Table of Contents

Classification FULL : images ET texte

- Extraction features texte avec BERT (768 colonnes)
- Extraction features avec VGG16 (20000 colonnes) : ACP pour réduire la dimension à 768 (90% d'inertie)

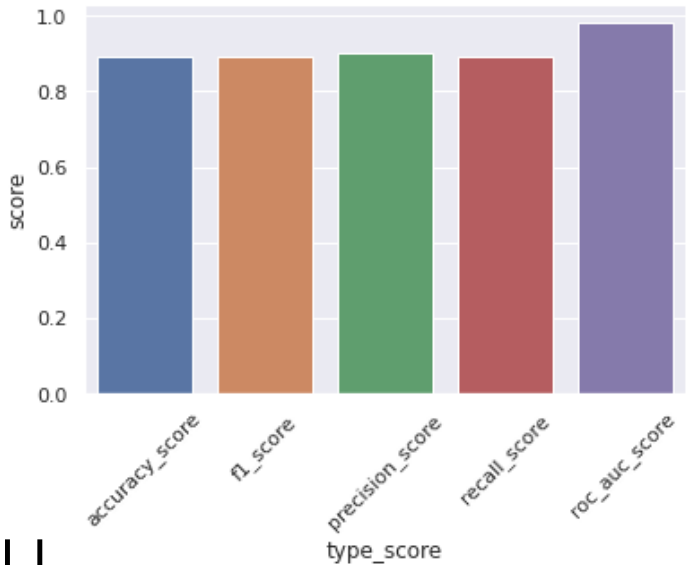
Forêt aléatoire sur GRIDSearch (nbr d'arbres = 1000, $max\_depth = [2, 10, None]$, $min\_samples\_split : [2, 5, 10]$)



Évaluation sur jeu de test :

Comparaisons scores

	FULL - texte	FULL - images
accuracy	+6 %	+12 %
f1	+6 %	+12 %
precision	+6 %	+12 %
recall	+6 %	+12 %
auc_ovr	+0 %	+3 %



=> Amélioration significative avec FULL

Segmentation FULL

Coefficient de silhouette de **0.06** (0.17 pour BERT) pour 14 clusters avec KMeans.

=> Identification de quelques groupes.

