Analyse de sentiment par réseaux neuronaux réccurents

Thomas Moreau & Bertrand Rondepierre

Télécom paristech - MDI343

13 mai 2014

Overview

Les Objectifs

- Catégoriser l'opinion générale exprimée par une phrase
- Plusieurs niveaux :
  - Binaire : positif/négatif
  - Fine : Très négatif, négatif, neutre, positif, très positif
- Représenter la phrase dans un espace propre qui permet de mettre en lumière l'opinion qu'elle contient
- Génération de phrases ?

Le Stanford Tree bank

- 11 855 critiques de film issues de Rotten tomatoes
- Structure de la phrase sous forme d'arbre et labels au niveau noeud
  $\Rightarrow$ Possibilité d'analyse fine à chaque noeuds

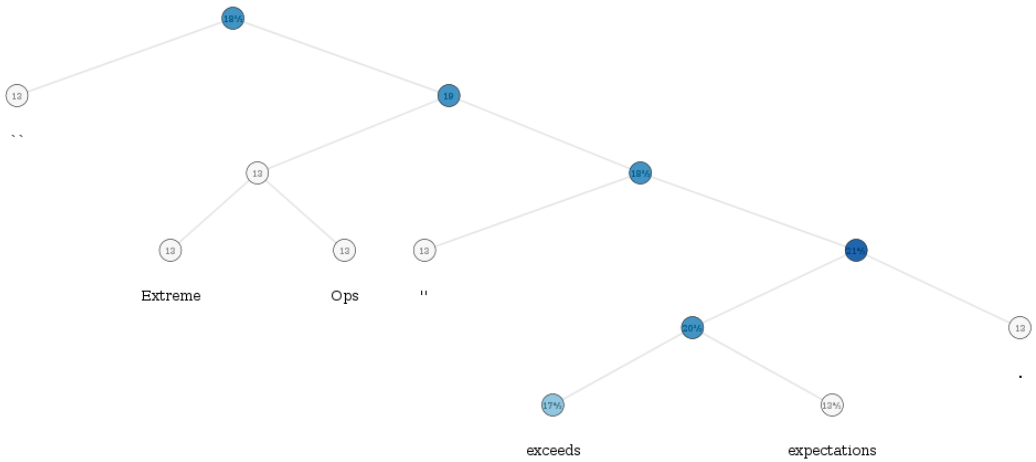


FIGURE : "Extreme Ops" exceeds expectations.

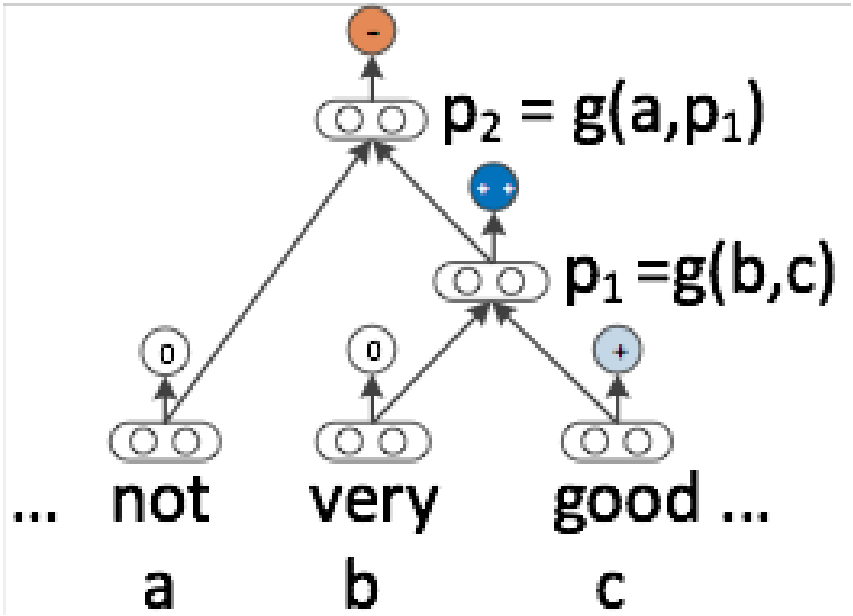
Le modèle

- **Classique** : Sac de mots - bag of word
- Ne tient pas compte de la structure de la phrase, seulement du nombre d'occurrences.
- Pourtant besoin de prendre en compte structure subtiles/fines : négation, expression etc...
  This movie was actually neither that funny, nor super witty.
- Idée : utiliser la structure d'arbre pour prédire le sentiment à tous les niveaux de l'arbre, puis au niveau de la phrase complète.
- On cherche aussi a optimiser la représentation des mots et phrases dans l'espace englobant.

