

Modèle de Langage, Plongement statistiques

Leader : Chachura Baptiste, Follower : Bouger Lisa

November 13, 2024

1 Introduction

Le but de ce projet est d'étudier différents modèles de plongements statistiques sur des problèmes de traitement du langage. Nous nous pencherons sur une architecture bag of words (BOW) et mettrons en place différentes méthodes d'évaluations de nos modèles.

2 Modèle Initial

2.1 Description du modèle

Le but de ce modèle est d'apprendre des plongements statistiques sur un corpus de texte donnée.

2.1.1 Apprentissage

Une fenêtre de taille fixée parcourt l'ensemble du corpus, nous attribuons le contexte (contenu de cette fenêtre) au mot central de la fenêtre, ce contexte est appelé contexte positif. Pour chaque contexte positif créé nous générons k contextes négatifs. Dans ce premier modèle la création de ces contextes négatifs se fait grâce à un tirage aléatoire sur l'ensemble du vocabulaire. Pour l'apprentissage nous créons deux matrices : une pour stocker les plongements de nos mots ainsi qu'une matrice qui représente nos contextes. Nous créons ensuite un classifieur binaire qui a pour but de nous prédire si un mot c appartient au contexte d'un mot m . on note :

$$P(+|m, c_{\text{pos}}) = \sigma(m.c) = \frac{1}{1 + e^{-m.c}}$$

où sigma est la fonction sigmoïde.

2.1.2 Fonction de perte

Pour chaque mini-batch de l'ensemble d'apprentissage, on souhaite maximiser la probabilité de l'exemple positif : $P(+|m, c_{\text{pos}})$ et minimiser la probabilité des exemples négatifs : $P(+|m, c_{\text{neg}_i})$. Cela revient à maximiser l'expression suivante :

$$P(+|m, c_{\text{pos}}) \prod_{i=1}^k P(-|m, c_{\text{neg}_i})$$

Pour simplifier les calculs nous chercherons à minimiser la fonction de perte suivante :

$$L = -\log \left[P(+|m, c_{\text{pos}}) \prod_{i=1}^k P(-|m, c_{\text{neg}_i}) \right]$$

2.1.3 Evaluation

L'évaluation de ce modèle se fait sur un fichier test de la forme :

vélo bicyclette chat
mangue goyave balais
...

Nous effectuons une mesure de similarité entre les mots de chaque ligne et souhaitons obtenir une similarité plus importante entre les mots 1 et 2 que entre les mots 3 et 4. Pour effectuer cette mesure de similarité nous utilisons la similarité cosinus :

$$\text{sim_cos}(\mathbf{m}_1, \mathbf{m}_2) = \frac{\mathbf{m}_1 \cdot \mathbf{m}_2}{\|\mathbf{m}_1\| \|\mathbf{m}_2\|}$$

2.1.4 Résultats

Dans cette partie nous allons comparer les résultats de 240 expériences durant lesquelles nous étudions l'impact de divers paramètres du modèle. Dans un premier temps nous fixons la taille des plongements à 100 ainsi que le taux d'apprentissage à 0.1. Nous faisons varier les autres paramètres dans les intervalles suivants :

- [5, 7, 10, 20] Pour le nombre d'itération
- [1, 2, 3, 5, 7] pour la taille de la demi fenêtre du contexte
- [1, 3, 5, 10] pour le nombre d'exemples négatifs pour un exemple positif
- [1, 5, 10] pour filtrer les mots rares

3 Second Modèle

Cette fois ci les exemples négatifs ne seront plus générés de manière aléatoire mais nous effectuerons un tirage où la probabilité de tirer un mot correspond à sa fréquence dans le corpus d'entraînement. Cette méthode de génération des exemples permet d'éviter une sur-représentation des mots rares de notre corpus dans les contextes négatifs.

4 Opérations sur les plongements

Le but de cette partie est d'étudier dans quelle mesure est-il possible d'utiliser les vecteurs de plongement afin d'effectuer des opérations sémantiques. Un exemple classique de ce genre d'opérations est 'Père' - 'Homme' + 'Femme' = 'Mère'. Pour évaluer cette capacité nous créons un jeu de données contenant une centaine d'opérations similaires sur ces mots.

Pour l'évaluation nous utilisons le modèle 43 (hébergé à <http://vectors.nlpl.eu/repository>), il contient plus de 2,5 millions de mots et des plongements de dimension 100.

5 Conclusion