Analyse de sentiment

Frédéric Wantiez et Pierre Vigier

Supélec - Projet de synthèse

15 juin 2016

Introduction

Définition

L'analyse de sentiments consiste à dire si un texte dégage un sentiment positif ou négatif.

Applications

Les principales applications de l'analyse de sentiments sont :

- en marketing : repérer les promoteurs et les détracteurs ;
- en finance : miner les réseaux sociaux et les blogs pour prédire les mouvements de marché et la popularité d'un produit.

Plan

Notations

Nous noterons :

- $V = \{w_1, ..., w_D\}$: l'ensemble de tous les mots ;
- $V^* = \cup_{n \geq 0} V^n$: l'ensemble de tous les textes sur le vocabulaire V ;
- $x_1, ..., x_N \in V^*$: les textes de notre ensemble de de données ;
- $y_1, ..., y_N \in \{0, 1\}$: les sentiments associés à chaque texte.

Formalisation

Notre objectif est de déterminer une fonction f telle que :

$$\forall i \in 1, ..., N, y_i \approx \hat{y}_i = f(x_i)$$

Mesure de performance

Nous allons utiliser la précision comme mesure de performance, nous la définissions telle que :

$$A(y_1, ..., y_N, \hat{y}_1, ..., \hat{y}_N) = \frac{\sum_{i=0}^{N} 1_{y_i = \hat{y}_i}}{N}$$

Plan

Exemple d'avis négatif

Story of a man who has unnatural feelings for a pig. Starts out with a opening scene that is a terrific example of absurd comedy. A formal orchestra audience is turned into an insane, violent mob by the crazy chantings of it's singers. Unfortunately it stays absurd the WHOLE time with no general narrative eventually making it just too off putting. Even those from the era should be turned off. The cryptic dialogue would make Shakespeare seem easy to a third grader. On a technical level it's better than you might think with some good cinematography by future great Vilmos Zsigmond. Future stars Sally Kirkland and Frederic Forrest can be seen briefly.

Exemple d'avis positif

```
The plot had some wretched, unbelievable twists.
However, the chemistry between Mel Brooks and Leslie
Ann Warren was excellent. The insight that she comes
to, ""There are just moments,"" provides a
philosophical handle by which anyone could pick up,
and embrace, life.<br /><br />That was one of several
moments that were wonderfully memorable.
```

Plan

Définitions

Nombre d'occurrences d'un mot

Pour un texte $t \in V^*$, posons :

$$\forall i \in \{1, ..., D\}, tf_{i,t} = card(\{j, t_j = w_i\})$$

Sac de mots

Le sac de mot d'un texte t est un vecteur b de $\mathbb{R}^D$ tel que :

$$\forall i \in 1, ..., D, b_i = tf_{i,t}$$

Exemple

$$S_1 = \text{"Bob aime les films d'action."}$$
$$S_2 = \text{"Alice préfère les films d'amour."}$$

Le vocabulaire commun aux deux phrases est :

$$V = \{\text{Bob}, \text{aime}, \text{les}, \text{films}, \text{d}, \text{action}, \text{Alice}, \text{préfère}, \text{amour}\}$$

Sacs de mots associés à S_1 et S_2 :

$$b_1 = (1, 1, 1, 1, 1, 1, 0, 0, 0)^T$$
$$b_2 = (0, 0, 1, 1, 1, 0, 1, 1, 1)^T$$

TF-IDF

Fréquence inverse de document

La fréquence inverse de document du mot w_i est définie par :

$$idf_i = log(\frac{N}{N_i})$$

Sac de mots pondéré

Le sac de mot pondéré d'un texte t est un vecteur b de $\mathbb{R}^D$ tel que :

$$\forall i \in 1, ..., D, b_i = tf_{i,t} \times idf_i$$

Résultats

Forêt aléatoire à 100 estimateurs + BOW	0.84356
Forêt aléatoire à 100 estimateurs + BOW + TF-IDF	0.83952
Régression logistique + BOW	0.84748
Régression logistique + BOW + TF-IDF	0.88308

FIGURE – Précisions des sacs de mots avec différents algorithmes (BOW : sacs de mots, TF-IDF : sacs de mots pondérés par IDF)

