



UMR 5205 CNRS

Natural Language Processing

Haytham Elghazel

Laboratoire d'InfoRmatique en Image et Systèmes d'information

Pôle Data Science, Equipe DM2L



INSA



Lyon 1

UNIVERSITÉ
LUMIÈRE
LYON 2



Contexte : Text Mining

- ❖ **Fouille de données textuelles** : processus d'extraction non triviale d'informations utiles inconnues *a priori* à partir de grands volumes de textes .
- ❖ **Spécificité** : les données sont sous une forme qui n'est pas directement exploitable par les méthodes classiques de ML
- ❖ **Plusieurs objectifs** :
 - ❑ **Identification de thèmes** : regroupement de (parties de) textes en thèmes inconnus *a priori* (**Topic Modeling**)
 - ❑ **Affectation de (parties de) textes à des catégories (classes)** prédefinies
 - ❑ **Recherche de « variables » explicatives pertinentes** utilisables ensuite conjointement avec d'autres variables (quantitatives, nominales)
 - ❑ **Extraction d'informations** : mise en correspondance des textes avec des « schémas » plus directement exploitables par des méthodes classiques de ML.

Contexte : Text Mining (Applications)

- ❖ Gestion de la relation client : Détermination de catégories de clients à partir de leurs échanges avec le service client, redirection des courriels mal adressés
- ❖ Identification de l'objet des retours négatifs fréquents, détermination des causes majeures de l'attrition de clientèle
- ❖ Détermination de l'image d'une famille de produits
- ❖ Détermination des attentes majeures dans l'évolution des produits
- ❖ Identification de tendances à partir de messages postés sur des médias sociaux : Produits ou familles de produits recherchés, caractéristiques recherchées pour des produits d'une certaine famille
- ❖ Analyse de comptes rendus médicaux
- ❖ Génération/Résumé de textes
- ❖ Traduction de textes

Text Mining : Préparation des données

- ❖ **Pré-traitement des données textuelles** : uniformisation du codage, élimination éventuelle de certains caractères spéciaux (sauts de lignes, symboles, etc. suivant l'objectif), «traduction» de langage SMS...
- ❖ **Extraction d'informations** : suppression des « mots ignorés » (stop words)
- ❖ **Extraction d'entités primaires** : mots, éventuellement composés (« chauve-souris »), locutions nominales (« chemin de fer »), verbales (« arrondir les angles »)...
- ❖ **Étiquetage grammatical** : caractérisation grammaticale de chaque composante du texte par une catégorie lexicale (ex. nom commun, nom propre, verbe, adverbe...) et une fonction (ex. sujet, complément d'objet direct...)

Text Mining : Préparation des données

- ❖ Extraction d'entités nommées : noms de personnes, de lieux (« Mont Blanc »), d'organisations, dates... qui jouent souvent un rôle important dans les opérations d'analyse de textes.
- ❖ Lemmatisation ou Racinisation : remplacer chaque mot (par ex. « pensons ») par sa forme canonique (« penser ») ou par sa racine (« pense ») ; peuvent engendrer des confusions (par ex. « organ » pour « organe » comme pour « organisation »). Cette étape est utile car elle permet de traiter comme un mot unique les différentes variantes issues d'une même forme canonique ou racine

Racinisation suffisante pour l'anglais, lemmatisation mieux adaptée au français

Text Mining : Exploitation

□ *Suite à la préparation (à voir si nécessaire) :*

- ❖ Représentation vectorielle des textes : pondération tf*idf, décomposition en valeurs singulières (SVD, LSA), analyse sémantique explicite (ESA), Word Embedding (plongement lexical) comme Word2vec, Bert, GPT, etc.
- ❖ Développement de modèles sur la base du contenu textuel seul ou en ajoutant des variables quantitatives et nominales
- ❖ Utilisation des modèles développés, évaluation des résultats et interprétation

Text Mining : challenges

□ *Attention à quelques challenges :*

- ❖ **Résolution référentielle** : cherche à identifier l'entité, explicitement présente ailleurs dans le texte, par ex., à qui fait référence **Il** dans « Barack Obama est le 44e président des États-Unis. Il est né le 4 août 1961 à Honolulu »
- ❖ **Analyse syntaxique (générale ou spécifique)** : de la négation (→ distinction entre affirmation et négation), « quantification » des adverbes (ex. « très abouti / plus ou moins abouti / peu abouti »)

Représentations vectorielles de textes

- ❖ **Objectif** : pouvoir manipuler des données textuelles avec les nombreux outils disponibles pour les espaces vectoriels
- ❖ Au préalable, possible suppression des « mots ignorés » (stop words) : prépositions, conjonctions, articles, verbes auxiliaires...
 - L'ensemble des mots à ignorer peut dépendre de l'objectif de l'analyse !
 - Doit être appliquée seulement *après* l'extraction de locutions (ex. « chemin de fer »)
- ❖ **Modèle vectoriel de texte** : affecter une dimension de l'espace à chaque terme (lemme, entité nommée...) trouvé dans la base de documents → chaque texte est représenté par un vecteur de grande dimension, (très) clairsemé.

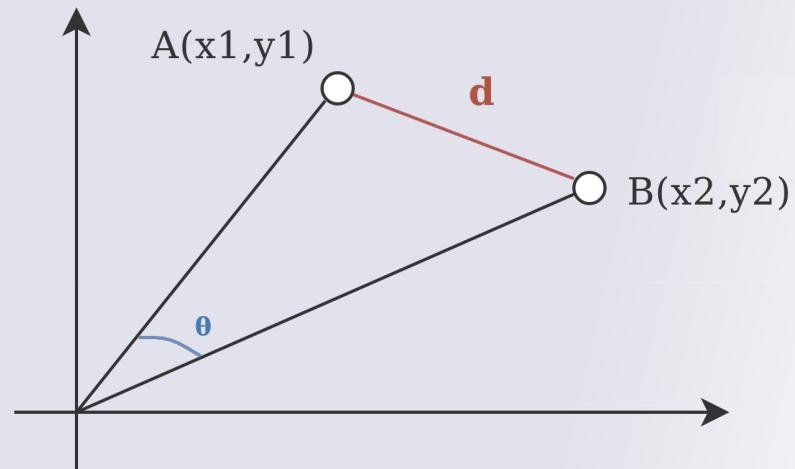
Représentations vectorielles de textes



Représentations vectorielles de textes

- ❖ Comparaison des vecteurs avec la distance cosinus : la norme du vecteur étant proportionnelle à la longueur du texte, mieux vaut mesurer l'angle entre vecteurs que la distance euclidienne.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}},$$

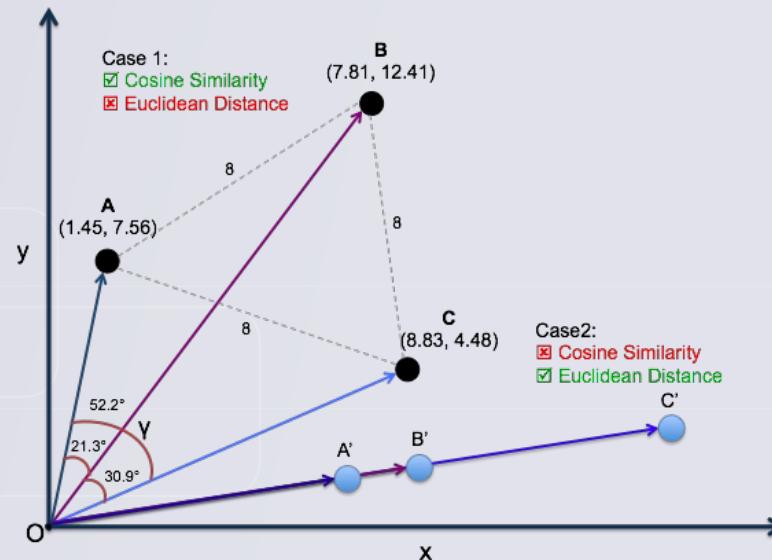


Représentations vectorielles de textes

Exemple 1

	Data	Science	Mining	Storage	Book
Doc 1	10	10	10	0	0
Doc 2	1	1	1	0	0
Doc 3	1	0	0	1	1

Exemple 2



Représentations vectorielles de textes

❖ Evolutions de la représentation vectorielle de base :

- Pondérations de termes (TF-IDF)
- Sélection de termes (test du χ^2)
- Analyse sémantique latente (LSA)