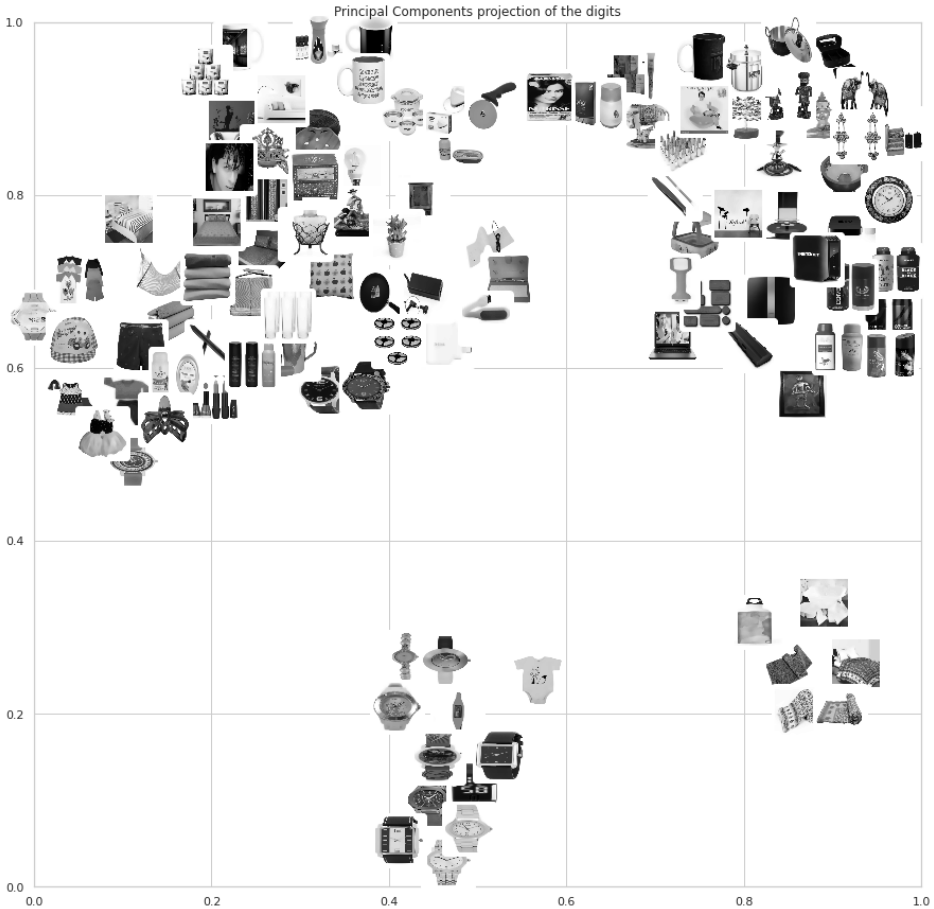


Table of Contents

Conclusion

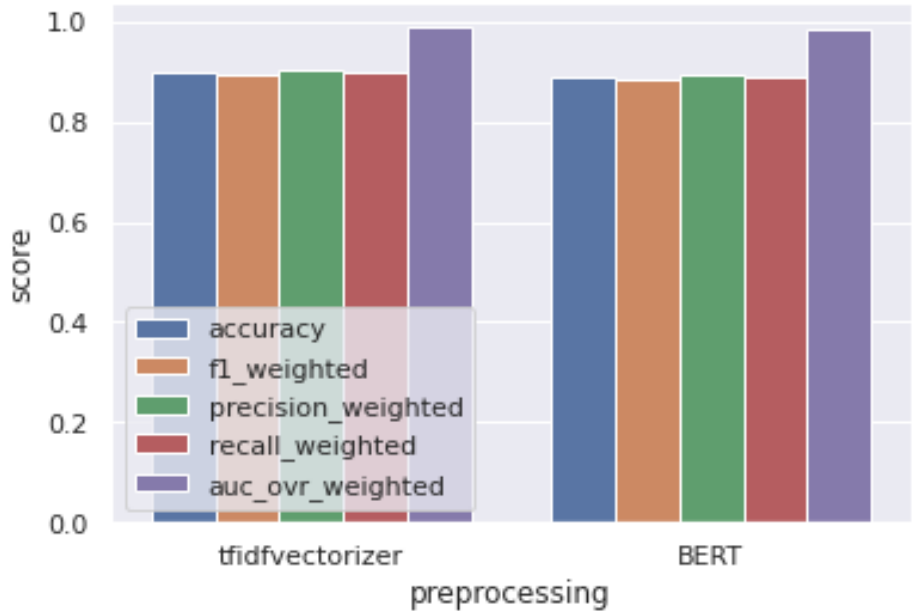
Des segmentations et classifications ont été réalisées avec :

- BERT et BOW Tf-Idf pour les **descriptions**,
- VGG-16 et BOWV ORB pour les **images**.

Segmentation :

- Descriptions : **BERT** a le meilleur coefficient de silhouette (0.17) et permet d'identifier clairement les clusters
- Images : VGG-16 ne segmente pas aussi bien que **BERT**.
- FULL (BERT+VGG-16) : coefficient de silhouette faible (0.06) mais identification claire de de quelques clusters

Classification sur les descriptions :