Réseau récurrent

Paramètres :

1. Dictionnaire des mots $L \in \mathbb{R}^{|V| \times d}$
2. $V \in \mathbb{R}^{d \times 2d \times 2d}$ le tenseur
3. $W \in \mathbb{R}^{d \times 2d}$ opérateur linéaire
4. $W_s \in \mathbb{R}^{C \times d}$ softmax à chaque nœud où on fait une prédiction.



Réseau récurrent

Forward pass (données $\rightarrow$ représentations + labels) :

1. L donne les représentation aux feuilles = mots
2. Pour deux enfants i_1, i_2 du nœud i :

$$x^i = f\left(\begin{bmatrix} x^{i_1} \\ x^{i_2} \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} x^{i_1} \\ x^{i_2} \end{bmatrix} + W \begin{bmatrix} x^{i_1} \\ x^{i_2} \end{bmatrix}\right)$$

   Où f est la fonction d'activation du réseau : $f = \tanh$ ici.

3. A chaque noeud où on dispose de la représentation x^i, softmax : $y^i = \sigma(W_s x^i)$ où :

$$\sigma_j(v) = \frac{\exp(v_j)}{\sum_{k=1}^{C} \exp(v_k)}$$

Fonction à optimiser

Pour une distribution de proba cible t^i au niveau du nœud i

- $E = \sum_{phrase} \sum_{nodes} D_{KL}(t^i || y^i) = \sum_{phrase} \sum_{nodes} \sum_{k} t^i_k \log y^i_k + C.$

- Socher construit la distribution cible avec du one-hot encoding, ie binarise les 5 labels TN,N,Neutre,P,TP
  $\rightarrow$ Pénalise de la même façon une erreur positif/très positif que très négatif/très positif.

- Notre approche :
  - Approcher (KL) la probabilité $P[S = 1]$ pour $S \in \{-1, 1\}$ représentant le sentiment.
  - Construire la probabilité cible avec chaque sentiment représentant un niveau sur l'échelle $[0, 1]$ (e.g neutre $= 0.5$)

Apprentissage

- Rétropropagation des erreurs $\frac{\partial E^i}{\partial \theta} = (y^i - t^i) W_s \frac{\partial x^i}{\partial \theta}$:

$$\frac{\partial x^i}{\partial \theta} = F(x^i_j) \left(2 \begin{bmatrix} x^{i_g} \\ x^{i_d} \end{bmatrix}^T V^{[j]} + W_{j:} \right) \frac{\partial}{\partial \theta} \begin{bmatrix} x^{i_g} \\ x^{i_d} \end{bmatrix} + \begin{bmatrix} x^{i_g} \\ x^{i_d} \end{bmatrix}^T \frac{\partial V^{[j]}}{\partial \theta} \begin{bmatrix} x^{i_g} \\ x^{i_d} \end{bmatrix} + \frac{\partial W_{j:}}{\partial \theta} \begin{bmatrix} x^{i_g} \\ x^{i_d} \end{bmatrix}$$

  Pour le noeud i avec F telle que $F(f(x)) = f'(x)$

- Stratégie d'optimisation : AdaGrad
  $\Rightarrow$ Principe : descente de gradient avec learning rate obtenu en diviser par la sommes des carrés des gradients déjà calculés. Favorise les features rare.
- Autre solution : RPROP (resilient backpropagation) :
  $\Rightarrow$ Augmentation/diminution multiplicative du learning rate en fonction de la dynamique du gradient.

