Quand les mots deviennent des vecteurs

Comment une IA représente le sens des mots

Mise en situation. Pour un ordinateur, un mot n'est qu'une suite de lettres. Les mots « chat » et « chas » (le trou d'une aiguille) ne diffèrent que d'une lettre, alors que « chat » et « chien » n'ont que leurs deux premières lettres en commun. Pourtant, pour nous, un chat ressemble bien plus à un chien qu'au trou d'une aiguille !
Pour qu'un agent conversationnel (ChatGPT, Le Chat, Gemini…) puisse travailler sur le **sens** des mots, on commence donc par transformer chaque mot en **nombres** : chaque mot devient un **point** d'un repère, de façon que deux mots de sens proches soient représentés par deux points proches.
Dans cette activité, vous allez découvrir cette idée… et vous servir de tout ce que vous savez sur les **distances** et les **vecteurs** !

1. Un mot, des coordonnées

Définition

On décrit chaque mot par une liste de nombres, appelés ses **coordonnées**. Le mot est alors représenté par un **point** dans un repère (ou par un **vecteur**). On parle de **plongement de mots** (en anglais *word embedding*).
Règle d'or : deux mots de sens proches doivent être représentés par des points **proches**.

Exemple

On décrit des animaux avec deux caractéristiques notées de 0 à 10 : leur **taille** (abscisse x) et leur **côté sauvage** (ordonnée y, de 0 pour un animal domestique à 10 pour un animal très sauvage).

Mot	chat	chien	lapin	cheval	vache	renard	loup	lion	tigre	ours
x **(taille)**	2	3	1	6	7	2	4	8	8	9
y **(sauvage)**	1	2	3	2	1	8	8	9	10	7

Par exemple, le mot `chat` est représenté par le point de coordonnées $(2\,;1)$: un petit animal, très peu sauvage.

Rappel – distance entre deux points

Dans un repère orthonormé, la distance entre $A(x_A\,;y_A)$ et $B(x_B\,;y_B)$ est :

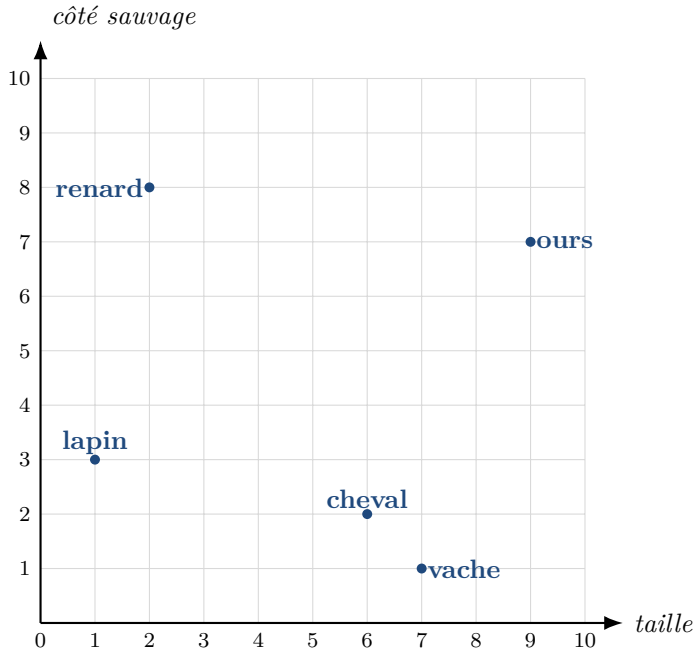
$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

▶ À vous de jouer !

1. Dans le repère ci-dessous, placer les points `chat`, `chien`, `loup`, `lion` et `tigre` (les autres sont déjà placés).
2. Calculer la distance entre `loup` et chacun des mots `chien`, `renard` et `lion`. Arrondir au dixième.
3. Quel est le mot le plus proche de `loup` ? Est-ce surprenant ?
4. En lisant le graphique, quel semble être le mot le plus proche de `tigre` ? Vérifier par un calcul.
5. Un biologiste dirait que le loup est un proche cousin du chien. Comment pourrait-on modifier la description des mots pour que `loup` se rapproche de `chien` ?

Coup de pouce

Pour la distance entre `loup`$(4\,;8)$ et `lion`$(8\,;9)$: on calcule d'abord les écarts $8-4$ et $9-8$, on les élève au carré, on les additionne, puis on prend la racine carrée.



