Classification multi-classes de descriptions d'emploi

Pol Labarbarie et Mehdy Hounkonnou

Université de Bordeaux

Vendredi 15 Janvier 2021

Introduction

Qu'est-ce que le NLP ?

1. NLP (Natural language processing) est un sous-domaine de l'informatique, de la linguistique et de l'IA.
2. Compréhension du contenu des documents pour diverses applications dont la classification.
3. Exemples : Siri (Apple), Google Translate, Alexa (Amazon).

Le Défi IA

1. Classification multi-classes avec 28 classes au total.
2. Données extraites de CommonCrawl, formant le modèle GPT-3 d'OpenAI à priori biaisé.
3. Objectif : Concevoir une solution qui est à la fois précise et équitable.

Conditions

Tous les temps de calcul sont obtenus avec un processeur Intel-Core i7-7820HQ (8 coeurs 2.9 GHz, 3.9GHz Turbo) aka "The quiet BEAST".

Plan

Statistiques descriptives

Descriptif du jeu de données

- 3 jeux de données : apprentissage, test et label.
- Nous disposons de descriptions d'emploi et l'objectif est de prédire un métier.
- Le jeu de données apprentissage : $n = 217197$ lignes (descriptions).
- Le jeu de données test : $n = 54300$ descriptions.
- Cadre d'apprentissage supervisé.

Caractéristiques du jeu de données

- Aucune donnée manquante.
- Présence de langues autres que l'anglais :

  03 - "Dr. Maria Ignasia Tjahjadi practices at Rumah Sakit Ibu dan Anak Hermina in Tanjung Priok, Jakarta. She completed S.Ked. from Atma Jaya University in 1996."

  - "Soy periodista con 10 años de experiencia trabajando en medios digitales. He sido redactora web y coordinadora de redacción. Tengo conocimientos en lenguaje..."
- Au total : 93 descriptions avec des mots non anglais.

Vocabulaire du jeu de données

- Présence de chiffres.
- Présence de mots sans signification dans le corpus.
- Présence de combinaisons de mots et de chiffres dépourvu de sens lexical ou de symboles. Exemple : '2006music', '__', 'aaaahc'