Paramètre

- Learning rate, mini batch size, regularisation

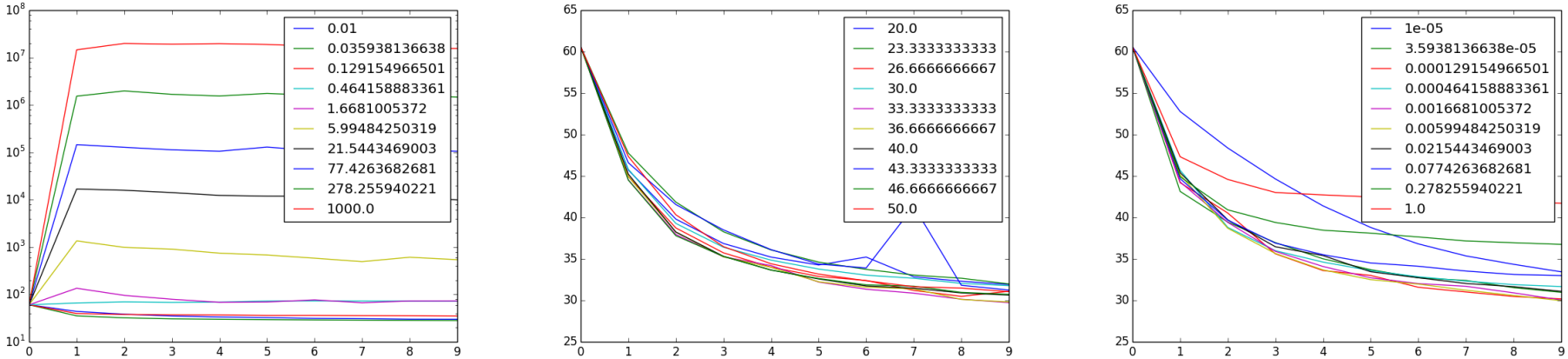


FIGURE : Courbe d'apprentisssage en Cross validation pour le learning rate, la taille du mini batch et le facteur de régularisation

Apprentissage

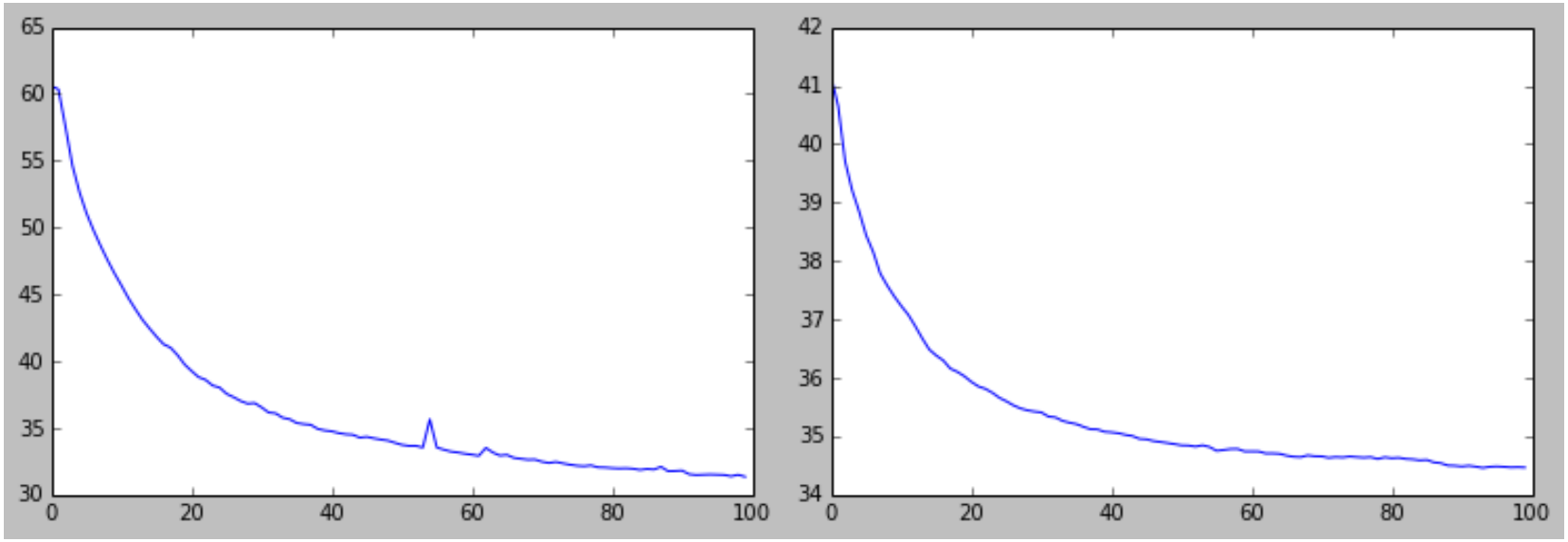


FIGURE : Courbe d'apprentisssage pour AdaGrad *(gauche)* et Rprop *(droite)*

Expérimentalement AdaGrad atteint un minimum local plus rapidement que Rprop qui se bloque beaucoup plus tôt.