Remarque

Dans les vraies IA, ce ne sont pas des humains qui choisissent les coordonnées : elles sont **calculées automatiquement** à partir d'énormes quantités de textes (voir la partie 5). Chaque mot reçoit **plusieurs centaines** de coordonnées, qui n'ont en général pas de nom : personne ne sait dire exactement ce que « mesure » chacune d'elles. La formule de la distance reste la même : avec trois coordonnées, on additionne trois carrés sous la racine, avec 300 coordonnées… 300 carrés.

2. Calculer avec des mots : les analogies

Rappel – vecteurs

- Le vecteur $\overrightarrow{AB}$ a pour coordonnées $\begin{pmatrix} x_B - x_A \\ y_B - y_A \end{pmatrix}$.
- Si D est l'image de C par la translation de vecteur $\overrightarrow{AB}$, alors $\overrightarrow{CD} = \overrightarrow{AB}$, et donc :

$$x_D = x_C + (x_B - x_A) \qquad \text{et} \qquad y_D = y_C + (y_B - y_A)$$

- $\overrightarrow{AB} = \overrightarrow{CD}$ si et seulement si $ABDC$ est un parallélogramme (éventuellement aplati).

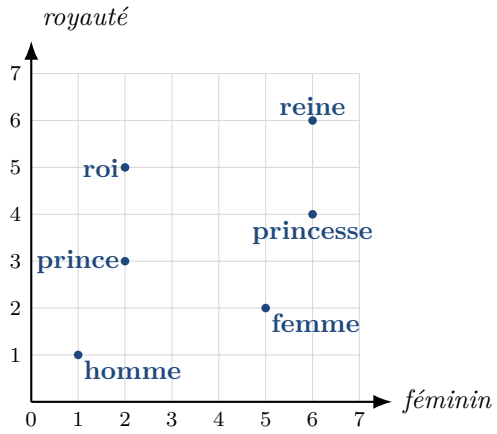
Définition

Une **analogie** est une phrase de la forme « *A est à B ce que C est à D* ». Par exemple : « *Paris est à la France ce que Rome est à l'Italie* ».
Avec des vecteurs de mots, pour trouver le mot D qui manque, la machine :

1. calcule les coordonnées du vecteur $\overrightarrow{AB}$;
2. calcule le point D, image de C par la translation de vecteur $\overrightarrow{AB}$;
3. cherche le mot dont le point est **le plus proche** de D (en dehors des trois mots de départ).

On donne les vecteurs de six mots (axe des abscisses : « féminin » ; axe des ordonnées : « royauté »).

Mot	x	y
homme	1	1
femme	5	2
prince	2	3
princesse	6	4
roi	2	5
reine	6	6



▶ À vous de jouer !

On cherche à compléter l'analogie : « `homme` *est à* `femme` *ce que* `roi` *est à . . .* ».

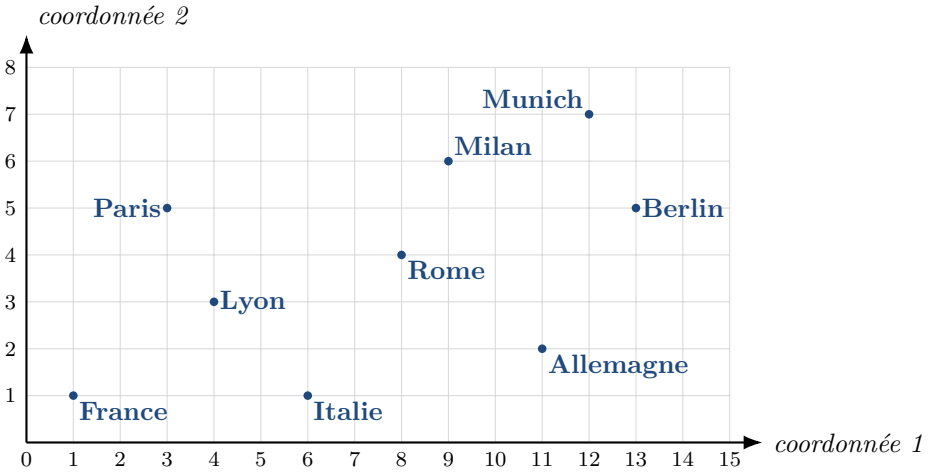
1. Calculer les coordonnées du vecteur $\vec{u}$ allant de `homme` à `femme`.
2. Calculer les coordonnées du point D, image de `roi` par la translation de vecteur $\vec{u}$. Quel mot se trouve en D ?
3. Quelle est la nature du quadrilatère formé par `homme`, `femme`, `reine` et `roi` (dans cet ordre) ? Justifier.
4. Calculer les coordonnées du vecteur allant de `prince` à `roi`. En déduire le mot qui complète : « `prince` *est à* `roi` *ce que* `princesse` *est à . . .* ».

