NOTE DE SYNTHÈSE

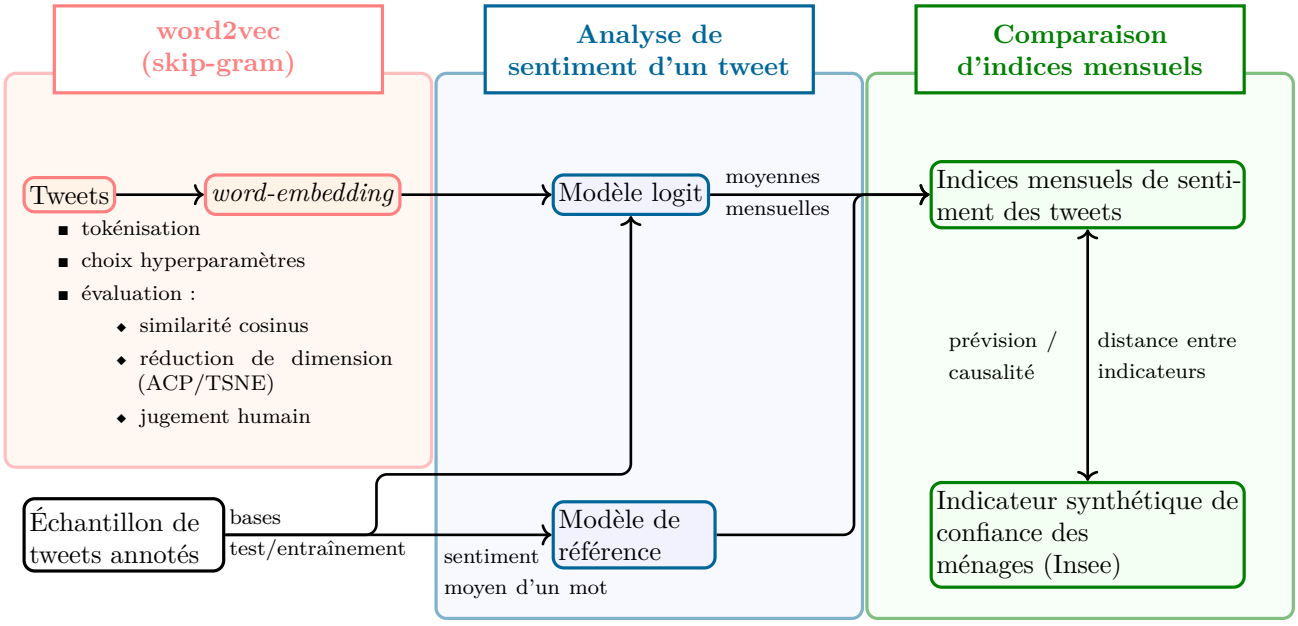
Word-Embedding et sentiments des ménages avec Twitter

Kim Antunez, Romain Lesauvage, Alain Quartier-la-Tente
sous l'encadrement de Benjamin Muller (Inria)

Grâce à l'évolution des méthodes d'apprentissage profond (*Deep Learning*), l'appréhension du langage naturel est aujourd'hui devenue une discipline à part entière (*Natural Language Processing*). Ce succès s'explique en partie grâce à l'émergence de techniques non supervisées d'apprentissage de représentation de structures linguistiques. Les méthodes de *word-embedding* (« plongement lexical » en français) permettent de représenter chaque mot d'un dictionnaire par un vecteur de nombres réels afin que les mots qui apparaissent dans des contextes similaires possèdent des vecteurs correspondants qui sont relativement proches (au sens d'une distance définie). Les modèles *word2vec*, développés par une équipe de recherche chez Google, sont parmi les plus célèbres.

En partie 1, nous décrivons le modèle *word2vec* que nous avons implémenté grâce à la librairie `Pytorch` de Python. Au-delà de la compréhension du modèle, nous nous sommes également initiés aux tests d'hyperparamètres et à son évaluation sur un corpus fictif (partie 2) grâce à plusieurs méthodes (calculs de similarités cosinus, opérations vectorielles sur les mots, méthodes de réduction de dimension ACP et T-SNE et jugement humain). Il a été fascinant d'observer à quel point le modèle présente d'excellents résultats en termes de capture sémantiques des mots dans un texte.

Dans la partie 3, nous avons cherché à construire un indice mensuel de sentiment des tweets et à le comparer aux indicateurs produits dans la statistique publique. L'entraînement et le test d'un modèle logit dont les prédicteurs correspondent aux dimensions des *word-embedding* nous a permis de remarquer qu'en plus de représenter la proximité entre mots, le modèle *word2vec* permet de capter dans une certaine mesure le sentiment de phrases.