Régression logistique

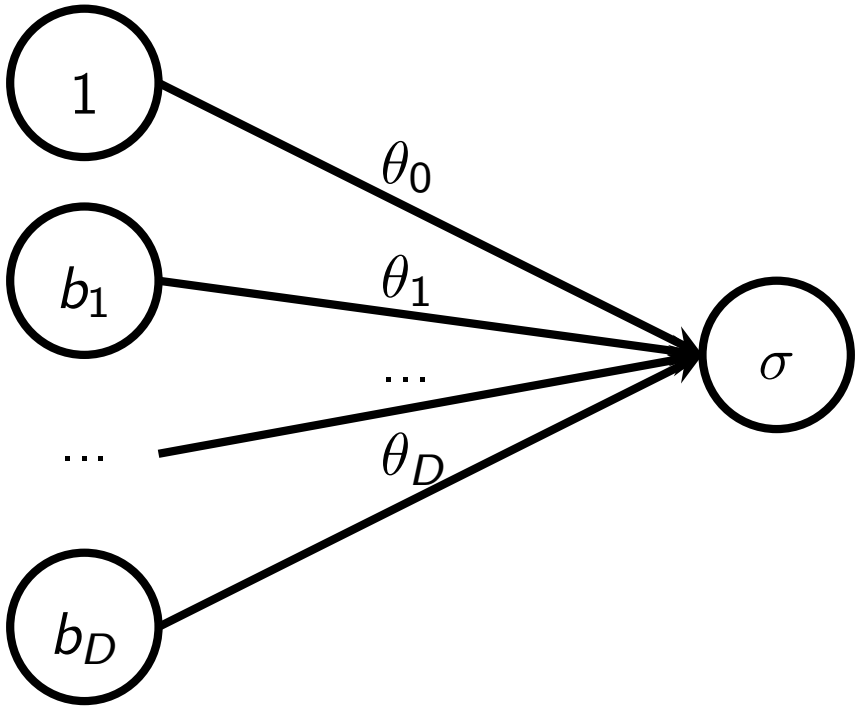


FIGURE – Représentation d'une régression logistique sous forme de réseau.

Mots négatifs

NEGATIVE

1. worst (-9.207327806842622)
2. bad (-7.259958398881469)
3. awful (-6.415897979257145)
4. waste (-6.330645081158189)
5. boring (-5.972322478351235)
6. poor (-5.290181559864669)
7. terrible (-4.680185463018839)
8. worse (-4.432747319290786)
9. nothing (-4.411951085231576)
10. dull (-4.377273111550886)

Mots positifs

POSITIVE
1. great (6.731768881148247)
2. excellent (6.010829696751981)
3. perfect (4.944364766157447)
4. best (4.721911978651118)
5. wonderful (4.516264655915644)
6. amazing (4.08891536519038)
7. today (3.6921102148583946)
8. favorite (3.658223120278884)
9. loved (3.589046275536792)
10. well (3.542215560495744)