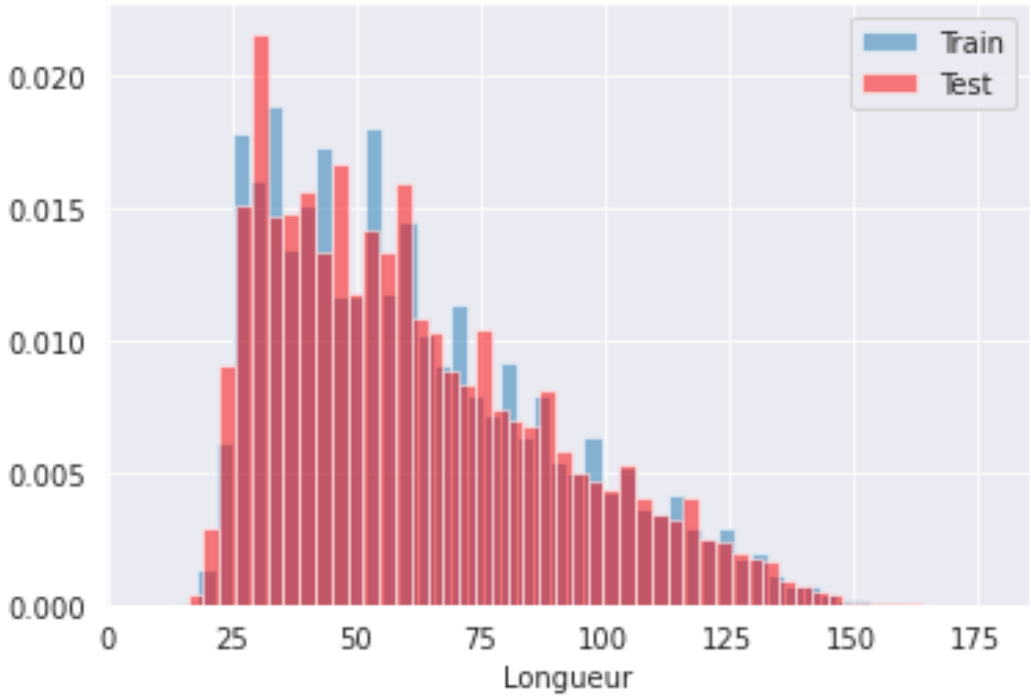


Figure: Histogrammes des données brutes

Statistiques descriptives

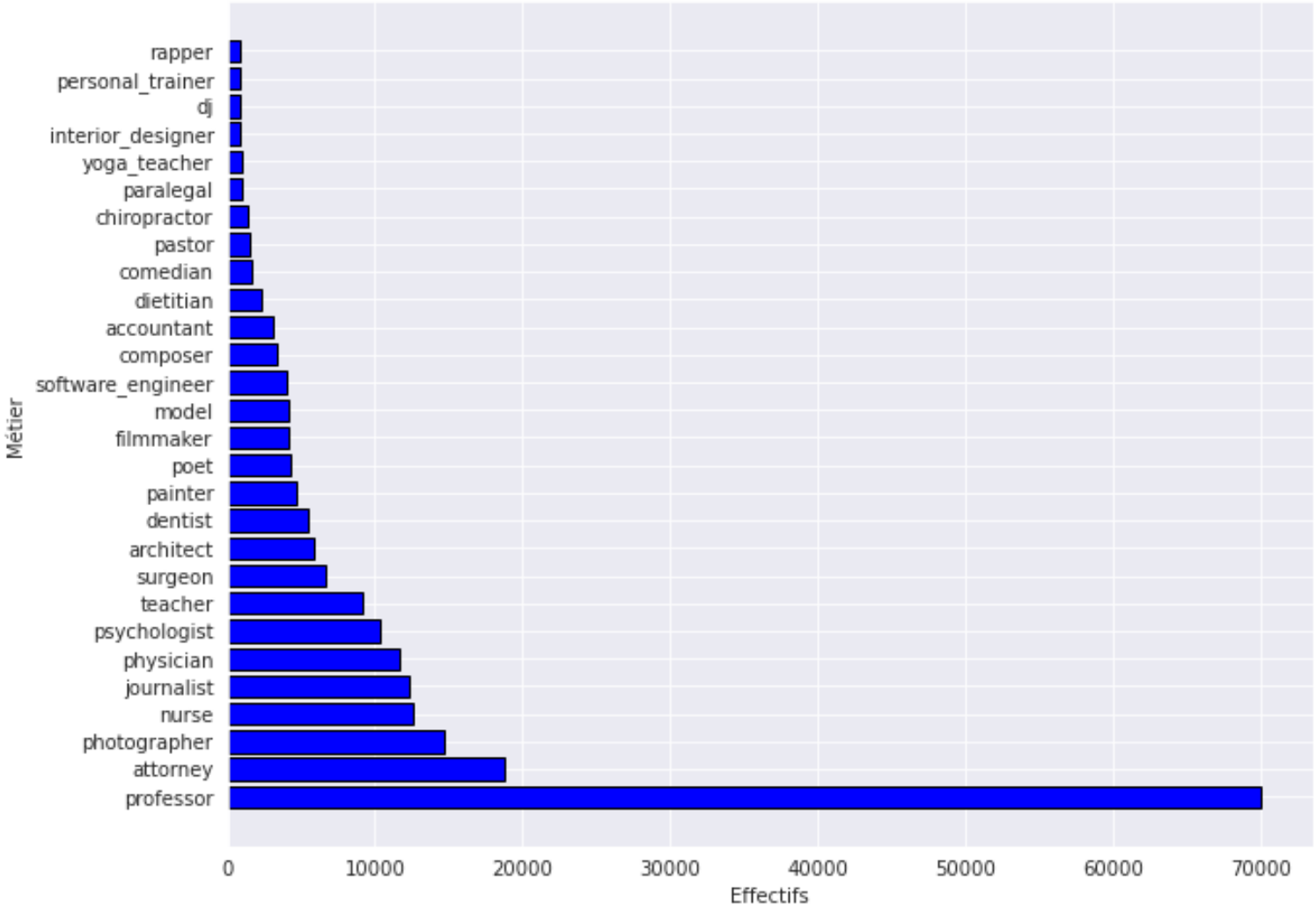


Figure: Histogrammes des répartitions des emplois

Traitement

1. Mise en minuscule des descriptions.
2. Création d'un dictionnaire des abréviations pour les étendre dans les descriptions. Exemple: ("gf" = "girlfriend","cuz" = "because","coz" = "because")
3. Suppression des caractères spéciaux.
4. Lemmatisation.