Bien que l'indice obtenu s'avère utile pour prédire l'indicateur synthétique de confiance des ménages (Camme, Insee), les différences observées entre les deux séries demeurent importantes en raison de leurs philosophies différentes et des limites de la base d'entraînement de tweets annotés (*domain-shift*, processus d'annotation, mots inconnus) qui invitent à des pistes d'amélioration (prétraitement plus approfondi des tweets, modèles d'analyse de sentiment plus élaborés...).

1 Implémentation du modèle *word2vec*

Les modèles *word2vec*, développés par une équipe de recherche chez Google, sont parmi les plus célèbres modèles de *Natural Language Processing* qui utilisent le *word-embedding* — plongement lexical en français. Il s'agit de représenter chaque mot par un vecteur dont la dimension est fixée par la valeur d'un hyperparamètre.

L'approche *Skip-gram* a pour objectif d'estimer, pour chaque mot du vocabulaire qualifié de **« contextes »**, la probabilité d'être proche d'un mot qualifié de **« focus »** (figure 1). Ainsi, les mots qui apparaissent dans des contextes similaires (« bonjour » et « salut » par exemple) seront représentés par des vecteurs relativement proches dans l'espace vectoriel de définition de ces vecteurs.

Pour transformer chaque mot en un vecteur, nous entraînons un réseau de neurones sur une tâche annexe : on construit un classifieur dont la tâche de prédiction est binaire pour chacun des mots du vocabulaire et répond à la question « Est-ce que ce mot contexte est susceptible d'être voisin du mot focus ? ». Ce n'est pas la prédiction en elle-même qui nous intéresse, mais plutôt le poids du classifieur en sortie du modèle qui correspondra aux *word-embeddings* de dimension *dim*.

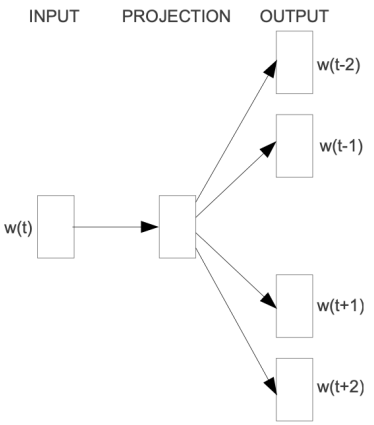


FIGURE 1 – Architecture du modèle Skip-gram.

Skip-gram est un réseau de neurones à deux couches avec deux matrices initialisées en générant des lois normales $\mathcal{N}(0,1)$:

- En entrée une matrice W_e de taille $n \times dim$;
- En sortie une matrice W_s de taille $n \times dim$.

Nous entraînons le réseau de neurones en le nourrissant de paires `[focus, contexte]` contenues dans les n mots du vocabulaire des tweets. Pour cela, on tire au hasard [1] un mot focus pour lequel on tire un mot *contexte* au hasard dans une fenêtre w. Ce mécanisme va être répété sur toutes les phrases du corpus et l'ensemble du corpus va être parcouru plusieurs fois. Le nombre de fois que l'ensemble du corpus est parcouru est appelé *epochs*.

À chaque étape, les matrices sont mises à jour par descente de gradient :

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta Loss(\theta^{(t)})$$

avec η le taux d'apprentissage et $Loss(\theta)$ la fonction de perte.

Le modèle *word2vec* a initialement été construit en utilisant une fonction de perte dérivée de la fonction *softmax*. L'inconvénient de cette méthode est qu'elle est très gourmande en temps de calcul ($\mathcal{O}(n)$ par phrase). L'algorithme a donc ensuite été amélioré en utilisant le *negative sampling*. Dans cette approche, on utilise une fonction de perte sigmoïde. De plus, plutôt que de mettre à jour l'ensemble des représentations vectorielles des mots pour chaque couple `[focus, contexte]`, on tire K mots au hasard du vocabulaire $(w_{neg,i})_{i=1..K}$ en considérant que ces mots ne seront pas des mots voisins de `focus`. La complexité est ici bien plus faible que pour la fonction softmax ($\mathcal{O}(K)$ par phrase), ce qui correspond à un gain considérable en termes de temps de calcul.

À la fin de l'algorithme, ce sont ces matrices qui donneront la représentation vectorielle des mots du vocabulaire. Ainsi, la ligne i de la matrice $W = \frac{W_e + W_s}{2}$ donnera la représentation du $i^{\text{ème}}$ mot du vocabulaire en dimension *dim*.

1. Pour éviter que les mots trop fréquents, souvent peu informatifs, soient sur-entraînés, on effectue un tout d'abord un sous-échantillonnage (*subsampling*).

