

# AMAL - TP 9

## Mécanisme d'attention

Nicolas Baskiotis - Edouard Oyallon - Benjamin Piwowarski - Laure Soulier

2021-2022

### Introduction (brève, cf cours)

Dans ce TP, nous allons nous intéresser aux mécanismes d'attention qui permettent aux réseaux de neurones de se focaliser sur un certains nombre d'entrées : par exemple, pour savoir si un sentiment est présent dans une phrase ou non, il n'est pas nécessaire de regarder l'ensemble des mots mais seulement les parties qui expriment des sentiments.

### Préparation des données et du modèle

Vous utiliserez les données IMDB qui contient 50,000 commentaires venant de Internet Movie Database (IMDb), et des plongements de mots Glove (voir le code fourni pour le TP). Le but est de classifier un commentaire comme étant “positif” (pos) ou “négatif” (neg) ; la mesure de performance est le taux de bonne classification.

## 1 Modèle de base

#### Question 1

Vous implémenterez un modèle simple basé sur les embeddings (figure 1) : un texte  $t = (t_1, \dots, t_n)$  est représenté par la moyenne des embeddings  $\mathbf{x}_i$  des mots qu'il contient, i.e.

$$\hat{\mathbf{t}} = \frac{1}{n} \sum_i \mathbf{x}_i$$

Utilisez un simple modèle linéaire avec un coût cross-entropique.

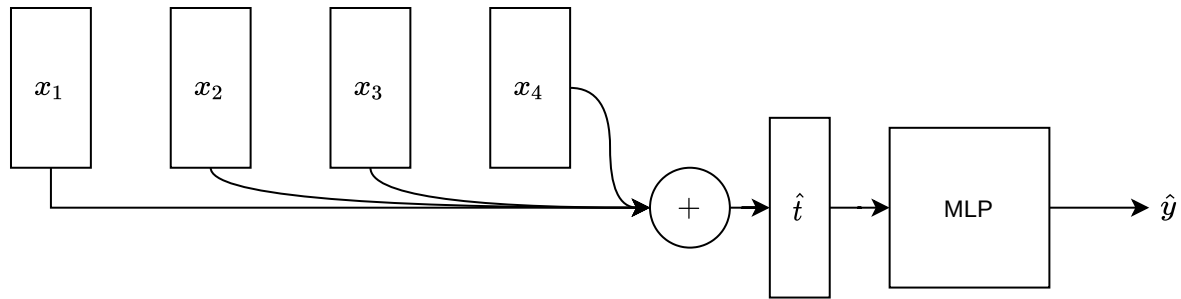


FIGURE 1 – Pooling générique

## 2 Attention simple

Le modèle précédent donne de piètres résultats car tous les mots sont utilisés avec la même pondération pour calculer la représentation barycentrique du texte. L'objectif de ce premier modèle (figure 2) d'attention très simple est de pondérer très fortement les mots qui ont un intérêt pour la tâche de classification afin de construire la représentation pondérée du texte :

$$\hat{\mathbf{t}} = \sum_i p(\mathbf{a}_i|\mathbf{t}) \mathbf{x}_i$$

avec les coefficients  $p(\mathbf{a}_i|\mathbf{t})$  qui représentent la probabilité de porter l'attention sur le  $i$ -ème token du texte. On utilise ensuite comme précédemment un classifieur linéaire.

On propose de modéliser ces probabilités de la manière suivante :

$$\log p(\mathbf{a}_i|\mathbf{t}) = \text{constante} + \mathbf{q} \cdot \mathbf{x}_i$$

avec  $\mathbf{q}$  un vecteur appelé **question** (ou *query*) : il permet d'exprimer la pertinence du  $i$ -ème token avec la question posée. Ce vecteur sera appris durant la phase d'apprentissage (il s'agit d'un embedding de la question).

### Question 2

Implémenter le modèle. Réfléchir à quelle fonction simple permet d'obtenir  $p(\mathbf{a}_i|t)$  à partir de  $\mathbf{q} \cdot \mathbf{x}_i$ .

Lors de l'apprentissage, calculer l'histogramme de l'entropie des attentions. Lors de l'inférence, observer pour des exemples précis sur quels mots se portent l'attention, ce qui permet de voir quel(s) mot(s) ont été importants pour la décision.