Resultat de Classification

	Fine		Binaire	
	All node	Root	All nodes	Root
Socher	80.7	45.7	87.6	85.4
Notre modèle	79.7	42.2	86.6	90.6

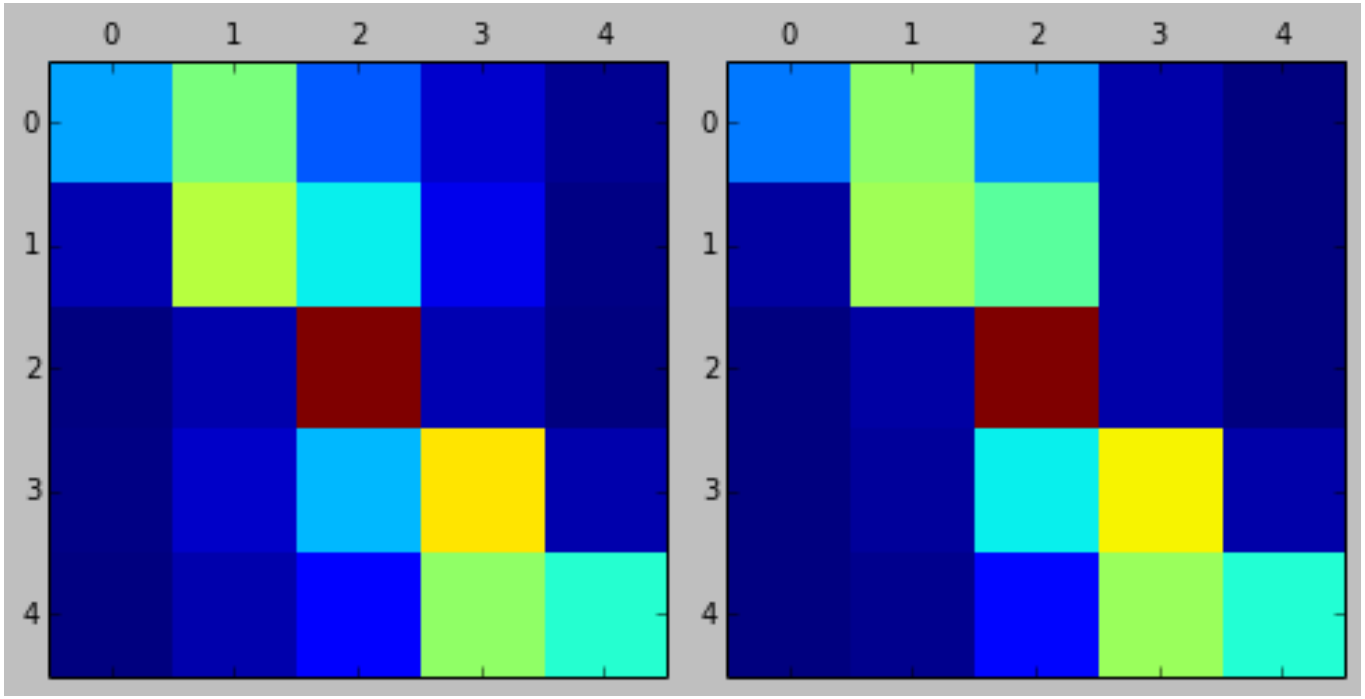


FIGURE : Matrice de confusion Noeuds : Socher (gauche) nous (droite)

Resultat de Classification

	Fine		Binaire	
	All node	Root	All nodes	Root
Socher	80.7	45.7	87.6	85.4
Notre modèle	79.7	42.2	86.6	90.6