TF-IDF

- ❖ Des pondérations spécifiques, prédefinies, peuvent être utilisées, par ex. sur-pondérer les termes des titres de sections et sous-sections, sur-pondérer certaines entités nommées, etc.
- ❖ Objectif TF-IDF : pondérer les termes suivant leur « importance » déterminée automatiquement
 - Fréquence d'un terme dans un document (*term frequency*, TF) : l'importance d'un terme pour un document est proportionnelle au nombre d'occurrences du terme dans le document

$$tf_{ij} = n_{ij} / \|d_j\|$$

n_{ij} étant le nombre d'occurrences du terme i dans le document j et $\|d_j\|$ la longueur du document d_j

- Inverse de la fréquence dans les documents (*inverse document frequency*, IDF) : l'importance d'un terme pour tous les documents est inversement proportionnelle au nombre de documents dans lequel il apparaît (les termes présents dans peu de documents sont plus discriminants que les termes présents dans beaucoup de documents)

$$idf_i = \log(n/n_i)$$

n étant le nombre total de documents, n_i le nombre de ceux contenant le terme i

TF-IDF

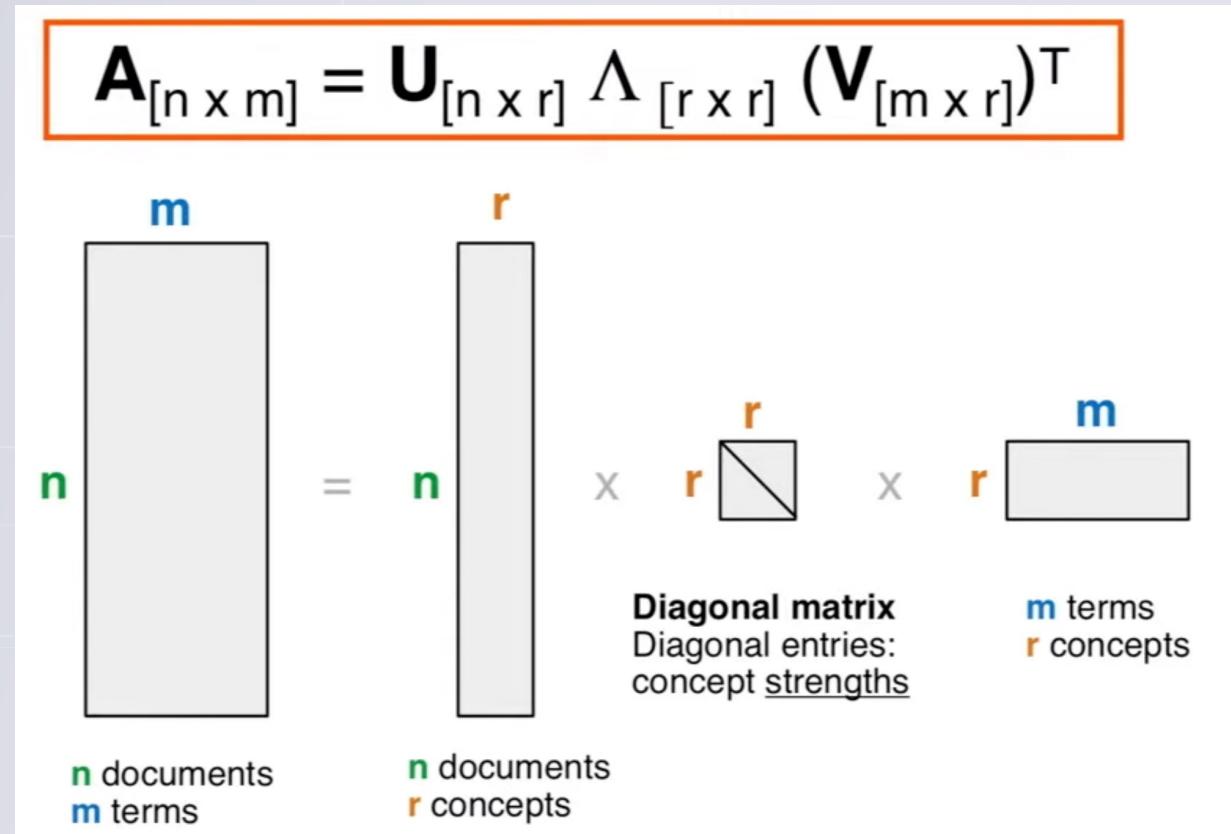


Latent Semantic Analysis

- ❖ **Objectif LSA :** recherche de « concepts », en (relativement) faible nombre, correspondant à des corrélations entre termes, pour représenter les documents d'une collection.
 - ❑ Remplacer les lemmes individuels par des « concepts » correspondant à des groupes de lemmes souvent présents ensemble (identifiés par la LSA)
 - ❑ Réduction du « bruit » engendré par les mots rares, problème de **synonymie et polysémie**.
- ❖ **Concept (latent) = groupe de mots (lemmes) présents souvent ensemble (corrélés) dans des documents :**
 - ❑ {procès, juge, tribunal, plainte, inculpé, procureur, avocat, condamnation}
 - ❑ {assiette, couteau, chef, mayonnaise, avocat, crevettes, plateau}

Latent Semantic Analysis - SVD

- ❖ Principe de la LSA : décomposition en valeurs singulières de la matrice
- ❖ Théorème : toute matrice A de taille $n \times m$ peut se décomposer en un produit matriciel :



Latent Semantic Analysis - SVD

$$\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \Lambda_{[r \times r]} (\mathbf{V}_{[m \times r]})^T$$

A: n x m matrix

e.g., n documents, m terms

U: n x r matrix

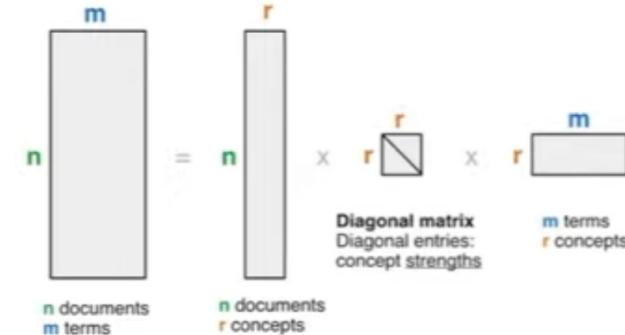
e.g., n documents, r concepts

Λ : r x r diagonal matrix

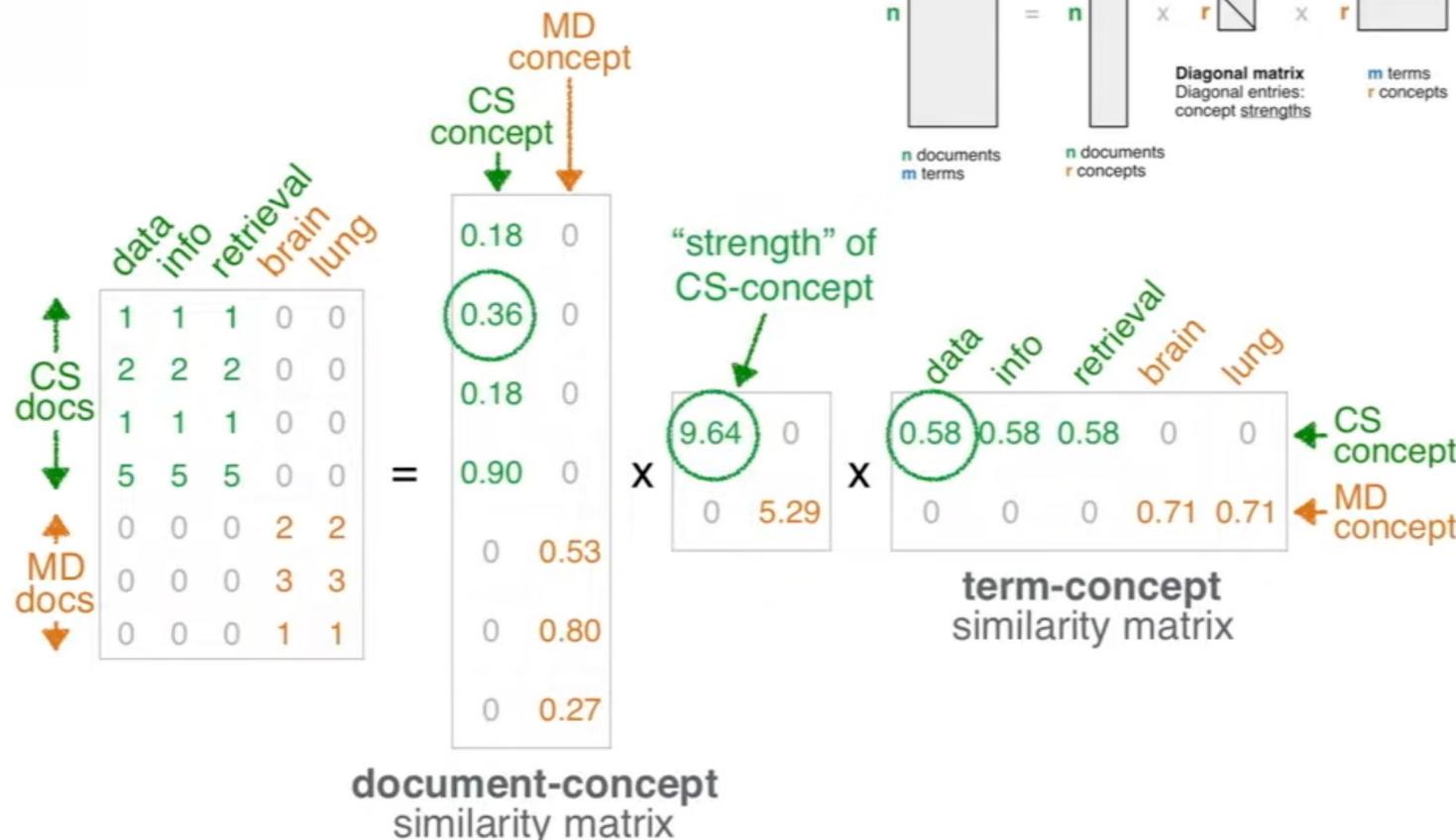
r : rank of the matrix; strength of each ‘concept’