2 Évaluation du modèle implémenté

Malgré l'utilisation généralisée des *word embeddings*, très peu de travaux théoriques expliquent ce qui est réellement capturé par ces représentations de mots. C'est pourquoi nous avons évalué la qualité des vecteurs-mots obtenus en sortie de notre modèle *word2vec* à l'aide de méthodes empiriques : la similarité cosinus entre deux mots, la réduction de dimension de l'espace des vecteurs-mots (ACP, T-SNE) et l'évaluation par jugement humain.

Grâce à l'évaluation de plusieurs modèles, nous avons pu déterminer les hyperparamètres les plus pertinents [2] au regard des données dont nous disposons (encadré 1). Selon nos critères d'évaluation, les résultats obtenus en sortie du modèle *word2vec* nous confirment la qualité de l'entraînement du corpus de 1,3 million de tweets [3] :

- Le coefficient de corrélation de Spearman avec le jugement humain est significatif (p-valeur proche de 0 %) et assez élevé (0,495).
- La recherche des plus proches voisins par similarité cosinus donne des résultats proches de l'intuition pour quelques mots de référence évalués (bonjour, femme, samedi...).
- La projection de certains vecteurs-mots sur les deux premiers axes d'une ACP (figure 2) est de qualité.
- Les sommes vectorielles sur les vecteurs-mots présentent des résultats plus mitigés. L'exemple de $\overrightarrow{Roi} - \overrightarrow{Homme} + \overrightarrow{Femme} = \overrightarrow{Reine}$ (figure 3) semble plutôt fonctionner.

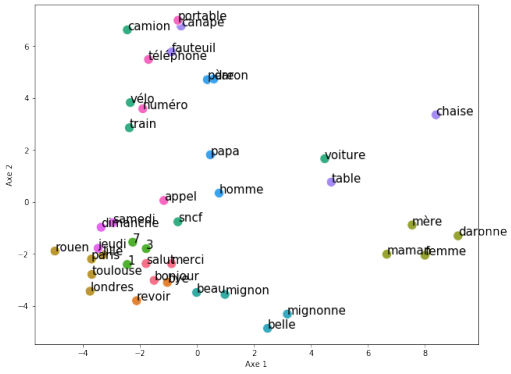


FIGURE 2 – ACP sur un corpus réduit de mots.

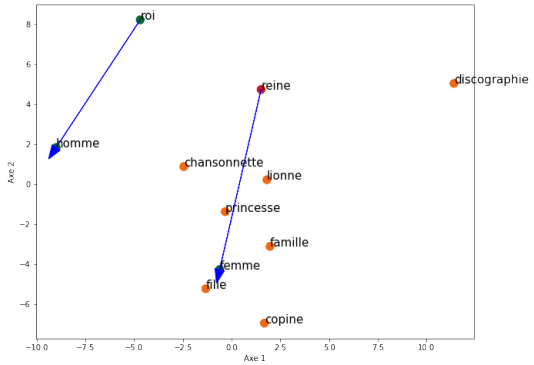


FIGURE 3 – $\overrightarrow{Roi} - \overrightarrow{Homme} + \overrightarrow{Femme} = ?$

Encadré 1 - Données utilisées

Tweets mensuels

Ces données correspondent à un ensemble de tweets postés en France entre 2011 et 2018, supposés être représentatifs de l'ensemble de tweets nationaux publiés durant cette période.
Pour entraîner notre modèle *word2vec*, nous avons mobilisé un premier échantillon de 1,3 million de tweets. Un second échantillon, composé de 4 200 tweets mensuels sur la période 2011-2018, a été utilisé pour construire notre indice mensuel de sentiment.

Tweets annotés sur les transports urbains

Cette base est composée d'un ensemble d'environ 23 000 tweets qui concernent les transports urbains (trains SNCF, métros et bus de la RATP) annotés en termes de sentiment (1 s'ils sont positifs, 0 s'ils sont négatifs).
Elle a été utilisée pour entraîner notre modèle d'analyse de sentiment.

Indicateur synthétique de confiance des ménages

L'indicateur synthétique de confiance des ménages provient de l'Enquête mensuelle de conjoncture auprès des ménages (Camme) de l'Insee.
C'est l'indicateur que nous cherchons à prévoir à partir de notre indice mensuel de sentiment.

2. $ep = 100$ pour le nombre d'« *epochs* »/ $lr = 0,02$ pour le « *learning rate* », ou taux d'apprentissage/ $w = 4$ pour la taille de la fenêtre (*window*) de sélection des mots contextes / $dim = 100$ pour la dimension des vecteurs-mots (ou *word-embeddings*).
3. Avant de nous attaquer au jeu de données complet, nous avons également évalué notre modèle sur un corpus fictif afin de nous assurer de sa robustesse et de sa validité.