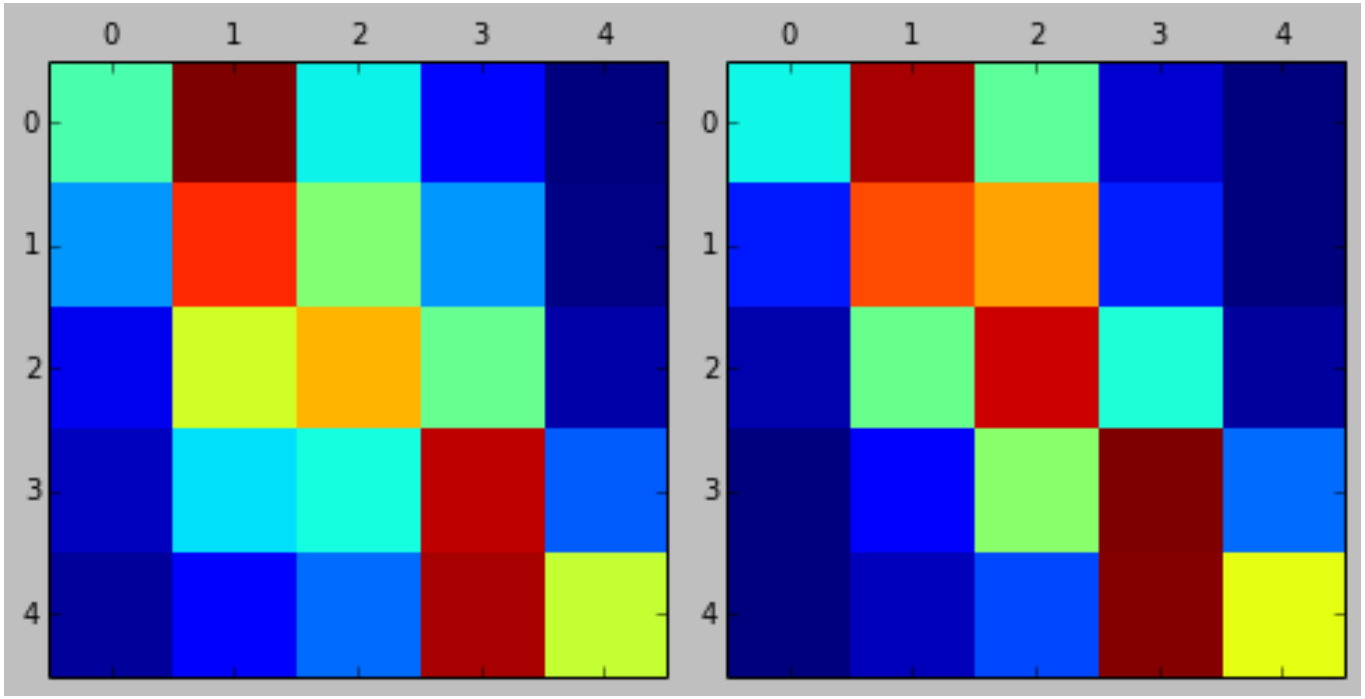


FIGURE : Matrice de confusion racines : Socher (gauche) nous (droite)

Résultat de Classification

On représente pour notre modèle la précision en fonction de ε, ie le pourcentage de classification correcte avec des intervalles $|y^i - t^i| < \varepsilon$

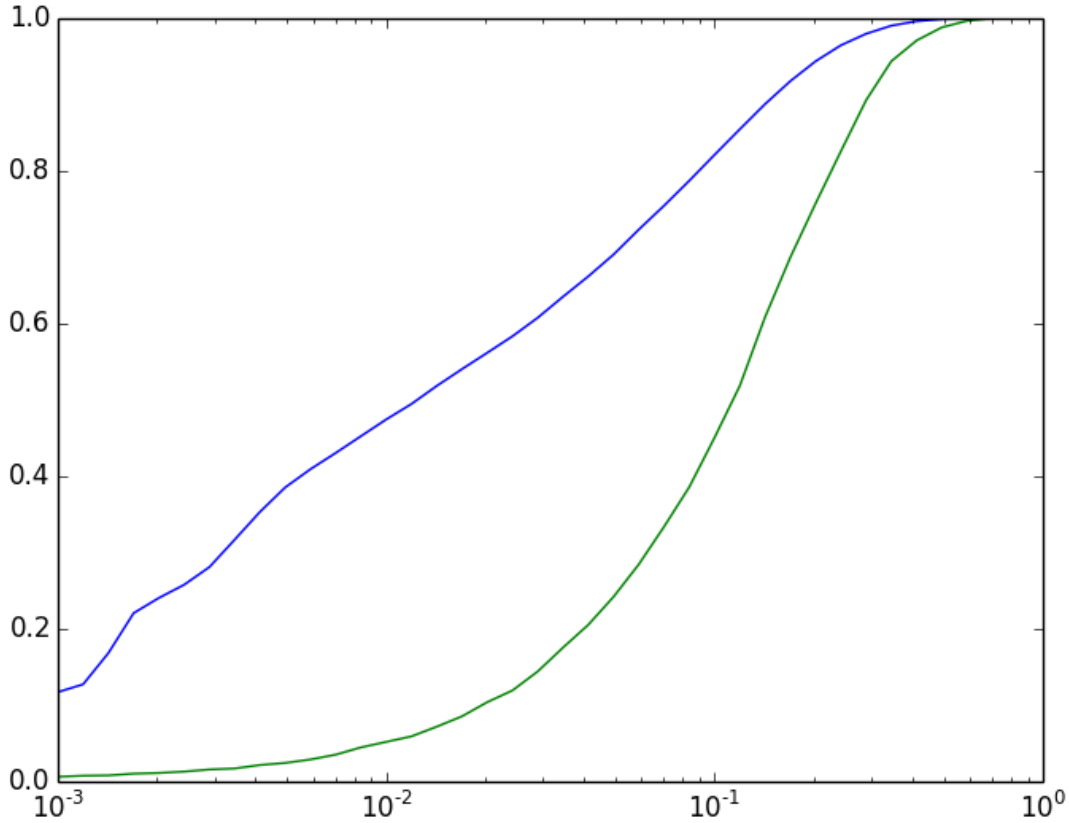


FIGURE : Precision de notre regression (log scale)

