# Big Data Analytics

Vecteurs Textuels

# Vecteurs Textuels

Représentation TF-IDF.
Un document = un vecteur de la taille d'un dictionnaire.

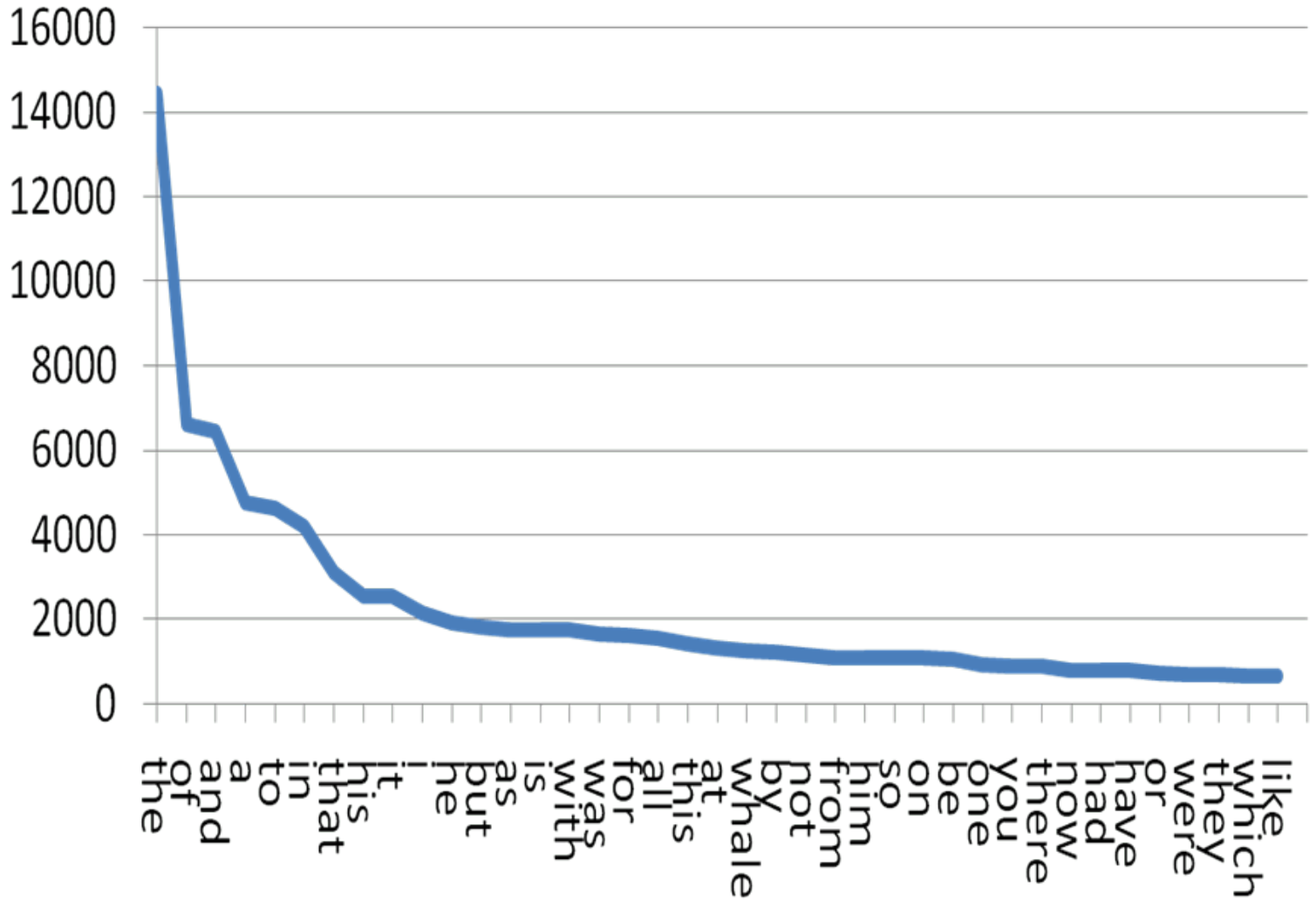
$$\boxed{w_{i,j} = tf_{i,j} * \log \frac{N}{df_i}}$$

où :
$tf_{i,j}$ =nombre d'occurence du mot i dans le document j,
$df_i$ =nombre de documents contenant le mot i,
$N$ =nombre total de documents

$\Rightarrow$ produict scalaire, SVM, arbres, réseaux de neurones, ...