Plan

Modèle Skip-gram

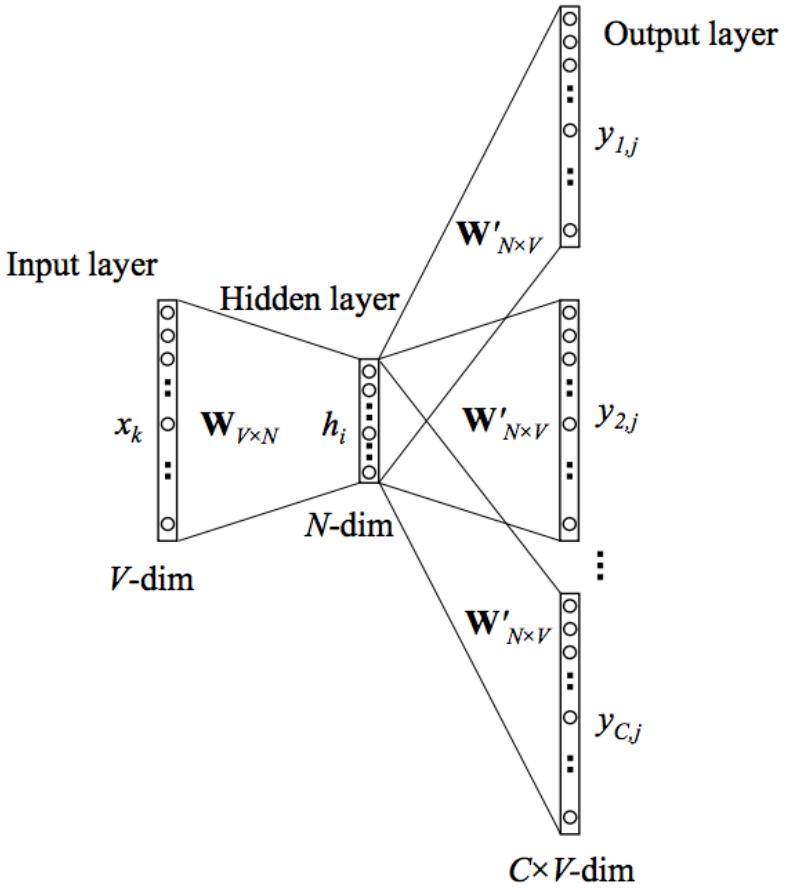