Coup de pouce

Pour la question 2 : on ajoute à chaque coordonnée de `roi` la coordonnée correspondante de $\vec{u}$: $x_D = x_{\text{roi}} + x_{\vec{u}}$ et $y_D = y_{\text{roi}} + y_{\vec{u}}$.

Dans la réalité, les points ne tombent jamais **exactement** sur un mot : c'est pour cela que la

machine cherche le mot le plus proche.



▶ À vous de jouer !

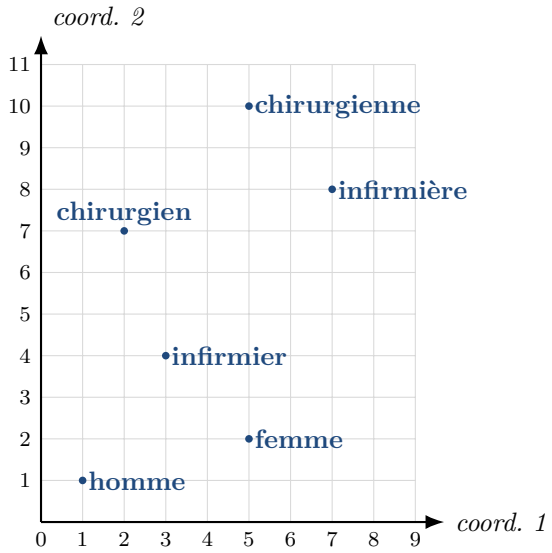
Les vecteurs de quelques mots sont représentés ci-dessus. On cherche à compléter : « `France` *est à* `Paris` *ce que* `Italie` *est à . . .* ».

1. Lire les coordonnées de `France`, `Paris` et `Italie`, puis calculer les coordonnées du vecteur $\overrightarrow{v}$ allant de `France` à `Paris`.
2. Calculer les coordonnées du point D, image de `Italie` par la translation de vecteur $\overrightarrow{v}$. Le point D correspond-il à un mot ?
3. Calculer la distance entre D et `Rome`, puis entre D et `Milan`. Quel mot la machine propose-t-elle ?
4. Que représente, en français, le vecteur $\overrightarrow{v}$?

3. Des vecteurs qui ont des préjugés

Les coordonnées des mots sont apprises dans des textes écrits par des humains (articles de presse, pages web. . .). Une machine a appris les vecteurs suivants (valeurs simplifiées pour l'activité, mais qui imitent un phénomène réellement observé).

Mot	x	y
homme	1	1
femme	5	2
chirurgien	2	7
chirurgienne	5	10
infirmier	3	4
infirmière	7	8



▶ À vous de jouer !

On demande à la machine de compléter : « **homme** *est à* **chirurgien** *ce que* **femme** *est à . . .* ».

1. Calculer les coordonnées du vecteur allant de **homme** à **chirurgien**, puis celles du point D, image de **femme** par la translation de ce vecteur.
2. Calculer la distance entre D et **infirmière**, puis entre D et **chirurgienne**. Quel mot la machine propose-t-elle ? Quel mot attendait-on ?
3. Faire de même pour : « **femme** *est à* **infirmière** *ce que* **homme** *est à . . .* » (comparer les distances à **chirurgien** et à **infirmier**).
4. D'où vient ce problème, selon vous ? Pourquoi serait-il grave si ces vecteurs étaient utilisés, par exemple, par un logiciel qui trie des candidatures à un emploi ?

Coup de pouce

La méthode est exactement celle de la partie 2. Attention aux signes dans le calcul des distances : $(-1)^2 = 1$.