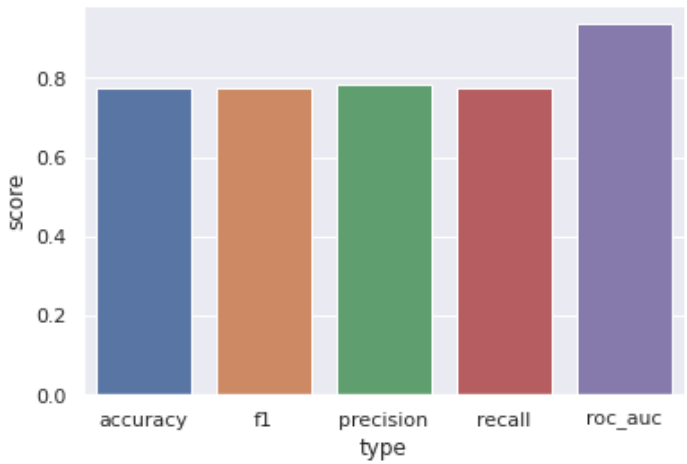
=> **BERT** et **BOW Tf-Idf** ont de très bons résultats

Classification sur les images :

=> **VGG16** est de loin meilleur que BOVW,

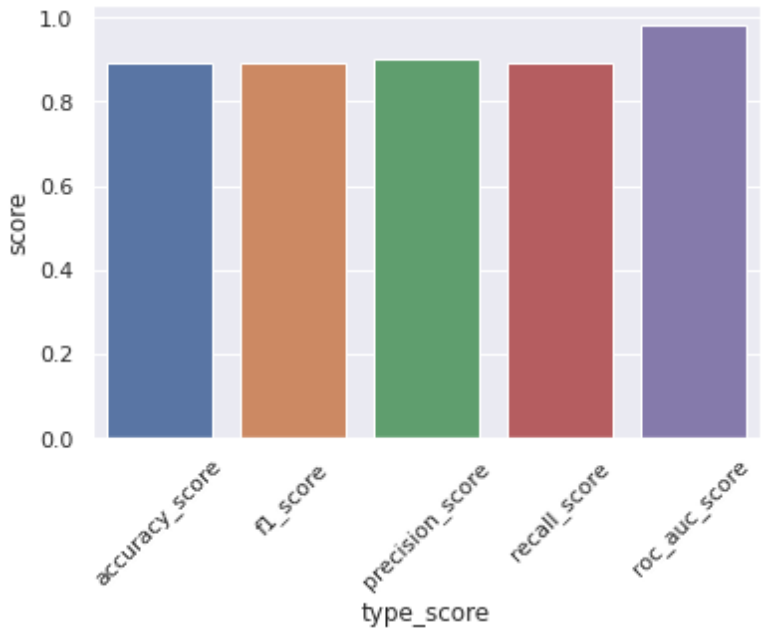
=> mais moins bon que **BERT**

	VGG16 - BERT
accuracy	-6 %
f1	-6 %
precision	-6 %
recall	-6 %
auc_ovr	-5 %



Classification avec FULL (images + descriptions)

	FULL - texte	FULL - images
accuracy	+6 %	+12 %
f1	+6 %	+12 %
precision	+6 %	+12 %
recall	+6 %	+12 %
auc_ovr	+0 %	+3 %



=> Résultats améliorés avec FULL

=> L'étude de faisabilité pour classifier automatiquement les produits par les images et les descriptions est tout à fait convaincante : **90%** de produit correctement classés

Perspectives

Projet :

- Refaire l'analyse avec plus de classes (en intégrant les sous-classes).
- Implémenter le moteur de classification.

