



Word-Embedding et sentiments des ménages avec Twitter

KIM ANTUNEZ, ROMAIN LESAUVAGE ET ALAIN
QUARTIER-LA-TENTE
11/06/2020
Ensaë — 2019-2020

Sommaire

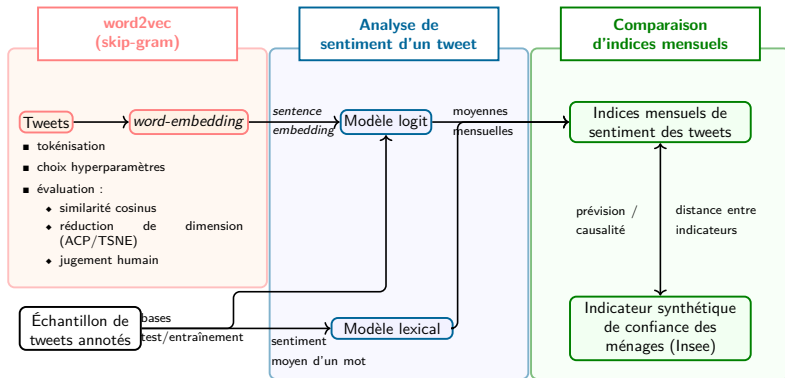
1. Introduction

2. *word2vec*

3. Évaluation du modèle

4. Indice de sentiments

Introduction



Sommaire

1. Introduction

2. *word2vec*

3. Évaluation du modèle

4. Indice de sentiments

Sommaire

1. Introduction

2. *word2vec*

3. Évaluation du modèle

3.1 Évaluation sur un corpus fictif

3.2 Choix des meilleurs hyperparamètres

3.3 Évaluation sur le corpus de tweets

4. Indice de sentiments

Comment évaluer le modèle ?

Les vecteurs-mots sont de grande dimension : comment juger de leur qualité et de leurs proximités ?

- **Similarité cosinus** : distance entre vecteurs-mots.
- **ACP et t-SNE** : réduire la dimension et analyser les proximités.
- **Jugement humain** : corrélations entre les proximités de nos vecteurs-mots et une base de proximités de mots construites par le jugement d'individus

Évaluation sur un corpus fictif

Idée : construire un corpus fictif pour lesquels on connaît le résultat attendu.

En pratique :

- On génère 10 groupes de mots composés d'un couple de référence et de 10 autres mots contexte.
- On construit 10 000 phrases en tirant au hasard :
 - 1 des groupes de mots ;
 - 1 des 2 mots « références » du groupe ;
 - 5 mots contextes ;
 - 3 mots bruits parmi une liste de 100 mots.
- On mélange les 9 mots de chaque phrase.

Résultats de l'évaluation

mot	similarité cosinus avec « grand »	mot	similarité cosinus avec « petit »
longueur	0,982	taille	0,987
petit	0,981	longueur	0,983
s	0,979	grand	0,981
⋮	⋮	⋮	⋮
susiens	-0,735	alesiez	-0,745
allates	-0,784	allates	-0,810

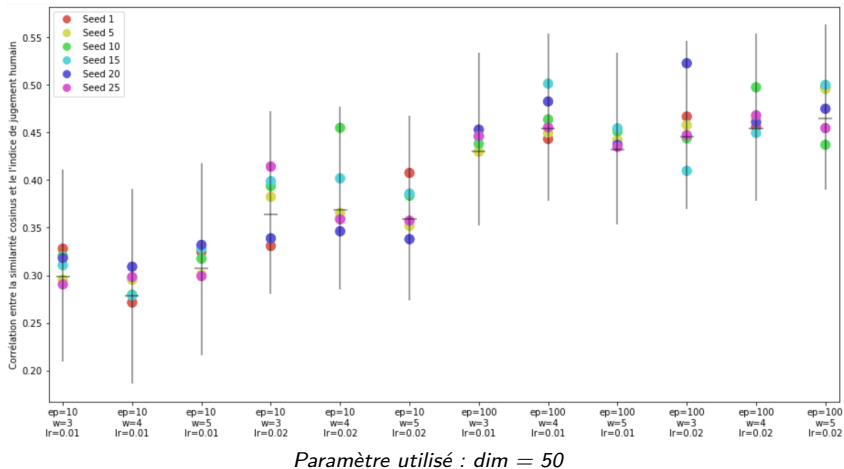
Paramètres utilisés : $ep = 50$ / $lr = 0,01$ / $w = 5$ / $dim = 10$.

➔ implémentation semble validée (résultats conformes aux attendus)

Déterminer les hyperparamètres

- *Word2vec* se base sur différents choix d'hyperparamètres :
 - o taille de la fenêtre (w)
 - o nombre d'epochs (ep)
 - o taux d'apprentissage (lr)
 - o dimension des *word-embeddings* (dim)
 - Détermination empirique des hyperparamètres :
 - o corrélation de Spearman entre nos vecteurs-mots et une base de jugement humain
 - o chronophage (il faut relancer le modèle à chaque fois).
- ➔ Utilisation complémentaire de Gensim puis validation avec notre implémentation.