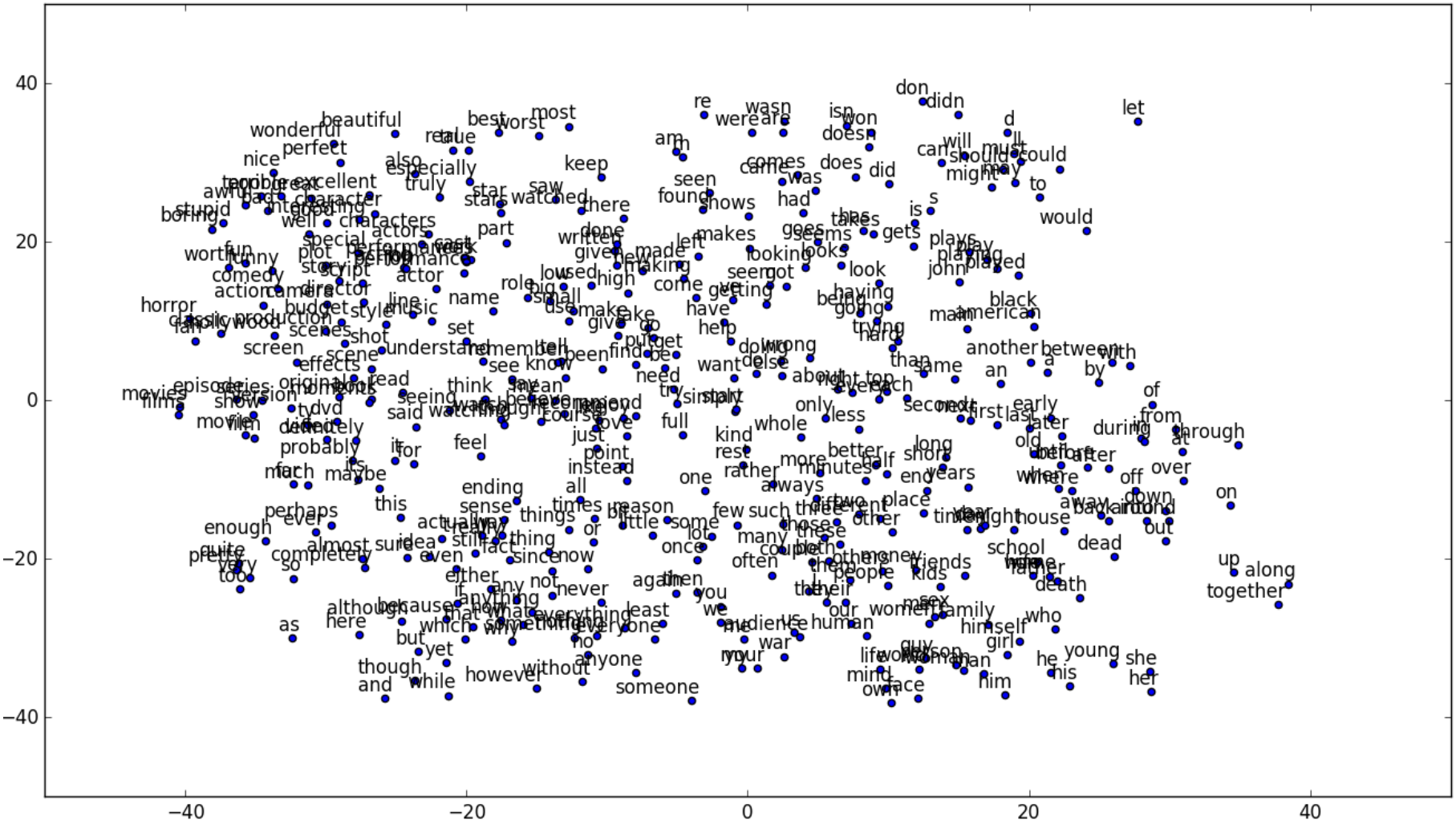
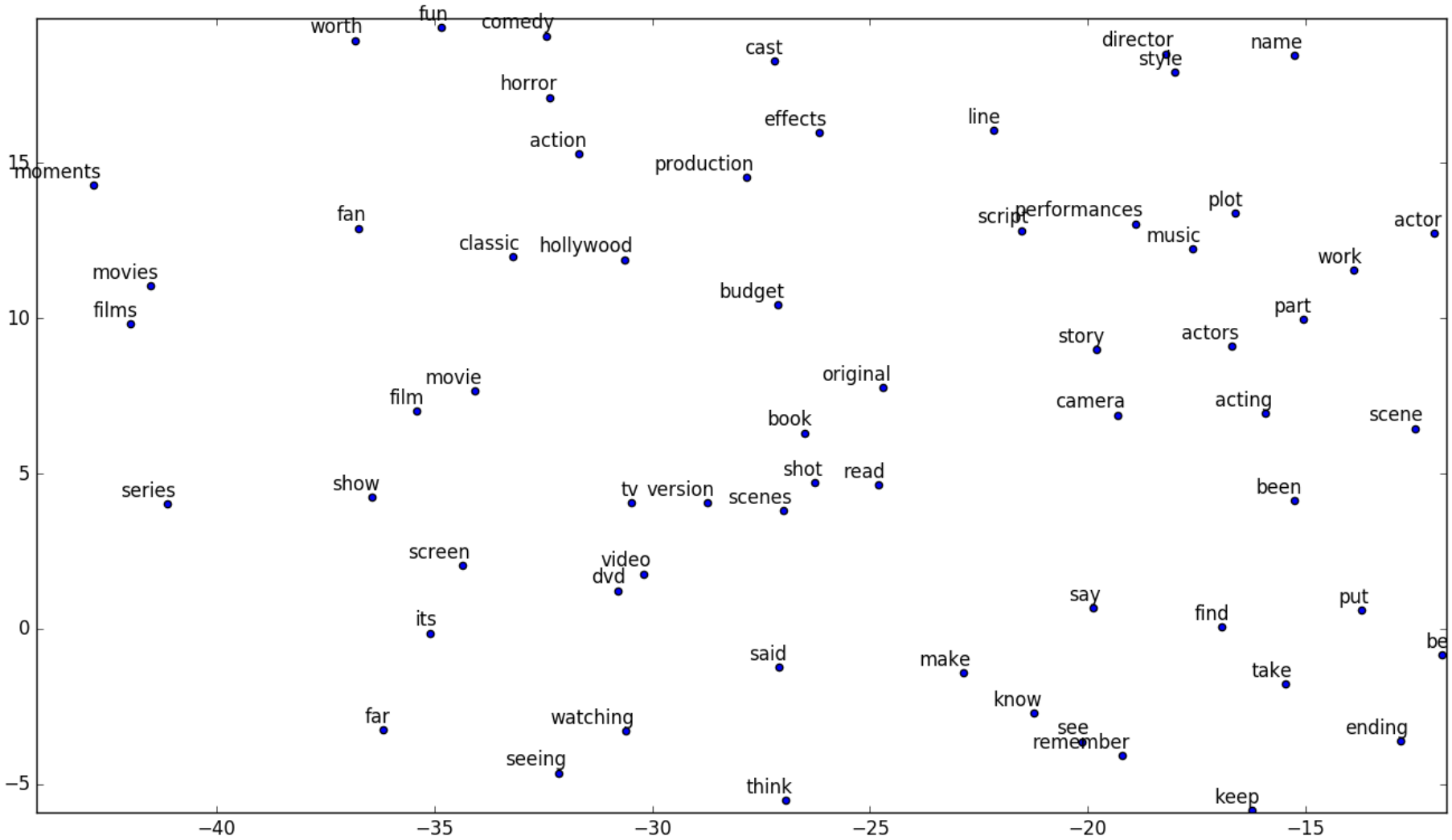