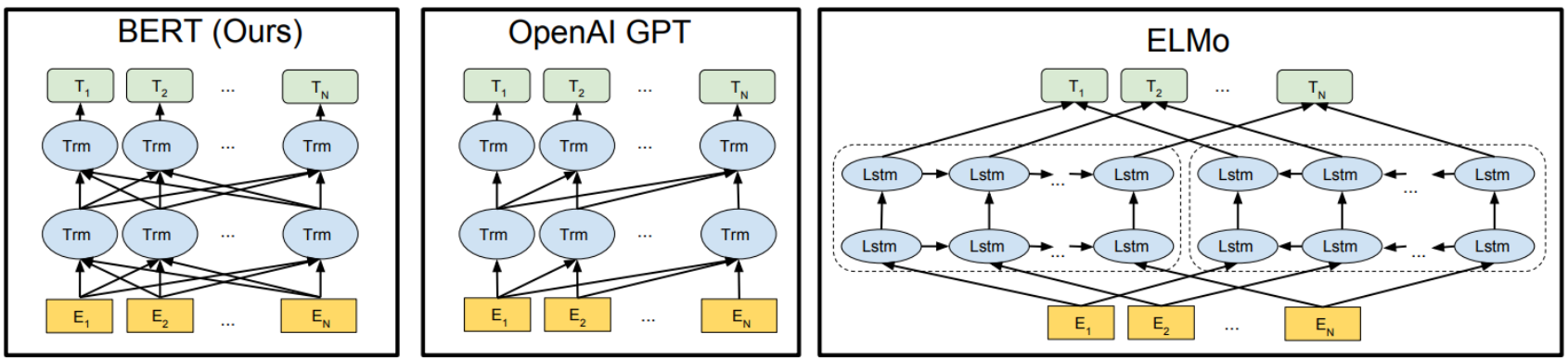
Personnellement :

- Les CNN sont **déterminants** pour la classification d'images.
- Analyser le texte par BOW **avant** de complexifier avec BERT