V: m x r matrix

e.g., m terms, r concepts



Latent Semantic Analysis - Exemple



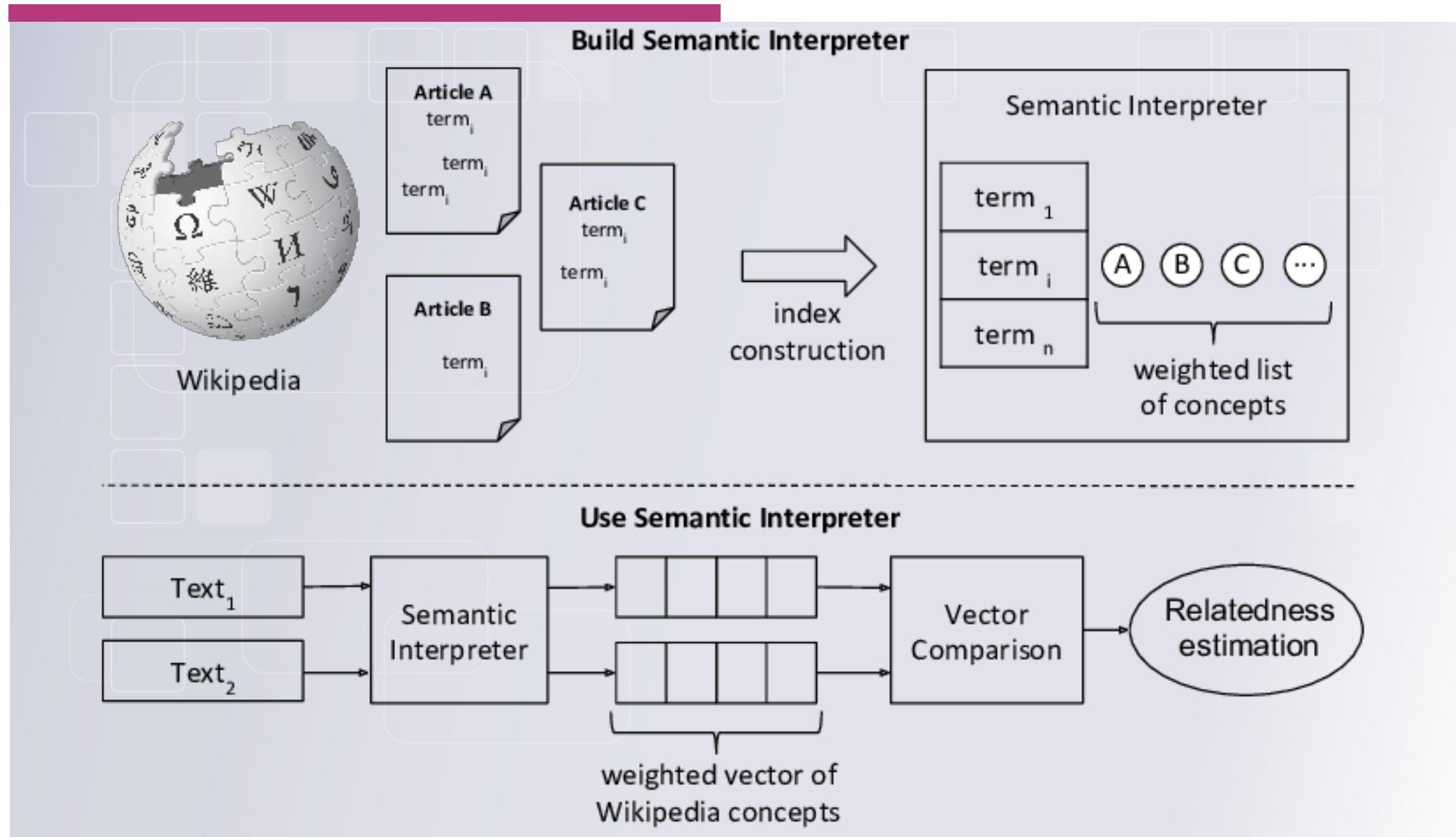
Explicit Semantic Analysis, ESA

- ❖ **Problème :** Représentations basées sur la matrice documents-termes très dépendantes de l'ensemble de documents utilisés
- ❖ **Idée de base :** Des termes seront plus proches s'ils sont utilisés ensemble dans plusieurs articles Wikipedia
- ❖ **Objectif :** recherche de « concepts », en (relativement) faible nombre, correspondant à des corrélations entre termes, pour représenter les documents d'une collection.

- Utiliser un corpus très grand (Wikipedia) pour construire la matrice documents-termes
- Représentations générales et riches des termes : chaque terme est représenté par un vecteur dont la dimension est donnée par le nombre de pages Wikipedia (*concepts*).
- Si le terme y est absent, la valeur est 0. Si le terme y est présent, la valeur est égale à sa pondération TF-IDF.
- Chaque document est représenté par le centre de gravité de l'ensemble des mots qu'il contient (pondérations TF-IDF prises en compte).

Contrairement aux résultats de LSA, avec ESA les dimensions des vecteurs sont interprétables (dans Wikipedia chaque document décrit un concept)

Explicit Semantic Analysis, ESA



Word Embedding

- ❖ Les représentations vectorielles précédentes sont basées sur la description d'un texte par l'ensemble des termes qu'il contient.
- ❖ Toute information liée au contexte des mots est ignorée.
- ❖ Le contexte d'un mot dans une phrase caractérise assez bien le mot à la fois sur l'aspect syntaxique et sur l'aspect sémantique.
- ❖ Il est alors utile d'exploiter le contexte pour construire des représentations vectorielles (**Word Embedding ou plongement lexical**) plus « raffinées ».
- ❖ Le **word embedding** opère en réduisant la dimension des représentations vectorielles « One Hot » des mots pour être capable de capturer le contexte, la similarité sémantique et syntaxique (genre, synonymes, ...) d'un mot.
- ❖ C'est une vectorisation des mots de sorte que les mots apparaissant dans des contextes similaires ont des significations apparentées.
- ❖ **Exemple :** on pourrait s'attendre à ce que les mots « remarquable » et « admirable » soient représentés par des vecteurs relativement peu distants dans le nouvel espace de représentations.