Représentations apprises - Mots

En représentant en 2D

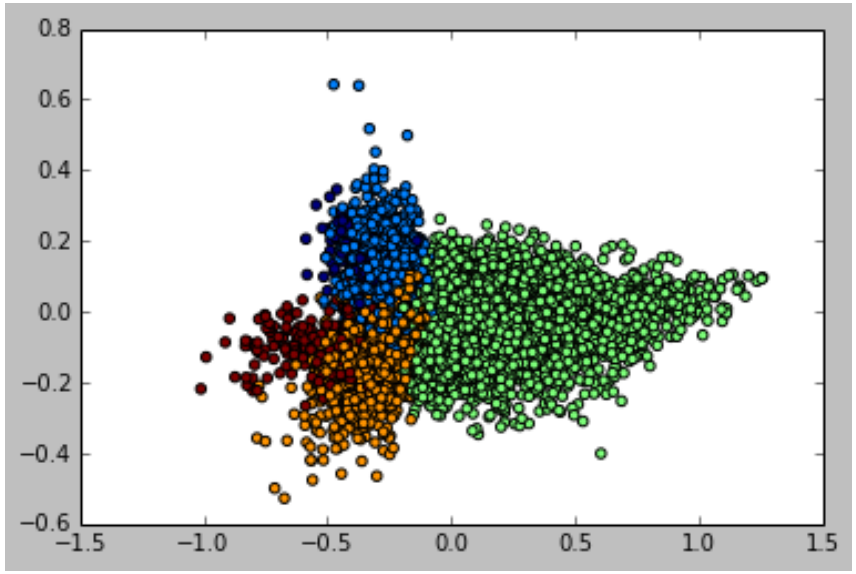


FIGURE : PCA 2D du modèle de Socher

Représentations apprises - Mots

n représentant en 2D

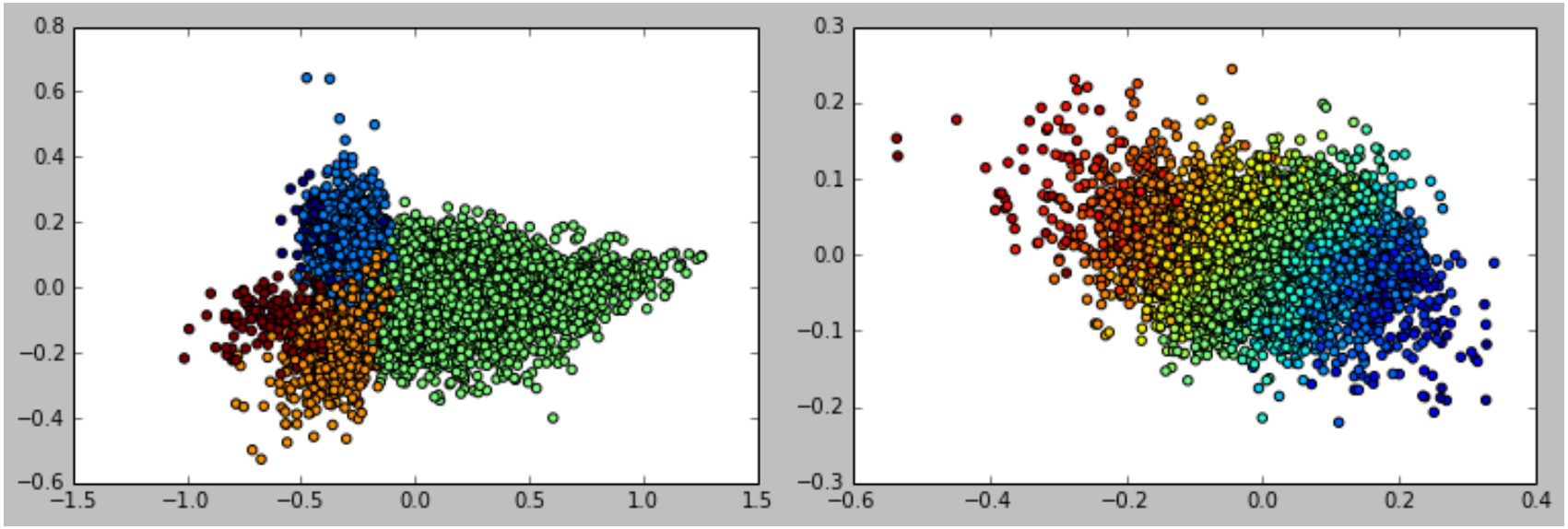


FIGURE : PCA de la représentation des mots en dimension 2.
. *(gauche)* Socher 30D *(droite)* Le nôtre 30D