FIGURE – Réseau de neurones du modèle Skip-gram (source : Wikimedia).

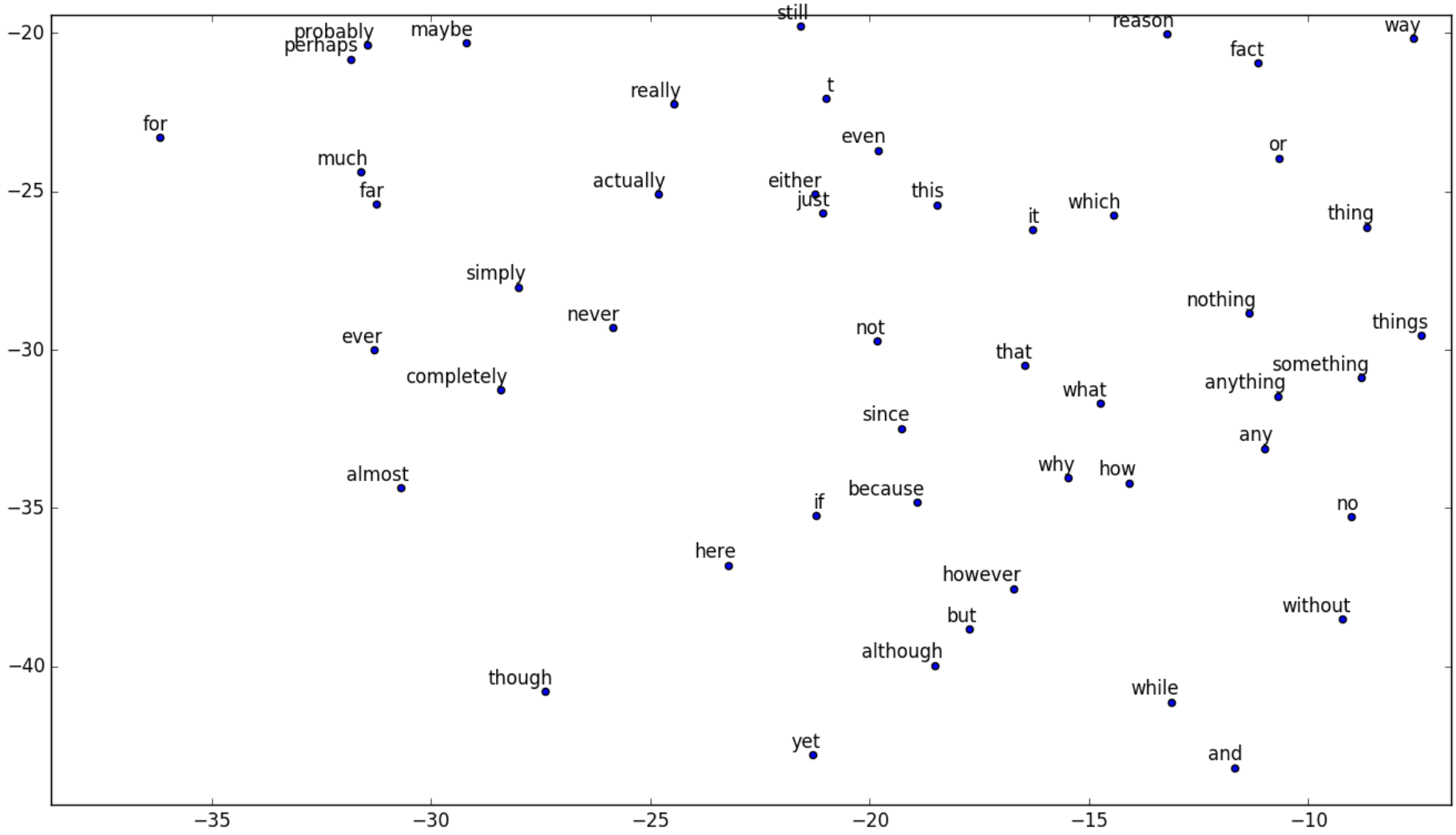
Visualisation t-SNE



Cluster de mots liés au cinéma



Cluster de mots-outils



Vecteur d'un texte

Moyenne de vecteurs de mots

Le vecteur représentant un texte t est le vecteur :

$$h_{vec}(t) = \frac{\sum_{w_i \in t} tf_{i,t} v_i}{\sum_{w_i \in t} tf_{i,t}}$$

Moyenne pondérées de vecteurs de mots

Le vecteur représentant un texte t est le vecteur :

$$h_{vec+idf}(t) = \frac{\sum_{w_i \in t} tf_{i,t} idf_i v_i}{\sum_{w_i \in t} tf_{i,t} idf_i}$$