# Vecteurs Textuels

Loi de Zipf, justifiant l'utilisation du terme $df_i$



# Vecteurs Textuels

Utilisation de N-gram de mots ou de caractères.

Ex : “Le chien mange de la viande”
Dictionnaire 2-gram de mots :

- [le-chien,chien-mange,mange-de,de-la,la-viande]

Dictionnaire 3-gram de caractères :

- [le_,e_c,_ch,chi,hie,ien,en_,n_m,_ma,man,...]

# Vecteurs Textuels

Latent Semantic Analysis :

$\approx$ ACP sur la matrice des documents, de telle sorte qu'on obtient des relations entre les mots.
Les directions principales de projection nous donne des "concepts" génraux liés au langage.

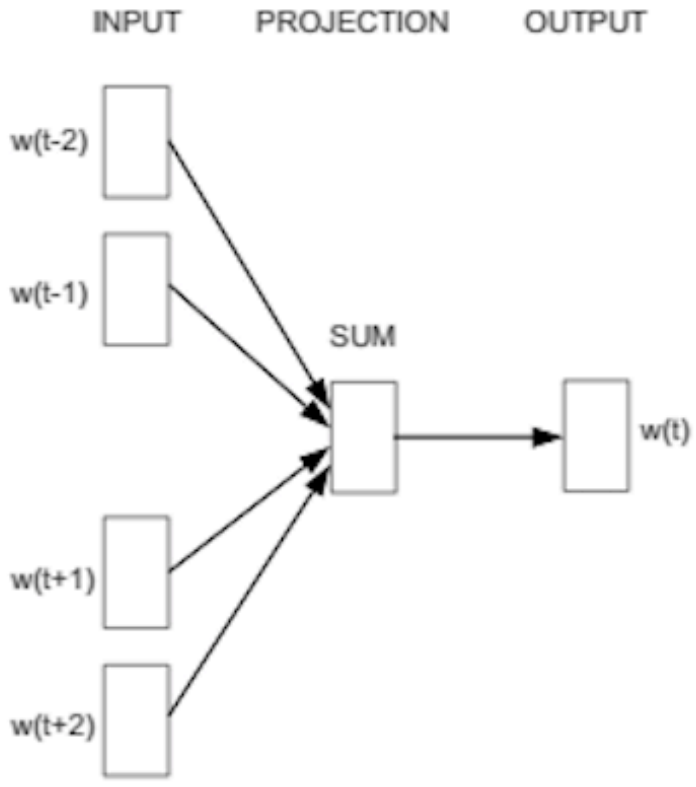
Par exemple : un axe correspond au "stop words", un autre au champ lexical du sport, un autre à l'économie, etc...