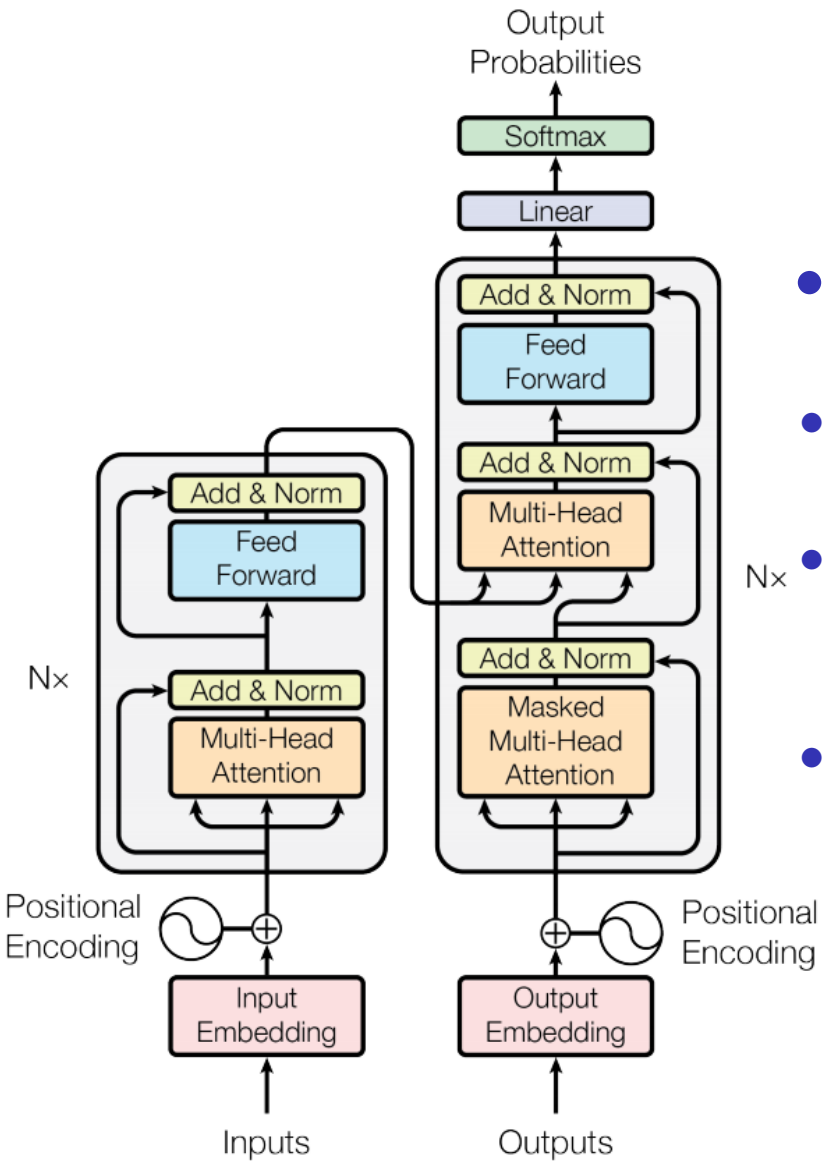
Merci de votre attention

BERT : précisions

Entièrement bidirectionnel

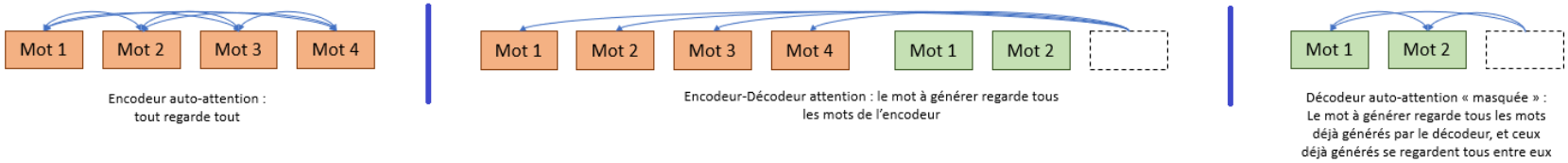


Transformers (*Attention is all you need*) :



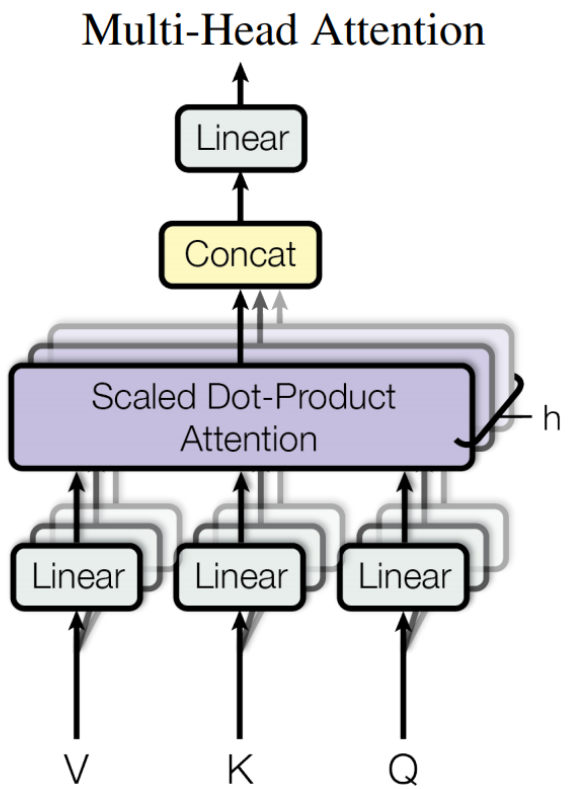
- Input Embedding : plongement de mot (dimension 512)
- Positionnal Embedding : information de position dans la phrase
- Encodeurs (gauche) : couches d'auto-attention "multi-têtes" + feed-forward neural networks (FFN)
- Décodeurs (droite) : couches d'auto-attention masquée + couches attention encodeur-decodeur + FFN

3 types d'attention multi-tête dans le transformer :



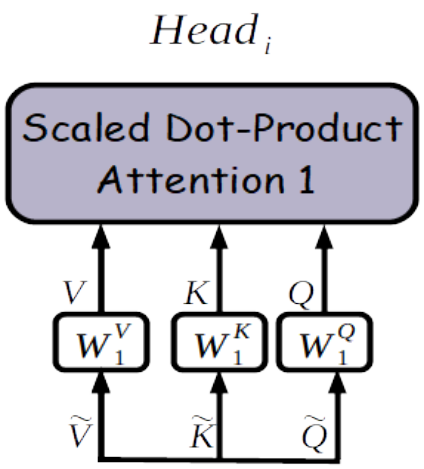
L'attention est **Multi-têtes** :