Epochs, fenêtre et taux d'apprentissage



Valeurs des hyperparamètres retenus

- **Nombre d'epochs** : qualité des résultats croît avec le nombre d'epochs
➔ $ep = 100$.
- **Taille de fenêtre** : capte des informations sémantiques différentes selon sa valeur
➔ $w = 4$.
- **Taux d'apprentissage** : 0,02 donne de meilleurs résultats
➔ $lr = 0,02$.
- **Dimension** : qualité des résultats croît avec la dimension jusqu'à 300 puis décroît. Peu de différences entre 100 et 300.
➔ $dim = 100$.

Évaluation sur le corpus de tweets (1/2)

« Notre » modèle

Spearman : 0,57 (p-v : 4,1 %)

➡ **bons résultats**

bonjour (669)	femme (264)	1 (765)	samedi (203)
😊 (0,59)	quelle (0,49)	5 (0,55)	soir (0,57)
😊 (0,59)	cette (0,46)	mois (0,51)	vivement (0,51)
merci (0,54)	une (0,44)	10 (0,49)	demain (0,50)
nuit (0,48)	vie (0,44)	2 (0,48)	end (0,48)
bisous (0,47)	grippe (0,44)	top (0,48)	weekend (0,47)
bonne (0,47)	belle (0,43)	depuis (0,47)	matin (0,45)
😞 (0,46)	ma (0,43)	saison (0,46)	jeudi (0,45)
vous (0,46)	magnifique (0,43)	ans (0,44)	prochain (0,43)
plaisir (0,44)	nouvelle (0,43)	jours (0,43)	week (0,43)
allez (0,43)	vidéo (0,39)	3 (0,43)	🌸 (0,42)

$ep = 80 / w = 4 / lr = 0,02 / dim = 100 / base : 100\ 000\ tweets$

Évaluation sur le corpus de tweets (1/2)

« Notre » modèle

Spearman : 0,57 (p-v : 4,1 %)

➡ **bons résultats**

bonjour (669)	femme (264)	1 (765)	samedi (203)
😊 (0,59)	quelle (0,49)	5 (0,55)	soir (0,57)
😊 (0,59)	cette (0,46)	mois (0,51)	vivement (0,51)
merci (0,54)	une (0,44)	10 (0,49)	demain (0,50)
nuit (0,48)	vie (0,44)	2 (0,48)	end (0,48)
bisous (0,47)	grippe (0,44)	top (0,48)	weekend (0,47)
bonne (0,47)	belle (0,43)	depuis (0,47)	matin (0,45)
😞 (0,46)	ma (0,43)	saison (0,46)	jeudi (0,45)
vous (0,46)	magnifique (0,43)	ans (0,44)	prochain (0,43)
plaisir (0,44)	nouvelle (0,43)	jours (0,43)	week (0,43)
allez (0,43)	vidéo (0,39)	3 (0,43)	🌸 (0,42)

$ep = 80 / w = 4 / lr = 0,02 / dim = 100 / base : 100\ 000\ tweets$

Modèle Gensim

Spearman : 0,50 (p-v : 0,0 %)

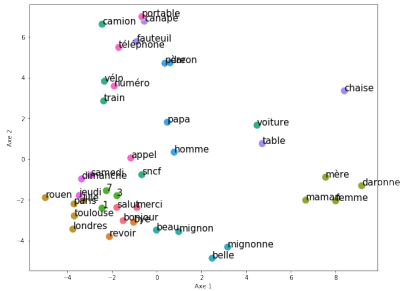
➡ **très bons résultats**

bonjour (17 043)	femme (6 177)	1 (21 055)	samedi (4 917)
bonsoir (0,85)	filles (0,86)	2 (0,65)	vendredi (0,88)
bjr (0,75)	copine (0,74)	3 (0,64)	jeudi (0,86)
hello (0,71)	meuf (0,71)	6 (0,63)	lundi (0,83)
salut (0,66)	demoiselle (0,66)	4 (0,62)	mercredi (0,83)
coucou (0,55)	nana (0,66)	7 (0,60)	dimanche (0,83)
transmets (0,49)	nièce (0,66)	5 (0,58)	mardi (0,76)
désagrement (0,48)	sœur (0,65)	9 (0,58)	demain (0,72)
avezvous (0,48)	barbe (0,65)	8 (0,56)	barathon (0,56)
bettembourg (0,48)	maman (0,64)	1e (0,55)	22h45 (0,55)
hey (0,47)	princesse (0,64)	34 (0,53)	20h (0,54)

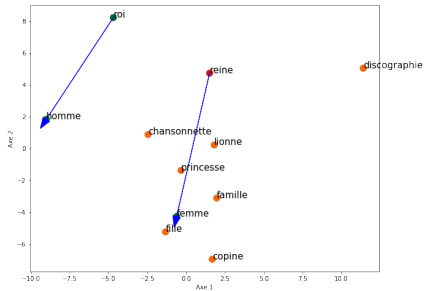
$ep = 100 / w = 4 / lr = 0,02 / dim = 100 / base : ensemble\ des\ tweets$

10 plus proches voisins par similarité cosinus

Évaluation sur le corpus de tweets (2/2)



ACP sur un corpus réduit de mots.