# Traitement du langage : word embeddings

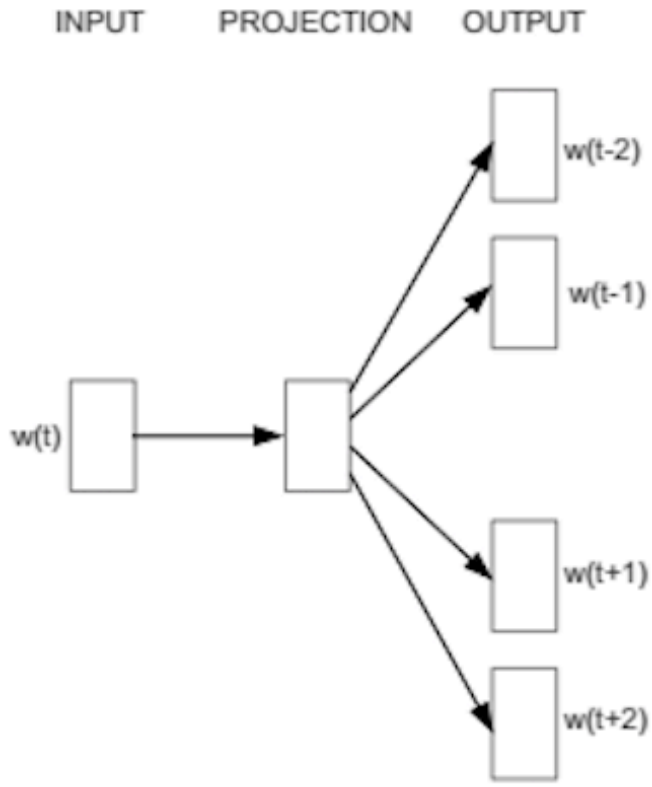
mot = indice dans un dictionnaire (dimension $> 30000$)
mot = vecteur “sémantique” (dimension $< 1000$)

- word2vec
- CBOW/Skip-Gram
- GloVe
- Thought vector (pour des phrases ou même des documents entiers)
- ...

# Traitement du langage : word embeddings

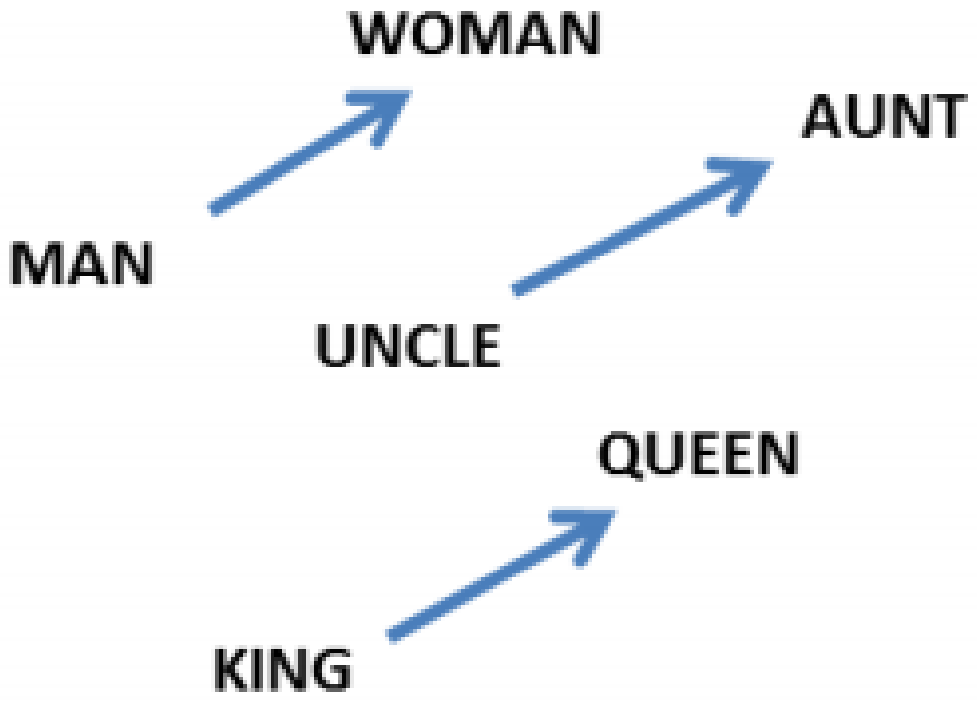



CBOW




Skip-gram

# Traitement du langage : word embeddings




Visualisation de l'espace word2vec

Word Embeddings à télécharger