	original_word	lemmatized_word
0	trouble	trouble
1	troubling	trouble
2	troubled	trouble
3	troubles	trouble

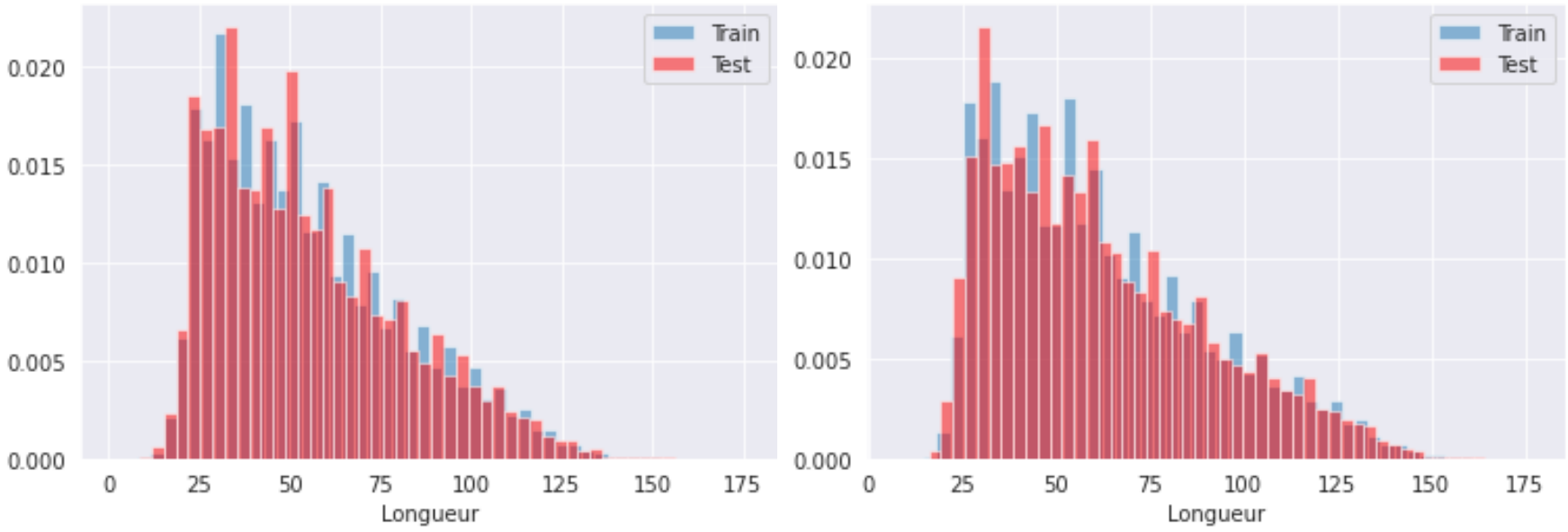
	original_word	lemmatized_word
0	goose	goose
1	geese	goose

Figure: Exemples du processus de lemmatisation

5. Suppression des mots non-anglais.

Traitement

1640173 mots non anglais enlevés.
Nous remarquons clairement une réduction de la longueur des phrases.



(a) Histogramme des données lemmatisées

(b) Histogramme des données brutes

Figure: Comparaison des histogrammes des longueurs des descriptions

Plan

La Baseline

1. Pas de traitement des descriptions.
2. Pondération TF-IDF appliquée au jeu de données.
3. Régression logistique avec pénalité Ridge.

Résultat : Taux de bonne classification 0.72.

TF-IDF qu'est-ce que c'est ?

TF-IDF (term frequency-inverse document frequency) est une mesure statistique qui évalue la pertinence d'un mot pour un document appartenant à une collection de documents.

TF-IDF Mathématiquement

Pour le mot i dans le document j, on pose $tf_{i,j} = \frac{n_{i,j}}{\sum_n n_{i,j}}$

$$w_{i,j} = tf_{i,j} \log(\frac{N}{df_i})$$

$n_{i,j}$ nombre d'apparitions du mot i dans le documents j,
N le nombre total de documents,
df_i le nombre de documents contenant le mot.

La Baseline

Résultats

La baseline donne un score sur Kaggle de 0.73244. Les résultats en local sont affichés à droite.