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

➡ Réduction de dimension des vecteurs-mots et (parfois) opérations sur les mots **convaincants**

Sommaire

1. Introduction

2. *word2vec*

3. Évaluation du modèle

4. Indice de sentiments

4.1 Prédire le sentiment d'un tweet

Prédire le sentiment d'un tweet

- **Idée** : associer à chaque tweet un sentiment :
 - o 1 s'il est positif
 - o 0 s'il est négatif.
- base de 23 000 tweets annotés sur les transports urbains
 - o **base d'entraînement** : 16 000 tweets
 - o **base de test** : 7 000 tweets
- 2 approches :
 - o **Modèle lexical** : utiliser l'information des tweets labélisés pour construire un sentiment moyen par mot.
 - o **Modèle logit** : utiliser les *word-embeddings* comme prédicteurs d'une régression logistique.

Modèle lexical : sentiment moyen des mots

Le sentiment prédit d'un tweet t composé de n mots sera :

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

- $\gamma \in [-1, 1]$ un seuil fixé ;
 - $\alpha_i = \frac{nb_+(i) - nb_-(i)}{nb_+(i) + nb_-(i)} \in [-1, 1]$ sentiment moyen du mot i calculé à partir du nombre de tweets positifs ($nb_+(i)$) et négatifs ($nb_-(i)$) dans lesquels il apparaît.
- ➡ $Accuracy^1 = 89,1 \%$ ($\gamma^* = -0,14$).

1. Taux de tweets dont le sentiment est bien prédit.

Modèle logit : prédiction grâce aux *word-embeddings*

Le sentiment prédit d'un tweet t sera :

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

Avec :

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

- Y_i le sentiment du tweet i ;
- $X_{i,1}, \dots, X_{i,n}$ les coordonnées de la *sentence-embedding* du tweet i ;
- ε_i le résidu de notre modèle, de fonction de répartition F_ε qui vaudra $F_\varepsilon(x) = \frac{1}{1+e^{-x}}$ dans le cas d'un modèle logit et $F_\varepsilon(x) = \Phi(x)$ (fonction de répartition d'une loi $\mathcal{N}(0,1)$) dans le cas d'un modèle probit.

Spécifications du modèle logit

Plusieurs points à traiter :

- Doit-on inclure les *stop-words*? OUI
- Comment traiter les mots inconnus? AFFECTER LE VECTEUR-MOT LOWFREQUENCY
- Modèle probit ou logit? LOGIT

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

Limites des modèles utilisés

Modèle lexical ici meilleur que le modèle logit car ...

- 1 Davantage de mots inconnus dans le modèle logit (36,2 % du vocabulaire contre 13,2 % dans le modèle lexical)

Limites des modèles utilisés

Modèle lexical ici meilleur que le modèle logit car ...

- 1 Davantage de mots inconnus dans le modèle logit (36,2 % du vocabulaire contre 13,2 % dans le modèle lexical)
- 2 Le processus d'annotation utilisé pour les tweets sur les transports urbains reproduit en partie par le modèle lexical (?)

Limites des modèles utilisés

Modèle lexical ici meilleur que le modèle logit car ...

- 1 Davantage de mots inconnus dans le modèle logit (36,2 % du vocabulaire contre 13,2 % dans le modèle lexical)
- 2 Le processus d'annotation utilisé pour les tweets sur les transports urbains reproduit en partie par le modèle lexical (?)
- 3 Le *domain shift*

Limites des modèles utilisés

Modèle lexical ici meilleur que le modèle logit car ...

- 1 Davantage de mots inconnus dans le modèle logit (36,2 % du vocabulaire contre 13,2 % dans le modèle lexical)
- 2 Le processus d'annotation utilisé pour les tweets sur les transports urbains reproduit en partie par le modèle lexical (?)
- 3 Le *domain shift*

➡ Utilisation d'une nouvelle base de test pour neutraliser certains de ces effets

Modèle logit alors meilleur que le modèle lexical
(*Accuracy* de 61,9 % contre 55,9 %).

Merci pour votre attention

 ARKEnsaë/TweetEmbedding

 Rapport du projet