Remarque

En 2016, des chercheurs (T. Bolukbasi et ses collègues) ont montré que des vecteurs de mots appris sur des articles de presse produisaient des analogies stéréotypées, comme l'annonce le titre de leur article : « *L'homme est au programmeur ce que la femme est à la femme au foyer ?* ». Il existe des méthodes pour corriger ces biais, mais elles ne les font pas complètement disparaître : il faut toujours garder un regard critique sur les réponses d'une IA.

4. Programmer en Python

Exemple

On reprend la fonction `distance` de l'activité sur les k plus proches voisins, et on écrit une fonction qui calcule le point D d'une analogie « A est à B ce que C est à D » :

```
from math import sqrt

def distance(xA, yA, xB, yB):
    return sqrt((xB - xA)**2 + (yB - yA)**2)

def analogie(xA, yA, xB, yB, xC, yC):
    xD = xC + (xB - xA)
    yD = yC + (yB - yA)
    return xD, yD

print(analogie(1, 1, 5, 2, 2, 5))
```

▶ À vous de jouer !

1. Quels mots de la partie 2 correspondent aux points A, B et C utilisés dans la dernière ligne ? Qu'affiche le programme ?
2. On stocke les mots de la partie 2 dans trois listes. Compléter la fonction `mot_le_plus_proche` (les `...` sont à remplacer) :

```
mots = ["homme", "femme", "prince", "princesse", "roi", "reine"]
xs = [1, 5, 2, 6, 2, 6]
ys = [1, 2, 3, 4, 5, 6]

def mot_le_plus_proche(x, y):
    meilleure_distance = distance(x, y, xs[0], ys[0])
    meilleur_indice = 0
    for i in range(1, len(mots)):
        d = distance(x, y, ..., ...)
        if d < meilleure_distance:
            meilleure_distance = ...
            meilleur_indice = ...
    return mots[meilleur_indice]
```

3. Que renvoie `mot_le_plus_proche(6, 6)` ? Et `mot_le_plus_proche(1, 2)` ?

Coup de pouce

Cette fonction ressemble beaucoup à la fonction `plus_proche_voisin` de l'activité sur les k plus proches voisins : seules les listes ont changé !

5. Pour aller plus loin : d'où viennent les coordonnées ?

Remarque

Les vraies coordonnées des mots reposent sur une idée du linguiste J. R. Firth (1957) : « *on reconnaît un mot à ses fréquentations* ». Deux mots qui apparaissent dans les **mêmes contextes** (entourés des mêmes mots) reçoivent des vecteurs proches. En 2013, le programme *word2vec*, conçu chez Google, calculait ainsi 300 coordonnées pour chaque mot à partir de milliards de mots de textes. Les modèles de langage actuels commencent tous par transformer chaque morceau de mot (*token*) en un vecteur de plusieurs milliers de coordonnées.

★ Question étoilée

On reprend le corpus de l'activité « Prédire le mot suivant » :

P1. `le chat mange la souris .`
P2. `le chat mange la viande .`
P3. `le chat dort sur le tapis .`
P4. `le chien chasse la souris .`
P5. `le chien mange la viande .`
P6. `la souris mange le fromage .`
P7. `la souris dort sur le lit .`

On donne à chaque nom deux coordonnées calculées à partir de ses « fréquentations » :

- x = nombre de fois où le mot est suivi d'un verbe (`mange`, `dort` ou `chasse`) ;
- y = nombre de fois où le mot est suivi du point `.` (fin de phrase).

1. Calculer les coordonnées de `chat`, `chien`, `souris`, `viande`, `fromage`, `tapis` et `lit`, puis placer les points dans un repère.
2. Calculer la distance entre `chat` et `chien`, puis entre `chat` et `viande`. Que remarque-t-on sur la position des mots ?
3. Pourquoi `souris` se trouve-t-elle « entre les deux groupes » ?
4. Quel problème posent `fromage`, `tapis` et `lit` ? Comment pourrait-on le résoudre ?

À retenir

- Une IA représente chaque mot par une liste de nombres : un **vecteur** (plongement de mots). Des mots de sens proches ont des vecteurs **proches**, ce que l'on mesure par une **distance**.
- Les **différences** de vecteurs représentent des relations : on peut calculer des **analogies** par translation, puis chercher le mot le plus proche.
- Les coordonnées sont apprises dans des textes écrits par des humains : elles reproduisent leurs **biais** et leurs **stéréotypes**. Une IA n'est pas neutre !