Accuracy: 0.78
Auc: 0.98
Detail:

	precision	recall	f1-score	support
0	0.66	0.42	0.52	293
1	0.79	0.74	0.76	818
2	0.75	0.57	0.65	182
3	0.53	0.53	0.53	1881
4	0.80	0.62	0.69	146
5	0.81	0.78	0.79	921
6	0.65	0.75	0.69	2516
7	0.83	0.49	0.62	188
8	0.82	0.68	0.74	1330
9	0.81	0.65	0.72	632
10	0.77	0.59	0.67	161
11	0.73	0.69	0.71	2339
12	0.84	0.66	0.74	342
13	0.69	0.61	0.65	797
14	0.87	0.80	0.83	2532
15	0.75	0.66	0.70	845
16	0.94	0.88	0.91	1096
17	0.88	0.55	0.67	284
18	0.79	0.74	0.77	775
19	0.81	0.89	0.85	13898
20	0.81	0.88	0.84	2878
21	0.84	0.60	0.70	152
22	0.76	0.67	0.71	2151
23	0.92	0.27	0.42	209
24	0.74	0.59	0.66	1162
25	0.80	0.80	0.80	670
26	0.83	0.87	0.85	3779
27	0.88	0.76	0.81	463
accuracy			0.78	43440
macro avg	0.79	0.67	0.71	43440
weighted avg	0.78	0.78	0.78	43440

Figure: Résultats de la baseline

La Baseline

Résultats

La matrice de confusion de la baseline est la suivante. On note qu'il se trompe beaucoup entre les professeurs, les teachers (enseignants) et les psychologues (classe 19, 3 et 22).