Word Embedding

Objectif

texte à apprendre : mon client est content de son assurance



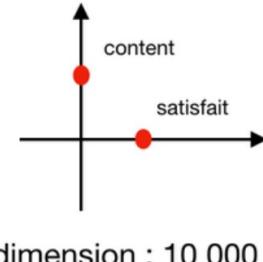
texte à prédire : l'assuré est satisfait de notre contrat



one-hot encoding

content satisfait

$$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 1 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$



word embedding

content satisfait

$$\begin{bmatrix} 0.35 \\ 0.1 \\ -1.1 \\ 0.1 \\ 0.1 \\ 0.9 \\ -1 \\ 2.1 \end{bmatrix} \quad \begin{bmatrix} 0.39 \\ 0.1 \\ -1.2 \\ 0.11 \\ 0.1 \\ 0.88 \\ -0.5 \\ 2 \end{bmatrix}$$



(bon compromis ~ entre 100 et 300)

Vecteur clairsemé

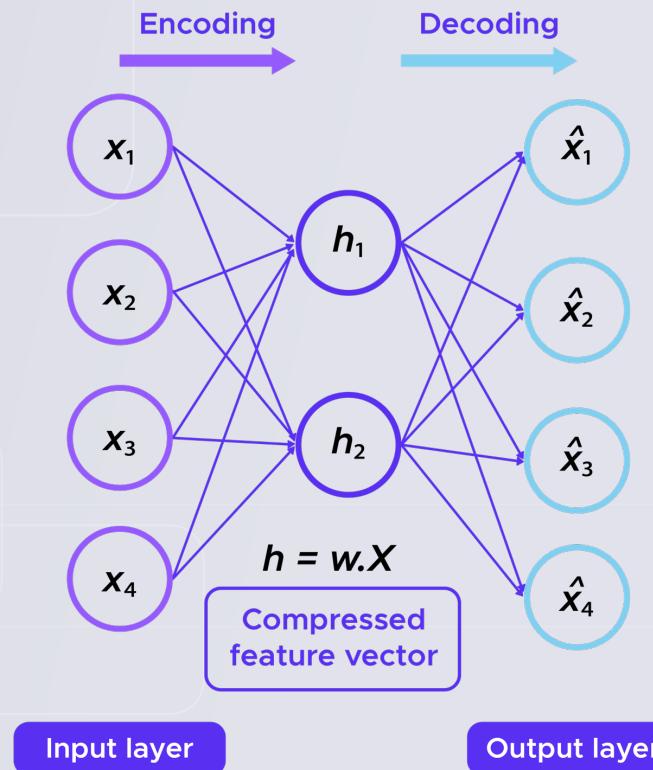
Vecteur dense

Word Embedding populaire : Word2Vec

- ❖ La méthode d'embedding généralement utilisée pour réduire la dimension d'un vecteur consiste à utiliser le résultat que retourne une couche dense :

Multiplier une matrice d'embedding W par la représentation « One hot » du mot

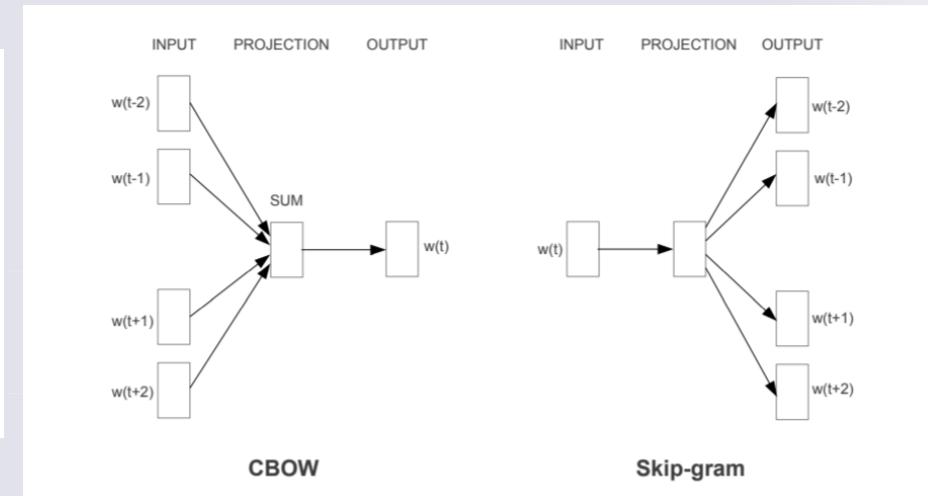
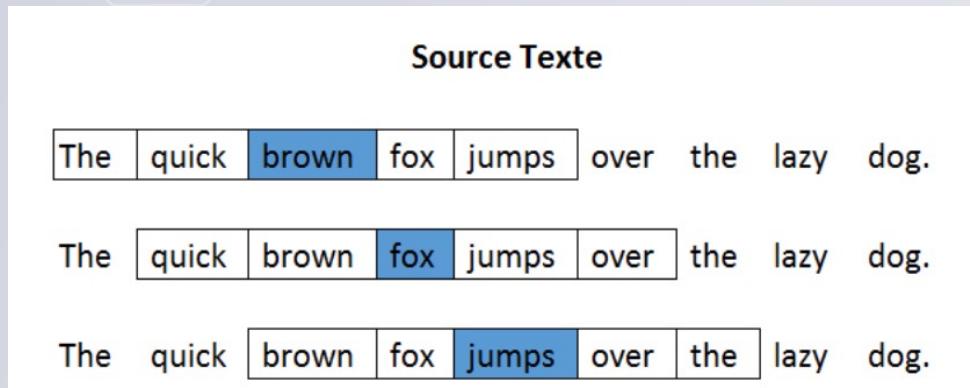
Encodeur-décodeur



Word2Vec

❖ Deux méthodes différentes utilisant des réseaux de neurones à 3 couches (1 couche d'entrée, 1 couche cachée, 1 couche de sortie) :

- Méthode continuous bag of words (CBOW) : prédire chaque mot à partir de son contexte
- Méthode Skip-gram : à partir de chaque mot prédire son contexte



Word2Vec

Exemple de texte :

notre client est content de son assurance automobile
notre client est content de son assurance automobile
client content assurance automobile

Générer des observations pour word2vec pour une fenêtre de taille 2

- | | cible | contexte |
|-----------------|--------|------------------------------|
| observation 1 : | client | content assurance automobile |
| observation 2 : | client | content assurance automobile |
| observation 3 : | client | content assurance automobile |
| observation 4 : | client | content assurance automobile |

Modèle CBOW :

cherche à prédire un mot à partir du contexte

client content ? automobile

contexte cible contexte



réseaux de neurones

Modèle skip-gram :

cherche à prédire les mots du contexte à partir d'un mot

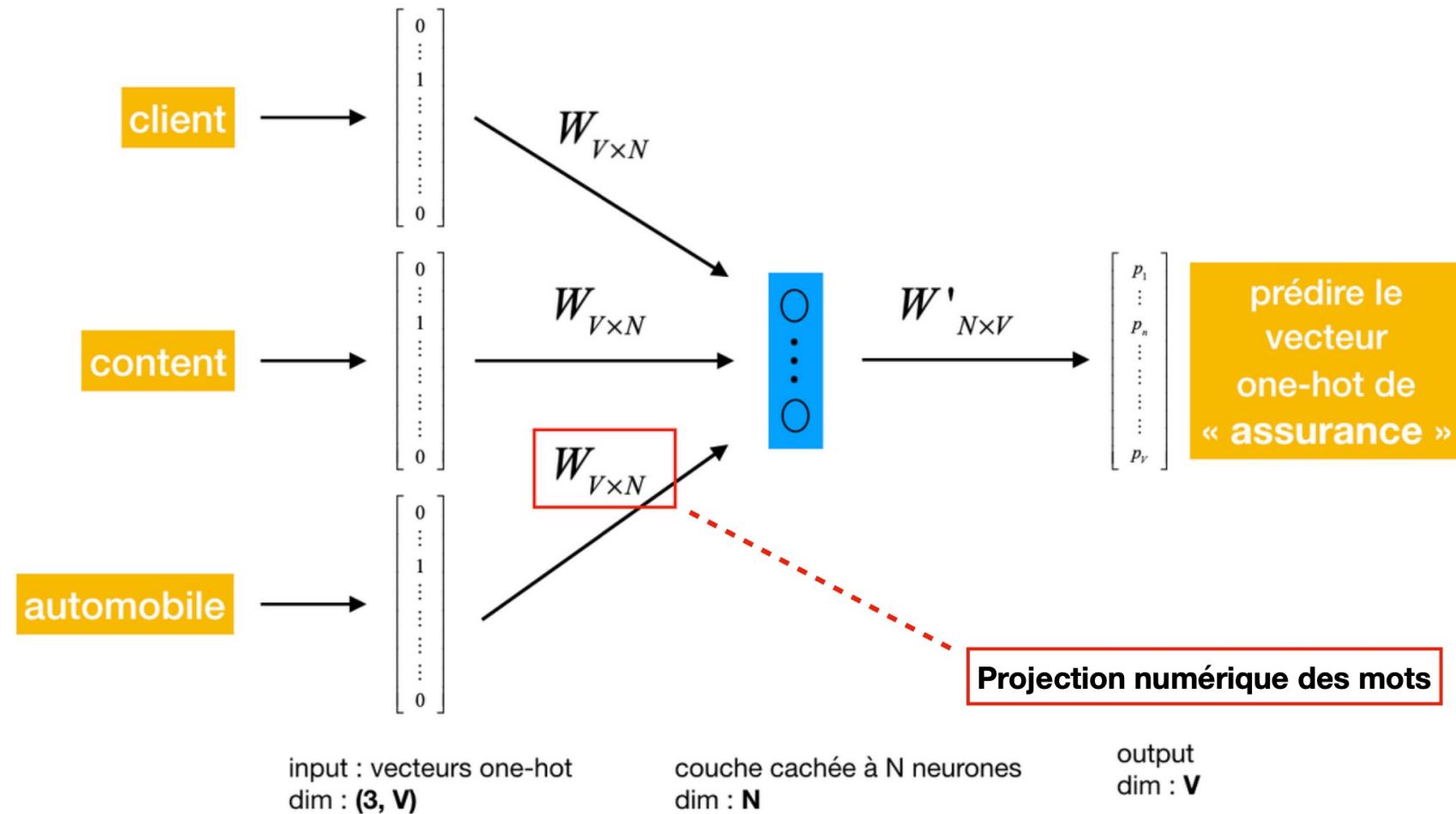
? ? assurance ?