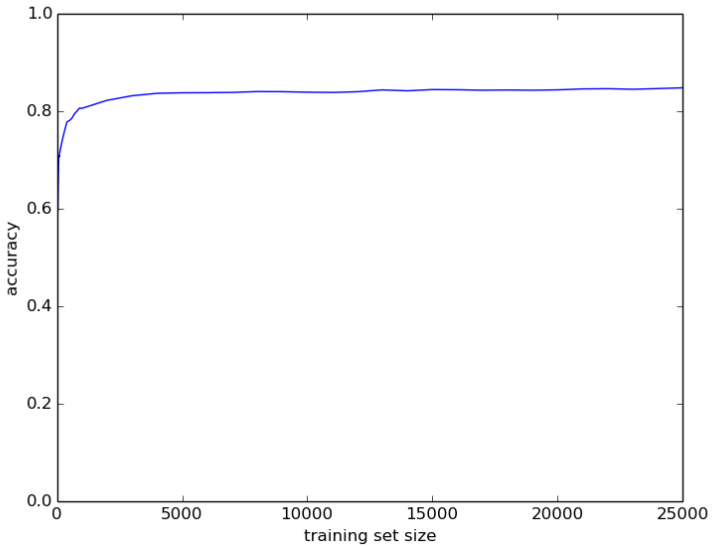
Résultats

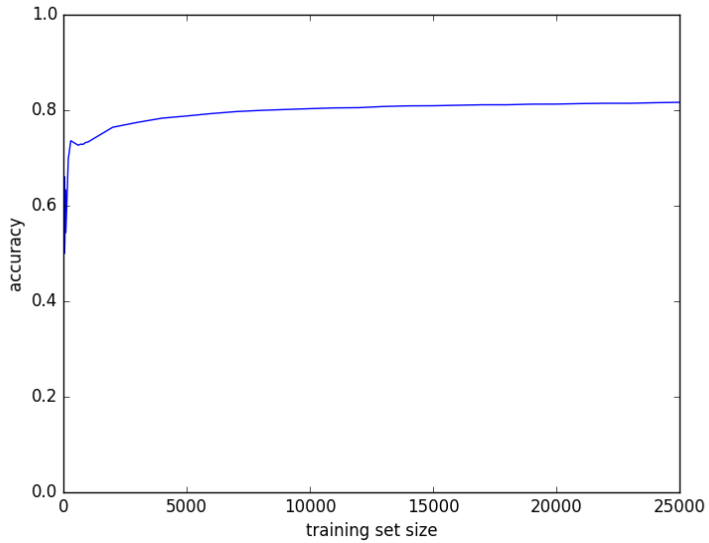
Forêt aléatoire à 100 estimateurs + Vec	0.7954
Forêt aléatoire à 100 estimateurs + Vec + TF-IDF	0.79416
Régression logistique + Vec	0.81652
Régression logistique + Vec + TF-IDF	0.81688

FIGURE – Précisions des vecteurs de mot avec différents algorithmes (Vec : vecteurs de mots, TF-IDF : vecteurs de mot pondérés par IDF)

Courbes d'apprentissage

Courbe d'apprentissage pour les sacs de mots (à gauche) et les vecteurs de mot (à droite) :





Plan

N-gramme

Les n-grammes, $n \in \mathbb{N}$, d'un texte t sont les n-uplets de mots consécutifs de t.

Exemple

Considérons la phrase $t =$ "Bob aime les films d'action.", les bigrammes ou 2-grammes de t sont :

$$\{(\text{Bob}, \text{aime}), (\text{aime}, \text{les}), (\text{les}, \text{films}), (\text{films}, \text{d}), (\text{d}, \text{action})\}$$

Bigrammes les plus courants

```
big_list.most_common(10)
[('ever seen', 1319),
 ('special effects', 1114),
 ('even though', 1043),
 ('one best', 919),
 ('low budget', 880),
 ('year old', 878),
 ('looks like', 838),
 ('waste time', 789),
 ('see movie', 772),
 ('much better', 746)]
```

C	n	TF-IDF	Précision
5000	1	Non	0.85164
10000	1	Non	0.85432
15000	1	Non	0.85664
5000	2	Non	0.8582
10000	2	Non	0.86844
15000	2	Non	0.87544
5000	1	Oui	0.88308
10000	1	Oui	0.88364
15000	1	Oui	0.88412
5000	2	Oui	0.88848
10000	2	Oui	0.89312
15000	2	Oui	0.89432
20000	2	Oui	0.8954

FIGURE – Précisions des sacs de n-grammes en entrée d'une régression logistique.