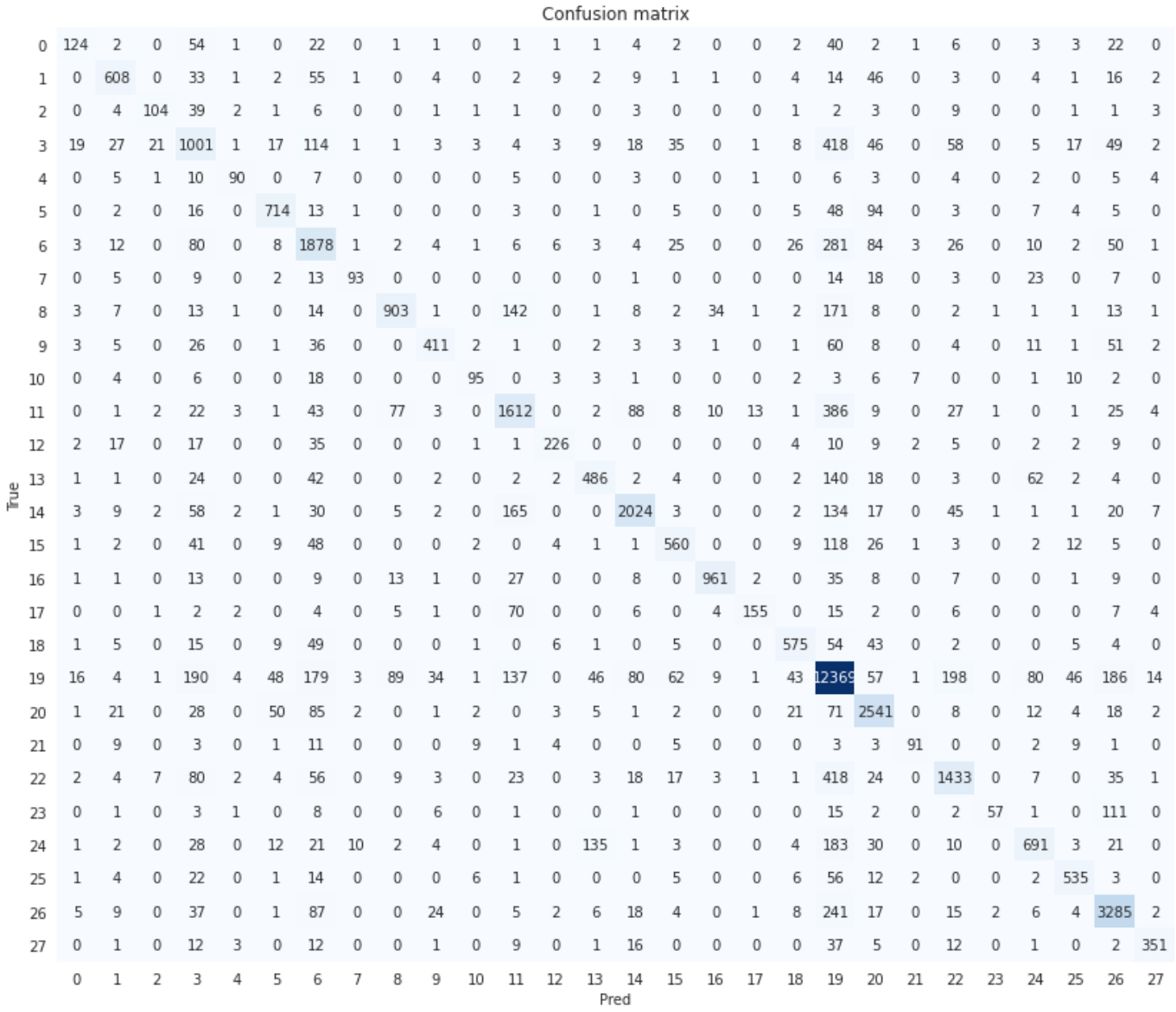


Figure: Matrice de confusion de la baseline

Comment améliorer la Baseline ?

Réduction de dimensionalité

L'ACP ne conduisant pas à de bons résultats. → Sélection de variables par test du khi-deux.

→ Permet d'obtenir un modèle plus parcimonieux mais plus interprétable que pour une ACP.

Agréger des prédicteurs réduit la variance

Si on note pour tout $1 \leq l \leq q$ Z_l un prédicteur, on peut montrer que

$$Var\left(\frac{1}{q}\sum_{l=1}^{q} Z_l\right) = \sigma^2 \frac{1-\rho}{q} + \sigma^2\rho \leq \sigma^2$$

où $Var(Z_l) = \sigma^2$ et $corr(Z_l, Z_{l'}) = \rho$

Choix du prédicteur

Machine à vecteurs de support (SVM) conduit à des meilleurs résultats (quelques %) que la Régression logistique.

→ Utilisation de Machine à vecteurs de support.

Baseline améliorée

Méthodologie

1. TF-IDF sans stop-words et avec notre corpus traité.
2. Choix du nombre de variables conservées du test du khi-deux obtenu par validation croisée 5-folds.
3. Agrégation (Bagging) de 100 Machines à vecteurs de support.

Résultats de la baseline améliorée

Les résultats de la baseline améliorée en local sont les suivantes, donnant un score sur Kaggle de 0.74231.

	precision	recall	f1-score	support
0	0.64	0.49	0.56	293
1	0.75	0.74	0.74	818
2	0.67	0.65	0.66	182
3	0.55	0.49	0.52	1881
4	0.68	0.64	0.66	146
5	0.77	0.79	0.78	921
6	0.66	0.73	0.69	2516
7	0.77	0.54	0.64	188
8	0.78	0.70	0.73	1330
9	0.75	0.66	0.70	632
10	0.76	0.60	0.67	161
11	0.71	0.69	0.70	2339
12	0.82	0.68	0.74	342
13	0.65	0.58	0.61	797
14	0.86	0.79	0.83	2532
15	0.71	0.67	0.69	845
16	0.91	0.90	0.90	1096
17	0.85	0.61	0.71	284
18	0.74	0.76	0.75	775
19	0.81	0.88	0.85	13898
20	0.82	0.86	0.84	2878
21	0.76	0.64	0.69	152
22	0.76	0.67	0.71	2151
23	0.77	0.30	0.43	209
24	0.70	0.58	0.64	1162
25	0.80	0.78	0.79	670
26	0.84	0.86	0.85	3779
27	0.82	0.81	0.82	463
accuracy			0.78	43440
macro avg	0.75	0.68	0.71	43440
weighted avg	0.78	0.78	0.77	43440

Figure: Résultats de la baseline améliorée

La Baseline améliorée

Résultats

La matrice de confusion de la baseline améliorée est la suivante. On note à une légère amélioration près les mêmes confusions de classes(19,3, 22..).