contexte cible contexte

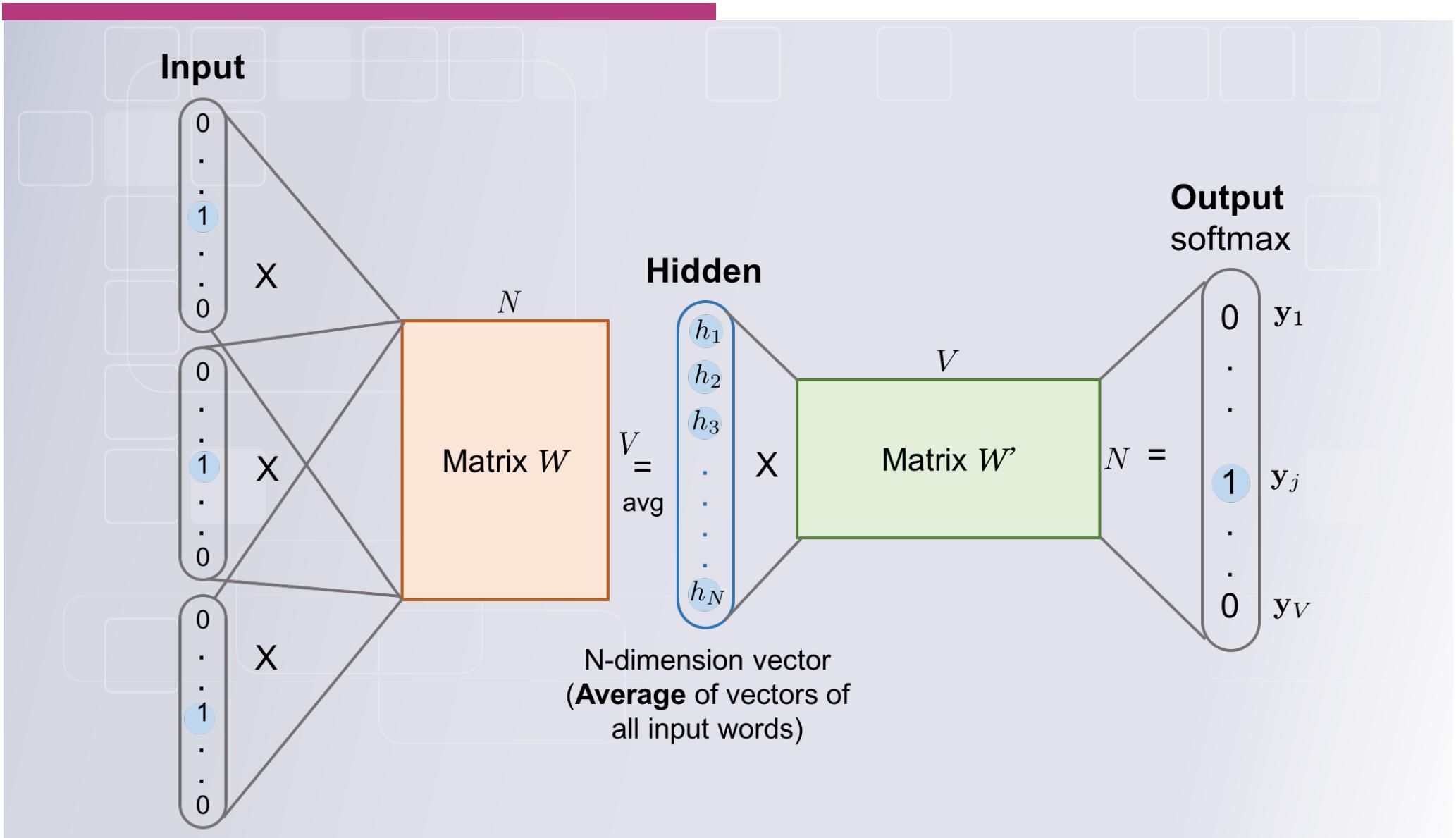


réseaux de neurones
client content automobile

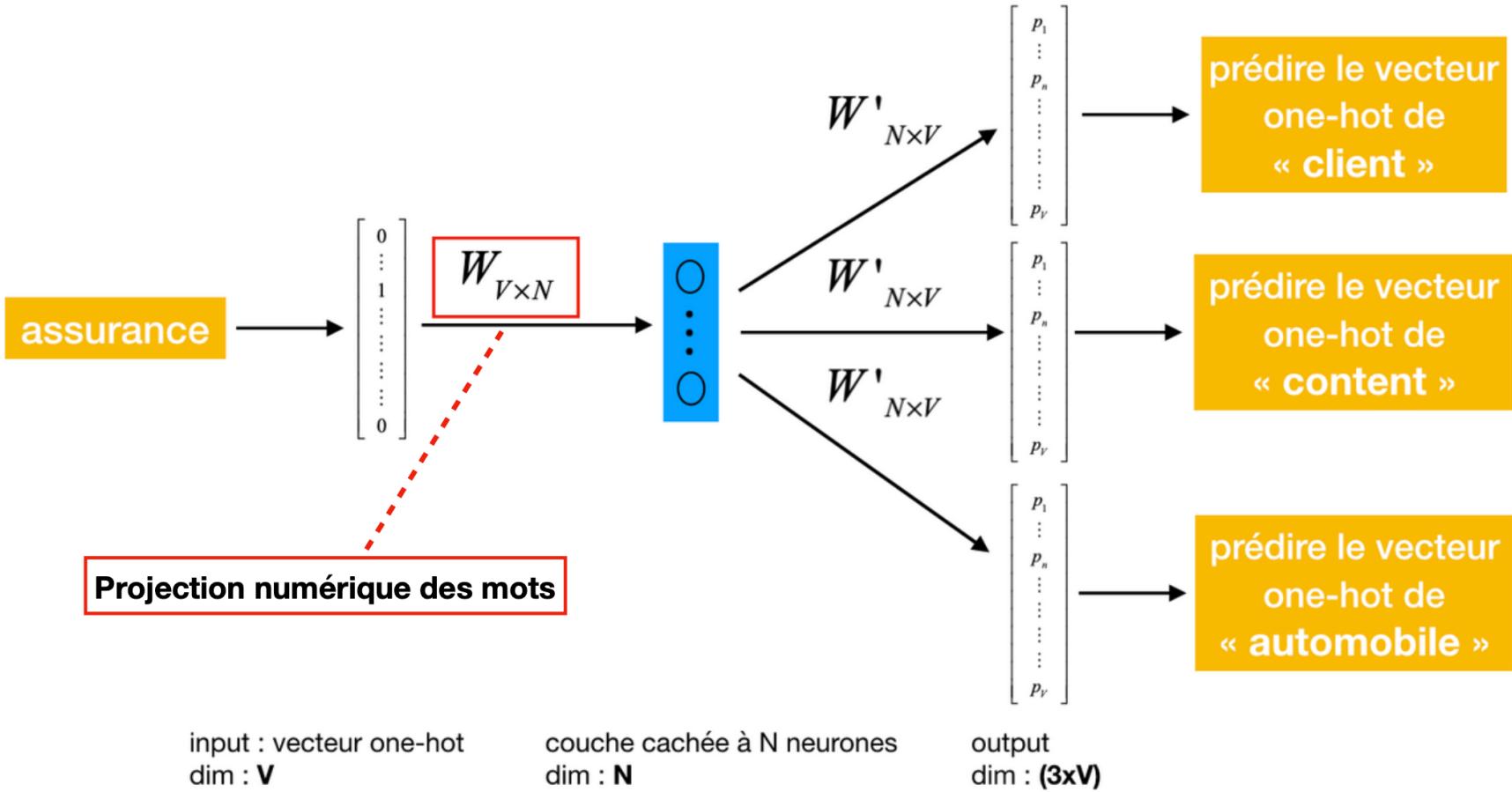
CBOW