Plan

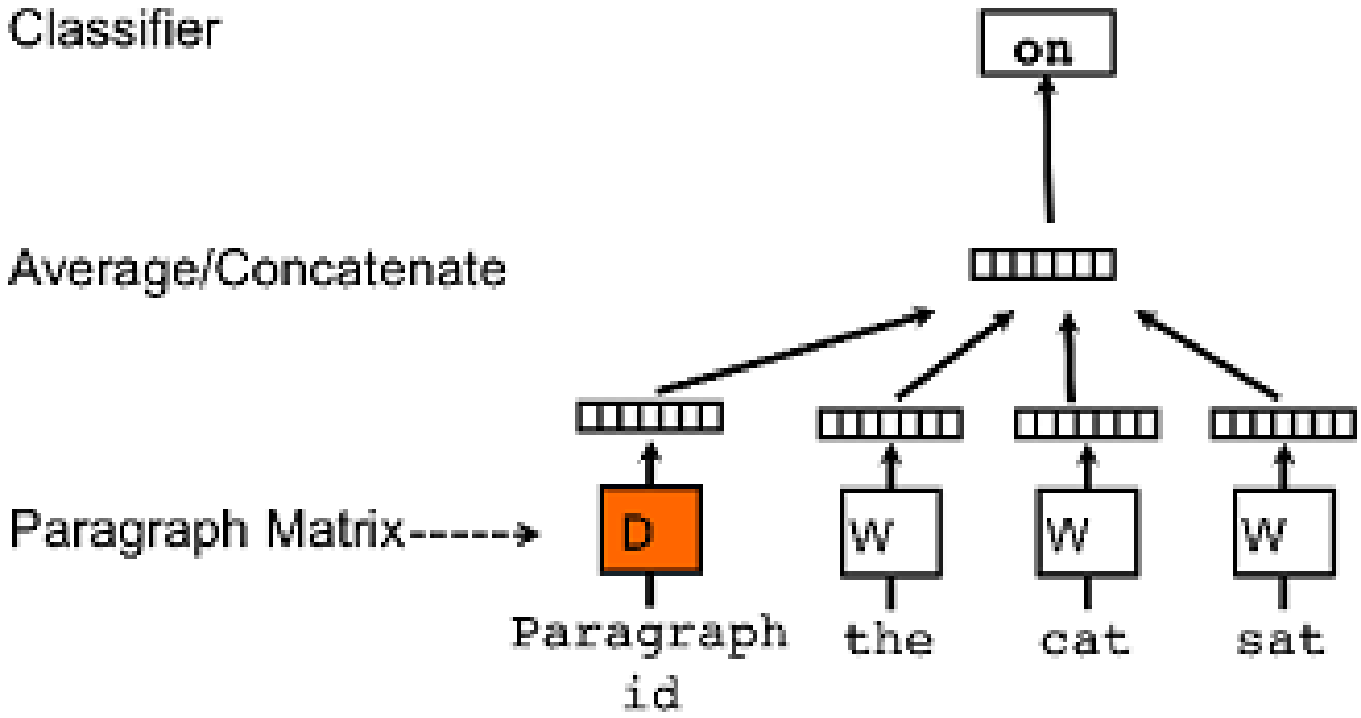


FIGURE – Framework d'apprentissage des paragraph vectors (source : Mikolov et al).