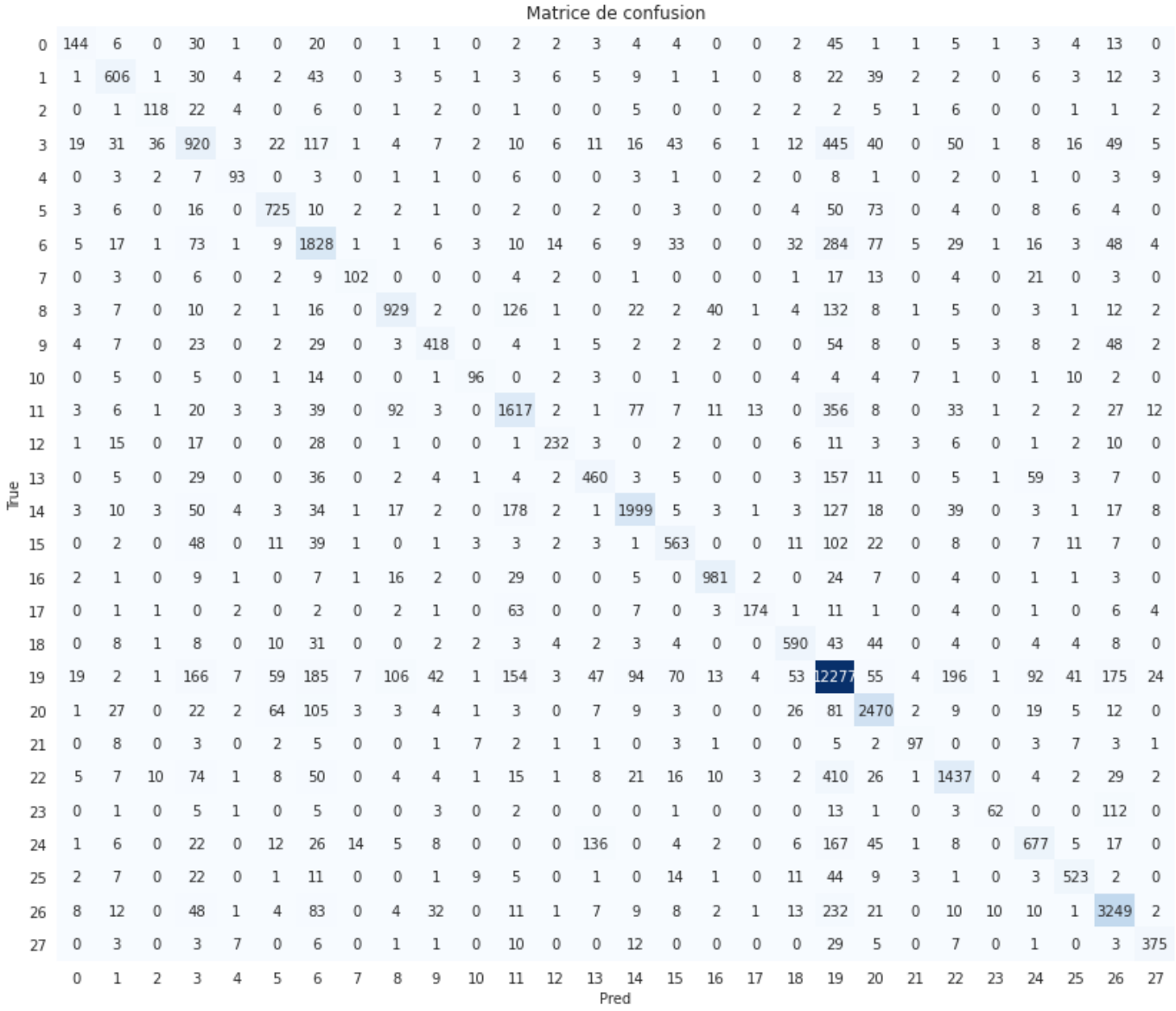


Figure: Matrice de confusion de la baseline améliorée

Perspectives d'améliorations

Essai sans succès

Même méthodologie que précédemment mais en unifiant les données. $\rightarrow$ Pas de bons résultats.

Améliorations

- Le test du khi-deux est effectué sur 40000 variables, il faudrait utiliser des méthodes de multiplicité des tests (ex : Bonferroni).
- L'agrégation de prédicteurs permet d'obtenir de meilleurs résultats mais n'améliore pas significativement les résultats, nous ne faisons que réduire la variance.
- D'après (Joulin et al. 2016), les méthodes standards comme les SVM et les RL sont limités en terme de performance pour la classification de document.