Représentations apprises - Mots

n représentant en 2D

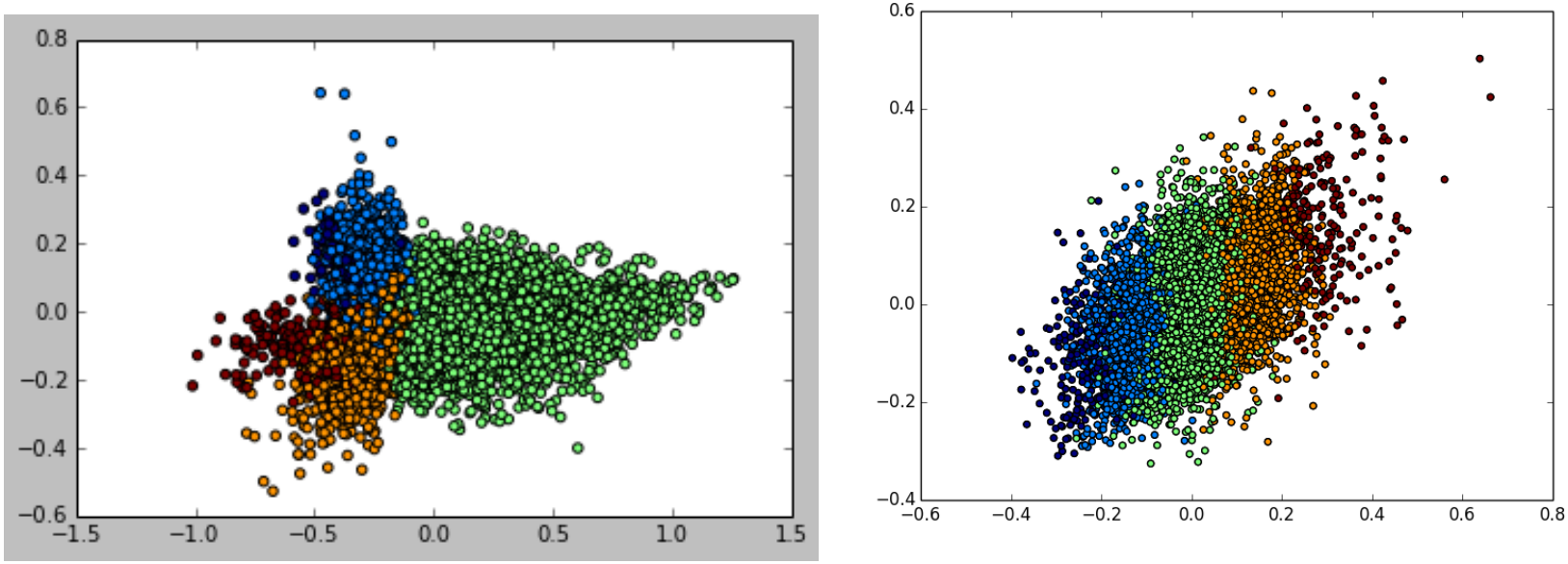


FIGURE : Socher 30D *(droite)* Le nôtre 2D

Représentations apprises - N-grams

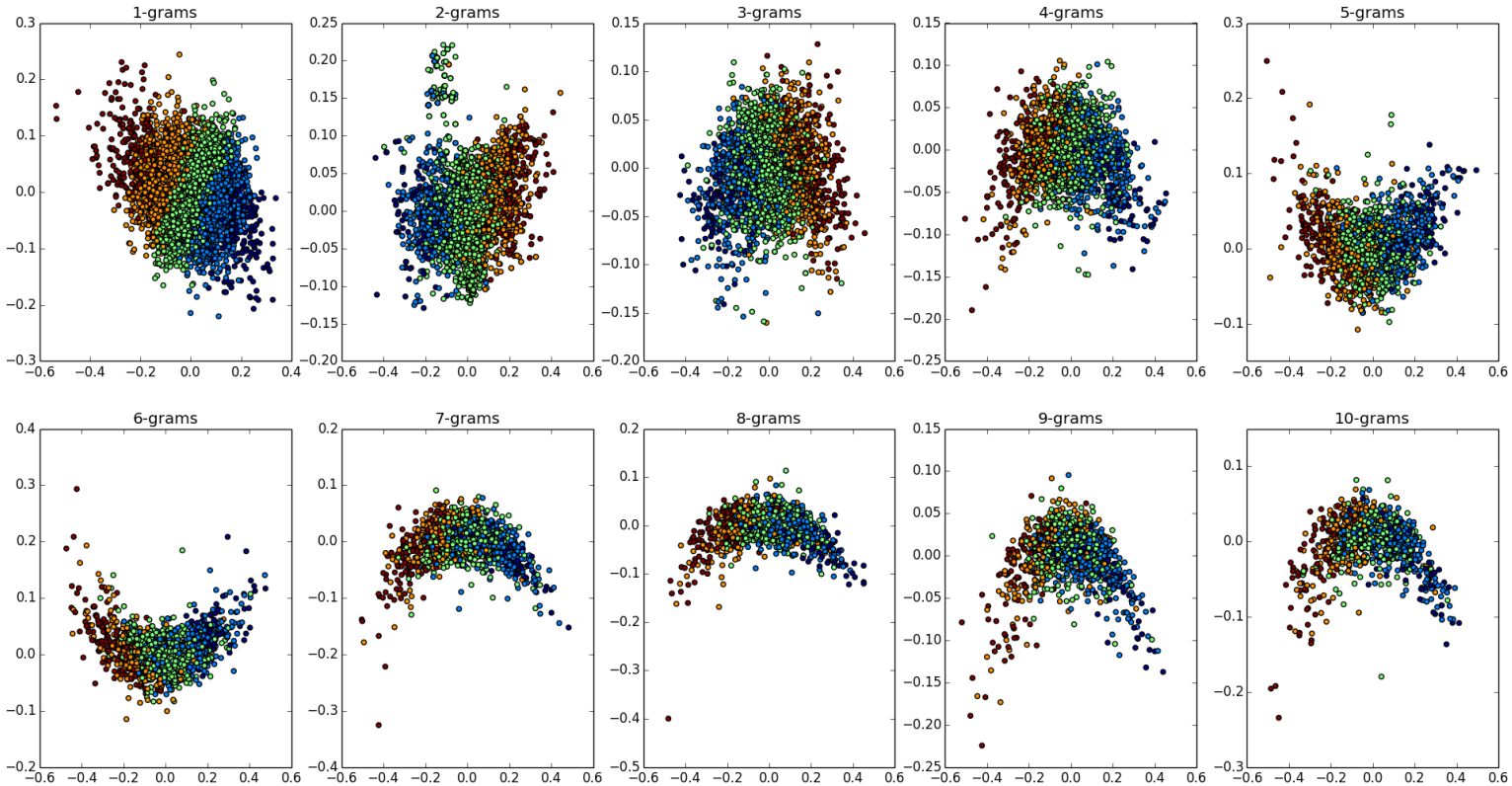
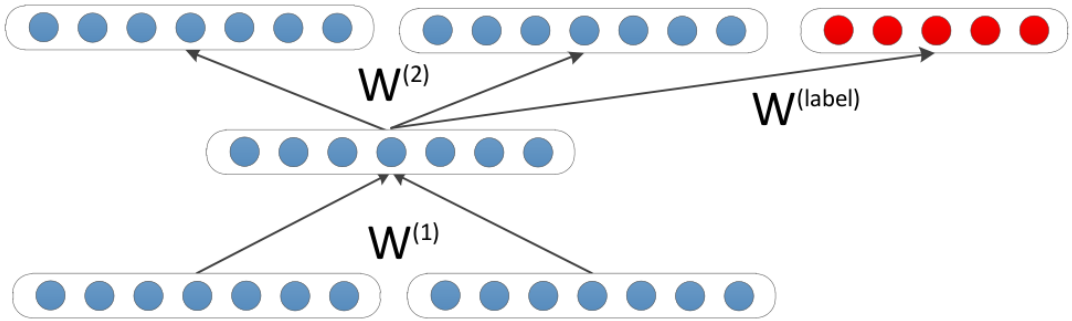


FIGURE : PCA de la représentation des mots en dimension 2 pour le modèle 30d

Auto encodeur ?

- Idée : les figures précédentes suggère que l'on recombine mal les représentations.
- Auto-encodeur optimise la reconstruction et peut espérer conserver la structure
  $\Rightarrow$ Meilleure transmission de l'information ?
- L'idée du modèle est



So far...

- Back propagation découle de celle de notre modèle précédent
- Ajout d'une dimension pour le sampling
- Pour le moment, pas de structure, mais du sentiment.

Question ?