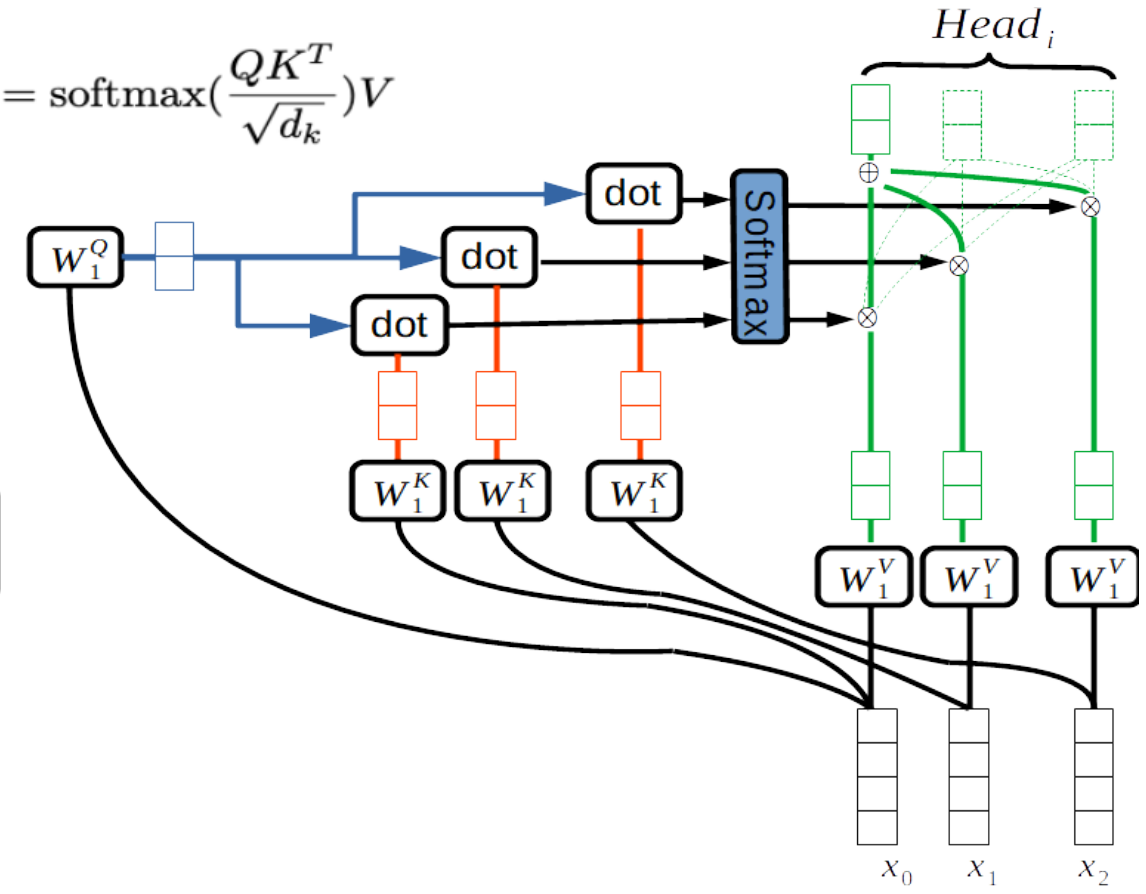
- Concaténation de "têtes"
- Chaque tête fait des projections / combinaisons des mots sous forme de :
  - Requêtes (Q)
  - Clés (K)
  - Valeurs (V)



Une tête :

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$





Tâches automatiques

- **MLM** : 15 % des mots en entrée sont masqués aléatoirement :
  - 80 % : jetons [MASQ] ,
  - 10 % : mots aléatoires,
  - 10 % : mots laissés tels quels.

  .
- **Prédiction de la phrase suivante**