Solution

$\rightarrow$ Les réseaux de neurones profonds.

Plan

L'état de l'art

D'après (Bojanowski 2019; Neveu n.d.), les méthodes d'apprentissage profond sont l'état de l'art pour la classification de document.

Plongement lexical (Word-embedding)

Associe à chaque mot d'un vocabulaire un vecteur de nombre réels généralement de taille 300.
→ Cette représentation permet de capter la sémantique des mots.

- Word2Vec (Mikolov et al. 2013)
- GloVe (Pennington and Socher 2014)

- Modèle entraîné à : Selon un mot prédire c mots de contexte autour de lui.
- Fonction softmax en sortie.
- Par rétropropagation du gradient on obtient les formules de mise à jour des poids de la couche W et W'.

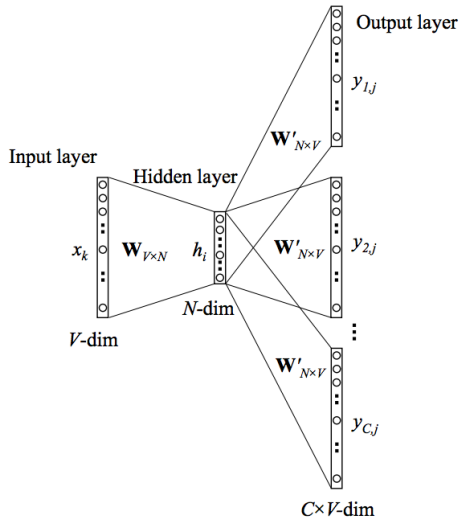


Figure: Architecture du modèle Skip-Gram

Représentation vectorielle et LSTM

Utilisation d'une représentation vectorielle précise → Téléchargement des vecteurs de mot.

Méthodologie

Utilisation du modèle de l'article vu en cours (Pietro n.d.).

1. Conversion de notre jeu de données apprentissage et test en index se référant au vocabulaire de la représentation vectorielle.
2. Entraînement d'un réseau avec deux couches LSTM sur 15 epochs.

Résultats

- 1 epoch = 45min - 1h.
- Taux de bons classements sur les données Kaggle d'environ 0.71.
- Difficultés à la généralisation → Transformer.

DistilBERT

Les Transformers c'est quoi ?

C'est un empilement de couches aux architectures identiques. Chaque couche est composée d'une couche d'attention et d'une couche à propagation avant (Bojanowski 2019; Team n.d.).

DistilBERT est une version distillée de BERT. Plus rapide, plus économique.

Méthodologie

1. Utilisation des descriptions non pré-traitées.
2. Tokenization avec la propre fonction du réseau.
3. Chaque séquence (description tokénizée) est de longueur maximale 80. (padding, truncation)
4. Réglage fin (fine-tuning) du modèle sur 3 epochs sinon sur-apprentissage.

Résultats

- 1 epoch = 2h.
- Mieux que la Baseline mais pas mieux que notre Baseline améliorée
  → Nécessité d'utiliser un Transformer plus gros → Serveur de calcul.

Plan

Conclusion

C'est déjà la fin

- Les méthodes standards ne sont pas très performantes mais permettent d'obtenir une Baseline solide.
- Pour des résultats meilleurs, il faut utiliser un réseau Transformer.
- Plus le réseau est gros, meilleurs seront les résultats.
- Utilisation d'un autre réseau Transformer pour synthétiser des descriptions en plus pour rééquilibrer les classes.

Bibliographie

Pitor Bojanowski. *Le langage naturel*. https://www.college-de-france.fr/site/stephane-mallat/seminar-2019-02-20-11h15.htm. Février 2019.

Armand Joulin et al. "Bag of Tricks for Efficient Text Classification". In: *arXiv preprint arXiv:1607.01759* (2016).

Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

Thibault Neveu. *L'état de l'art du NLP - Julien Chaumond, CTO Hugging Face - Podcast IA*. https://www.youtube.com/watch?v=RL8QQk-LJp8t=314s.

Jeffrey Pennington and Socher. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

Mauro Di Pietro. *BERT for Text Classification with NO model training*. https://towardsdatascience.com/text-classification-with-no-model-training-935fe0e42180.

The Hugging Face Team. *Librairie Transformers Python*. https://huggingface.co/transformers/quicktour.html.