Analyse sémantique du modèle

```
In[1]: model.most_similar('interesting')
Out[1]:
('enjoyable', 0.549770712852478),
 ('entertaining', 0.5360784530639648),
 ('important', 0.5295416712760925),
 ('intriguing', 0.4986931085586548),
 ('amazing', 0.4958794713020324 7),
 ('exciting', 0.4942770004272461),
 ('excellent', 0.4942253828048706),
 ('awesome', 0.46036389470100403),
 ('amusing', 0.453685730695724 5),
 ('impressive', 0.44812250137329 1)
```

Régression linéaire + PV-DM (concat.)	0.883
Régression linéaire + PV-DM (mean)	0.823

TABLE – Précision des paragraph vectors avec les deux variantes utilisées pour des vecteurs de dimension 300

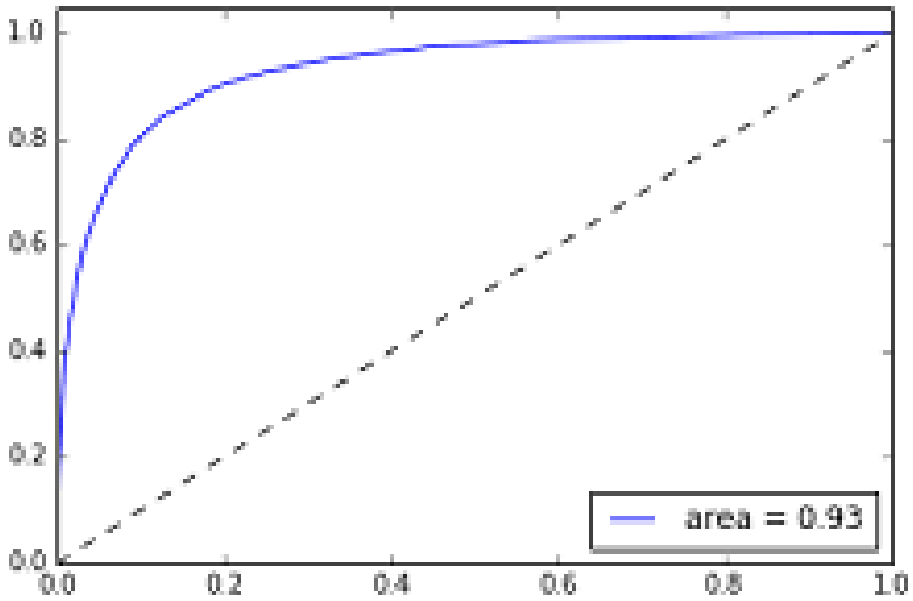


FIGURE – Courbe ROC pour le modèle PC-DM avec concaténation.

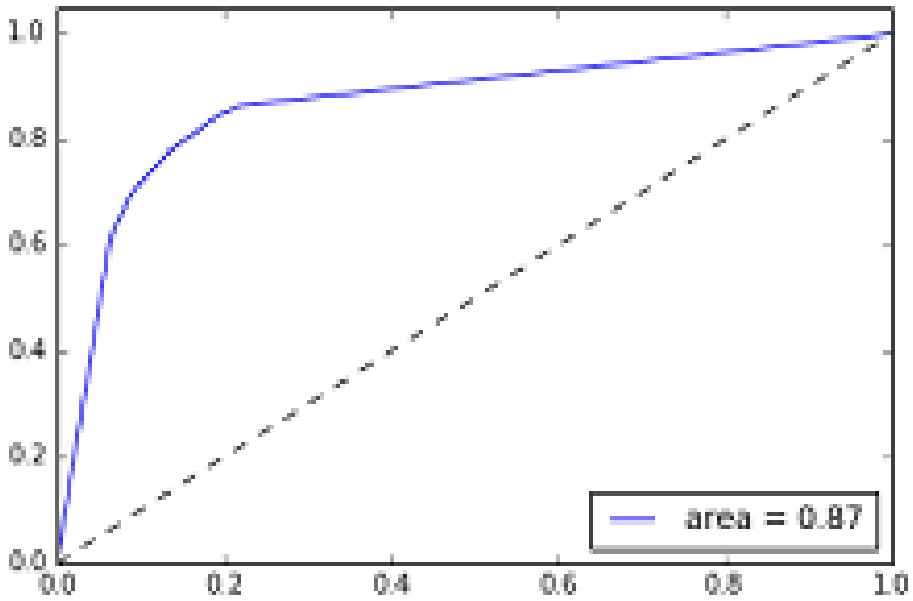


FIGURE – Courbe ROC pour le modèle PC-DM avec moyenne.

Plan

Finalement, nous avons concaténé les vecteurs de paragraphes et les vecteurs obtenus à partir des n-grammes. Nous obtenons une précision de 0.90116.

Conclusion

Merci !