3 Construction d'un indice mensuel de sentiment moyens des tweets

Nous utilisons ici vecteurs-mots en sortie du modèle *word2vec*, couplé avec une base de tweets qui concernent les transports urbains qualifiés de positifs ou négatifs, afin de créer un indice mensuel de sentiment moyens des tweet.

Nous cherchons tout d'abord à construire un modèle permettant de prédire le sentiment (0 ou 1) associé à un tweet à partir des mots qui le composent. Nous comparons les deux modélisations suivantes en évaluant leur efficacité. Pour cela, le corpus de la base de tweets sur les transports a été séparée en une base d'entraînement (environ 16 000 tweets) et une base de test (environ 7 000 tweets).

Modèle 1 : sentiment moyen des mots (référence)

Ce premier modèle de prédiction du sentiment utilise l'information des tweets labelisés pour déterminer un sentiment moyen par mot. Le sentiment prédit d'un tweet t composé de n mots sera :

$$S_{1,\gamma}(t) = \mathbb{1}\left\{\frac{1}{n}\sum_{i=1}^{n}\alpha_i \geq \gamma\right\} \qquad \in \{0,1\}$$

avec $\gamma \in [-1,1]$ un seuil fixé, $\alpha_i = \frac{nb_+(i)-nb_-(i)}{nb_+(i)+nb_-(i)} \in [-1,1]$ le sentiment moyen du mot i calculé à partir du nombre de tweets positifs ($nb_+(i)$) et négatifs ($nb_-(i)$) dans lesquels il apparaît.

➔ *Accuracy*[4] = 89,1 % ($\gamma^* = -0,14$).

Modèle 2 : *word-embeddings*

Ce deuxième modèle est une régression binaire avec comme prédicteurs chacune des 100 dimensions des vecteurs-mots. Les vecteurs-mots sont moyennés pour obtenir la représentation vectorielle du tweet : la « *sentence-embedding* ». Le sentiment prédit d'un tweet t sera :

$$S_{2,\gamma}(t) = \mathbb{1}\{\mathbb{P}(Y_i = 1|X_i) \geq \gamma\} \qquad \in \{0,1\}$$

Avec :

$$Y_i = \mathbb{1}\left\{\sum_{i=1}^{n}\beta_i X_{i,j} + \varepsilon_i \geq 0\right\} \quad \mathbb{P}(Y_i = 1|X_i) = F_\varepsilon\left(\sum_{i=1}^{n}\beta_i X_{i,j}\right)$$

- Y_i le sentiment du tweet i ;
- $X_{i,1}, \ldots, X_{i,n}$ des coordonnées de la *sentence-embedding* du tweet i ;
- ε_i le résidu de notre modèle de fonction de répartition $F_\varepsilon(x) = \frac{1}{1+e^{-x}}$ (logit).

➔ *Accuracy* = 69,8 % ($\gamma^* \simeq 0,5$).

À partir de ces modèles, nous avons construit deux indicateurs de sentiment des tweets en calculant la moyenne mensuelle des prévisions des sentiments des *sentence-embeddings* des tweets (graphique 4), puis avons comparé ces indicateurs avec l'indicateur synthétique de confiance des ménages (Camme, Insee).

Les tendances des deux indicateurs de sentiment diffèrent de celle de l'indicateur Camme. Toutefois, contrairement à l'indice du modèle 1, l'indice réalisé à partir du modèle 2 apporte une information significative pour prévoir l'évolution de l'indicateur Camme[5].

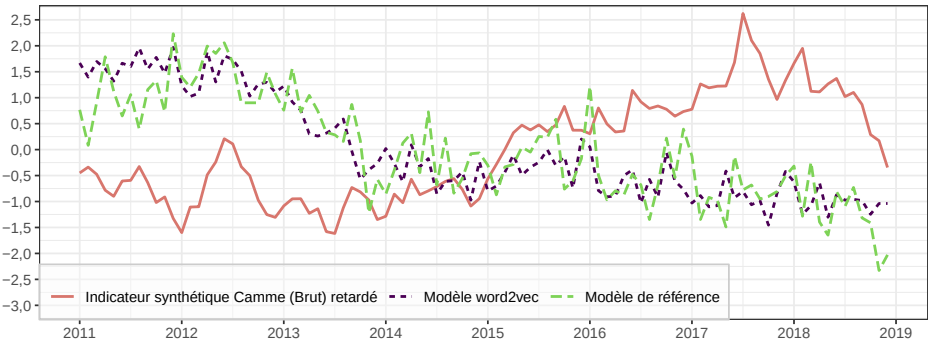


FIGURE 4 – Les trois indicateurs (centrés-réduits sur la période).

4. Taux de tweets dont le sentiment est bien prédit.
5. Il le cause au sens de Granger (p-valeur de 0,05).

Retrouvez l'ensemble du projet et son code sur https://github.com/ARKEnsae/TweetEmbedding