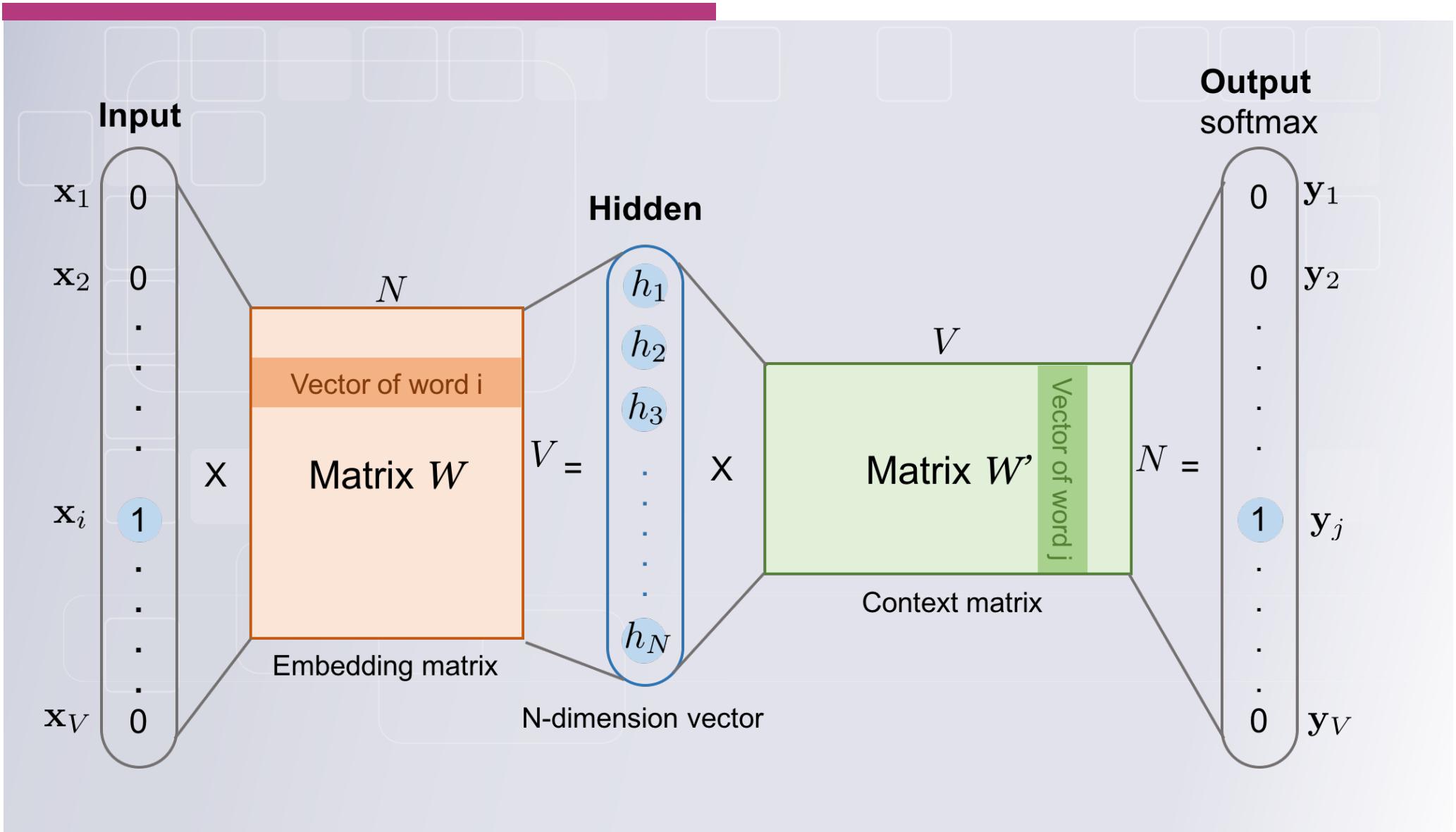
CBOW



SKIP-GRAM

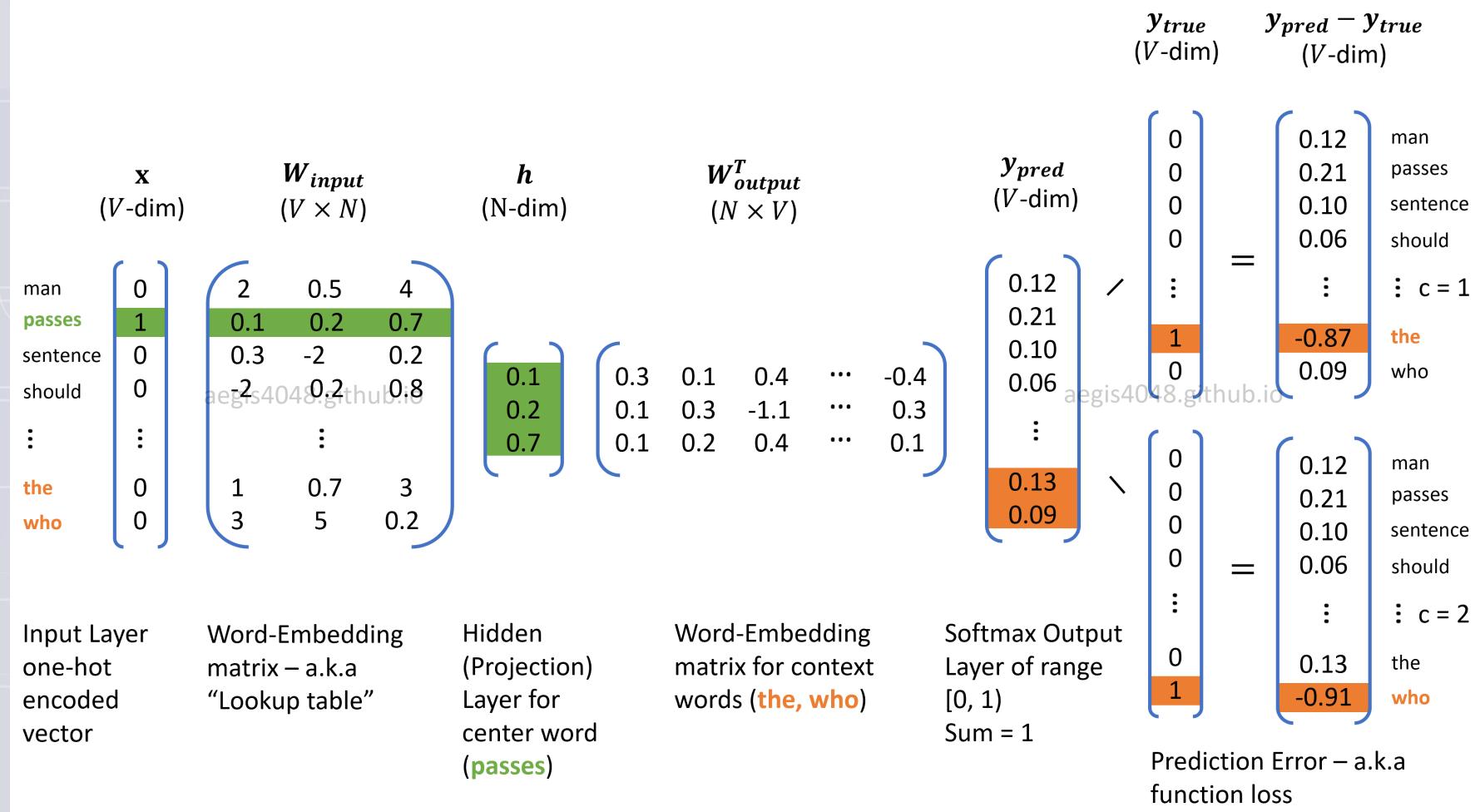


SKIP-GRAM



SKIP-GRAM

The man who passes the sentence should swing the sword.