**Point de vocabulaire et attention globale :** les embeddings des mots peuvent être vus comme des clés  $\mathbf{k}_i$  (*key* en anglais) dont on cherche les plus similaires à la question posée

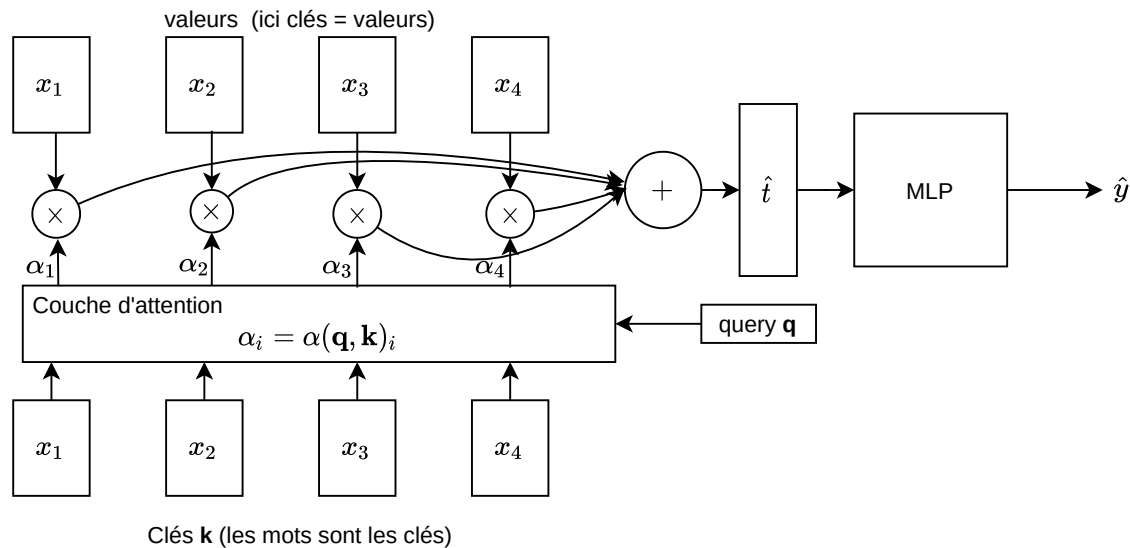


FIGURE 2 – Modèle d'attention

**q.** À chaque clé est associée une valeur (dans notre cas la clé et la valeur sont les mêmes - l'embedding du mot) et cette valeur est pondérée par la similarité entre la question et la clé - l'attention portée à la  $i$ -ème valeur. On note généralement le vecteur d'attention  $\alpha(\mathbf{q}, \mathbf{k})$ .

### 3 Question et valeur

On peut souhaiter faire dépendre la question  $\mathbf{q}$  du texte lui même : par exemple, le vocabulaire pour parler de ses sentiments sur un film d'horreur ne sera pas le même que pour des films de romance (quoique...).

#### Question 3

Afin de faire cela, vous allez calculer la question  $\mathbf{q}$  en fonction du texte en utilisant la représentation du texte moyennée  $\hat{\mathbf{t}}_m$  (exercice 1) et une transformation linéaire (vous pouvez complexifier ce modèle à volonté en fonction de vos résultats).

Utiliser ensuite cette question pour calculer la représentation du document avec

$$\log p(\mathbf{a}_i | \mathbf{t}) = \text{constante} + \mathbf{q}(\hat{\mathbf{t}}_m) \cdot \mathbf{x}_i$$

De même, il peut être intéressant de changer la représentation d'un token pour la décision en changeant la *valeur* d'un plongement de mot :

$$\hat{\mathbf{t}} = \sum_i p(\mathbf{a}_i|\mathbf{t})v_{\theta}(\mathbf{x}_i)$$

## 4 Optionnel : Modèles contextuels et attention

### Question 4

Créez un nouveau modèle qui combine un LSTM ou GRU afin de créer des plongements *contextualisés*. Vous pouvez utiliser une ou plusieurs couches de LSTM/GRU. La couche d'attention porte cette fois sur la séquence d'états issus de la dernière couche récurrente. On utilise comme dans les questions précédentes une dernière couche linéaire pour classifier qui prend en entrée la somme pondérée des états.

## 5 Optionnel : Entropie

### Question 5

Afin de forcer l'attention à se focaliser sur une partie de l'entrée, il est possible d'utiliser un critère entropique. Modifiez les modèles d'attentions en ajoutant cette régularisation et observez son effet.