SKIP-GRAM

The man who passes the sentence should swing the sword.

$$\begin{array}{ll}
 \mathbf{y}_{true} & \mathbf{y}_{pred - y_{true}} \\
 (V\text{-dim}) & (V\text{-dim})
 \end{array}$$

$$\mathbf{y}_{pred} \quad \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0.12 \\ 0.21 \\ 0.10 \\ 0.06 \\ \vdots \\ 0.13 \\ 0.09 \end{pmatrix} \quad \begin{pmatrix} 0.12 \\ 0.21 \\ 0.10 \\ 0.06 \\ \vdots \\ -0.87 \\ 0.09 \end{pmatrix} = \begin{matrix} \text{man} \\ \text{passes} \\ \text{sentence} \\ \text{should} \\ \vdots \\ \text{the} \\ \text{who} \end{matrix} \quad \text{aegis4048.github.io}$$

$$\mathbf{y}_{pred} \quad \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 0.12 \\ 0.21 \\ 0.10 \\ 0.06 \\ \vdots \\ 0.13 \\ 0.09 \end{pmatrix} = \begin{matrix} \text{man} \\ \text{passes} \\ \text{sentence} \\ \text{should} \\ \vdots \\ \text{the} \\ \text{who} \end{matrix} \quad \text{c} = 1$$

$$\mathbf{y}_{pred} \quad \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 0.12 \\ 0.21 \\ 0.10 \\ 0.06 \\ \vdots \\ 0.13 \\ 0.09 \end{pmatrix} = \begin{matrix} \text{man} \\ \text{passes} \\ \text{sentence} \\ \text{should} \\ \vdots \\ \text{the} \\ \text{who} \end{matrix} \quad \text{c} = 2$$

$$\sum_{c=1}^C e_c = \quad \begin{matrix} c = 1 \\ c = 2 \end{matrix} \quad \begin{pmatrix} 0.12 \\ 0.21 \\ 0.10 \\ 0.06 \\ \vdots \\ -0.87 \\ 0.09 \end{pmatrix} + \begin{pmatrix} 0.12 \\ 0.21 \\ 0.10 \\ 0.06 \\ \vdots \\ 0.13 \\ -0.91 \end{pmatrix} = \begin{pmatrix} 0.24 \\ 0.42 \\ 0.20 \\ 0.12 \\ \vdots \\ -0.74 \\ -0.82 \end{pmatrix} = \begin{matrix} \text{man} \\ \text{passes} \\ \text{sentence} \\ \text{should} \\ \vdots \\ \text{the} \\ \text{who} \end{matrix}$$

$$\sum_{c=1}^C e_c = \quad \begin{matrix} iter = 1 \\ iter = 2 \\ iter = 3 \\ iter = 4 \end{matrix} \quad \begin{pmatrix} 0.24 \\ 0.42 \\ 0.20 \\ 0.12 \\ \vdots \\ -0.74 \\ -0.82 \end{pmatrix} \xrightarrow{\text{Update } \theta} \begin{pmatrix} 0.18 \\ 0.30 \\ 0.12 \\ 0.09 \\ \vdots \\ -0.32 \\ -0.46 \end{pmatrix} \xrightarrow{\text{Update } \theta} \begin{pmatrix} 0.08 \\ 0.12 \\ 0.04 \\ 0.02 \\ \vdots \\ -0.12 \\ -0.13 \end{pmatrix} \xrightarrow{\text{Update } \theta} \begin{pmatrix} 0.01 \\ 0.02 \\ 0.01 \\ 0.00 \\ \vdots \\ -0.01 \\ -0.02 \end{pmatrix} = \begin{matrix} \text{man} \\ \text{passes} \\ \text{sentence} \\ \text{should} \\ \vdots \\ \text{the} \\ \text{who} \end{matrix}$$

SKIP-GRAM

The man who passes the sentence should swing the sword.

$$\begin{array}{c} \mathbf{x} \\ \text{(V-dim)} \end{array} \quad \begin{array}{c} W_{input} \\ \text{(V} \times N) \end{array} \quad \begin{array}{c} \mathbf{h} \\ \text{(N-dim)} \end{array}$$

man 0 \times 2 0.5 4
passes 1 0.1 0.2 0.7
sentence 0 0.3 -2 0.2
should 0 -2 0.2 0.8
⋮ ⋮
the 0 1 0.7 3
who 0 3 5 0.2

=

0.1
0.2
0.7

Projection of word vector for “passes” from the embedding matrix

Word2Vec

- ❖ CBOW est moins coûteuse mais Skip-gram donne en général de meilleurs résultats (surtout lorsque le corpus de textes utilisé n'est pas très grand)
- ❖ Avec cette représentation, les mots se regroupent par similarité de contexte
- ❖ L'objectif est de représenter les termes d'un corpus à l'aide d'un vecteur de taille k (paramètre à définir, parfois des centaines, tout dépend de la quantité des documents), où ceux qui apparaissent dans des contextes similaires (taille du voisinage V , paramètre à définir) sont proches (au sens de la distance cosinus par exemple).
- ❖ Une forme d'additivité :
$$\text{vec(Madrid)} - \text{vec(Spain)} + \text{vec(France)} = \text{vec(Paris)}$$
- ❖ Il existe des modèles pré-entraînés sur des documents (qui font référence, ex. Wikipedia ; en très grande quantité) que l'on peut directement appliquer sur nos données ([Wikipedia2vec](#) et [Google Word2Vec](#))

Word2Vec

❖ Il existe des méthodes qui construisent, comme Word2Vec, pour chaque mot un plongement lexical :

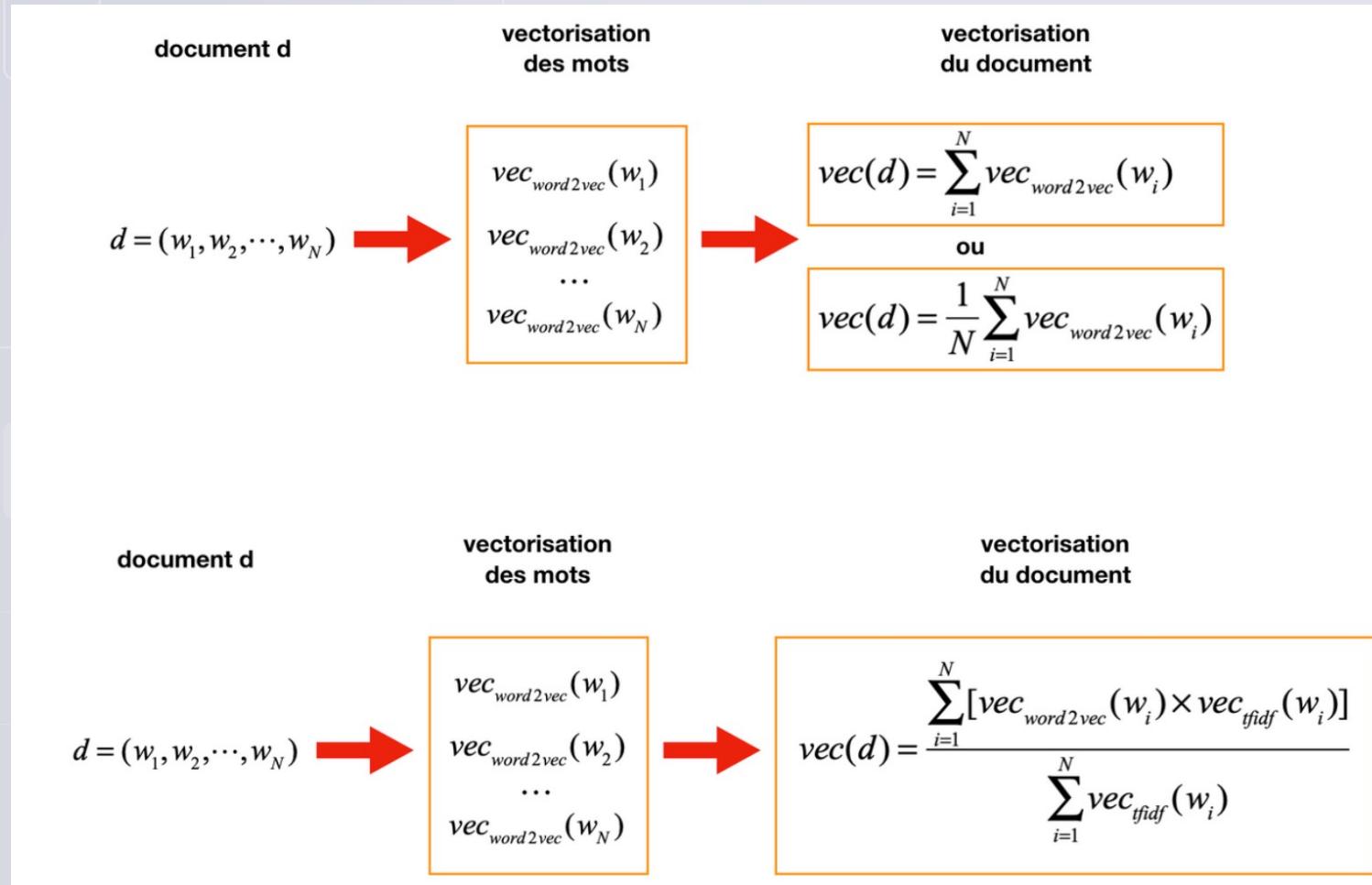
- GloVe** : Comme Word2Vec, permet d'obtenir des word embeddings seulement pour les mots rencontrés dans les textes du corpus traité (mots du vocabulaire)
- FastText** : vise à résoudre le problème des mots hors vocabulaire
 - Méthode similaire à Word2Vec, sauf que l'unité de base n'est pas le mot mais le n-gramme de caractères
 - Hypothèses :
 - La combinaison des embeddings des n-grammes est un bon embedding d'un mot,
 - Les mots hors vocabulaire sont composés de n-grammes rencontrés dans les mots du corpus
 - L'embedding d'un mot est obtenu à partir des embeddings de ses n-grammes (par ex. pour le mot **vecteur** les trigrammes sont {vec,ect,cte,teu,eur})

Des termes aux documents : vecteur de représentation

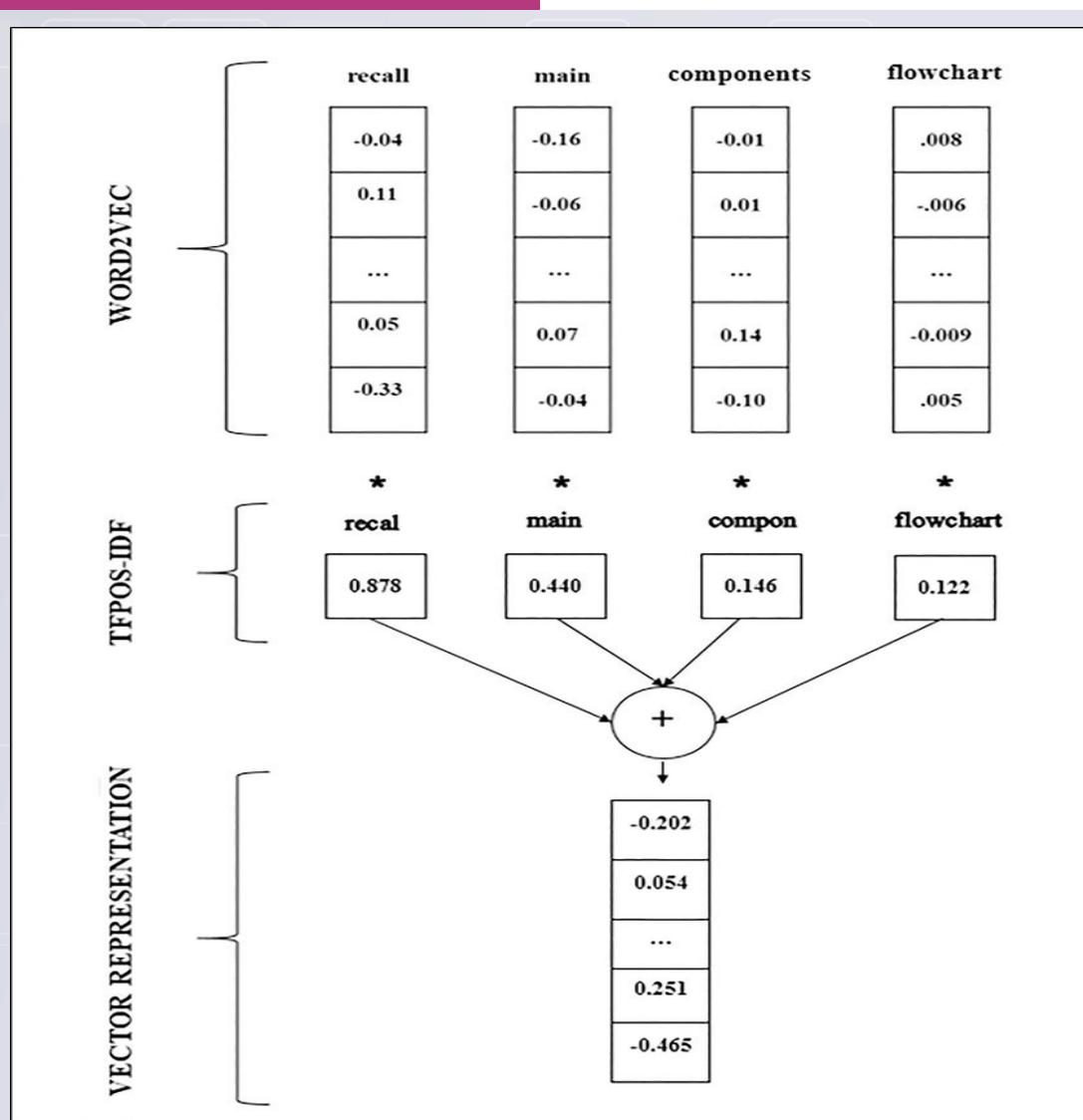
- ❖ Disposer d'une représentation des documents dans le nouvel espace est indispensable pour pouvoir appliquer les algorithmes de machine learning
- ❖ Texte court ou petit ensemble de mots clés (< 10 – 15 termes):
 - Le centre de gravité des vecteurs représentant les mots en général en supprimant les stop words et parfois en pondérant les mots (par ex. pondérations TF-IDF) constitue une solution facile
- ❖ Texte plus long :
 - Centre de gravité des vecteurs représentant les mots : centre de gravité devient peu spécifique (nombre élevé de mots) donc peu discriminant ⇒ approche à éviter
 - Méthodes possibles : Doc2Vec, Utilisation du LSTM

Des termes aux documents : vecteur de représentation

Comment vectoriser un document ?



Des termes aux documents : vecteur de représentation

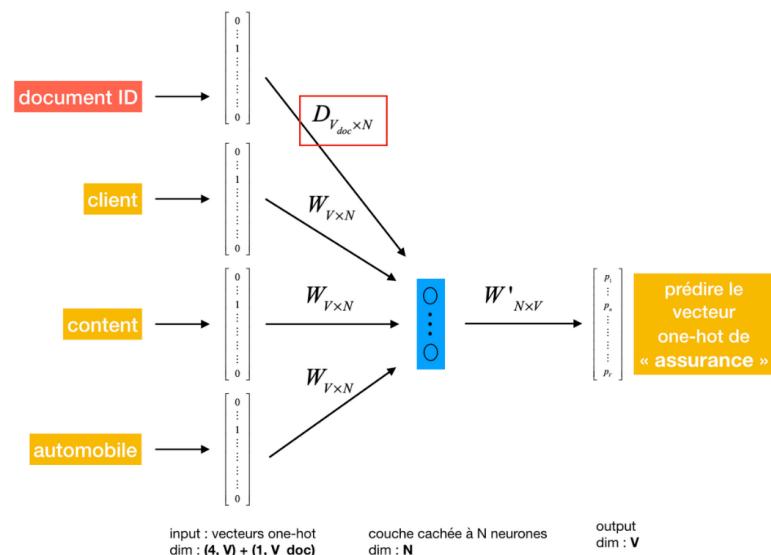


Doc2Vec

❖ **Principe : vectorisation des documents (resp. mots) de sorte que les documents (resp. mots) apparaissant dans des contextes similaires ont des significations apparentées.**

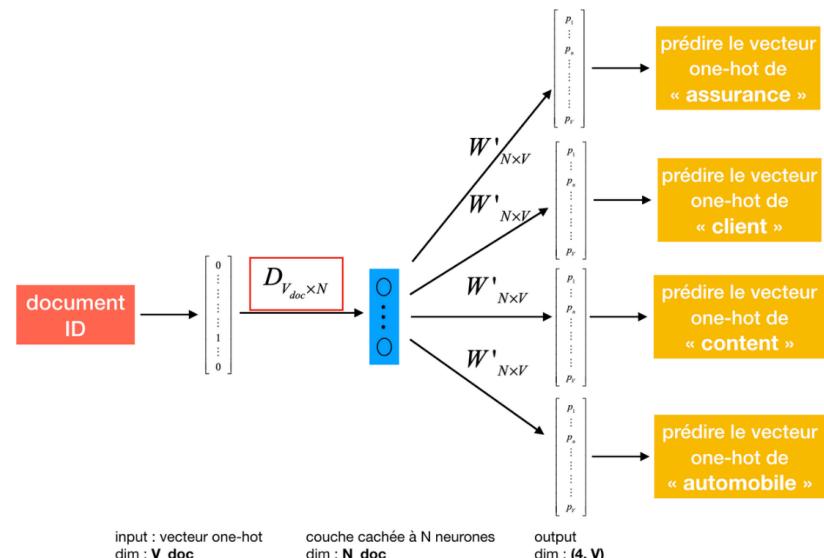
Distributed Memory (DM) :

$$p(w_i | w_{i-h}, \dots, w_{i+h}, d)$$



Distributed Bag-Of-Words (DBOW) :

$$p(w_{i-h}, \dots, w_{i+h} | d)$$



Installation

❖ Keras

❖ Python NLP libraries : NLTK, SPACY, Gensim,
TextBlob

❖ Google Pre-trained Embedding :

<https://drive.google.com/file/d/0B7XkCwpI5KDYNINUTTlSS21pQmM/edit?usp=sharing>

❖ News Category Dataset (HuffPost)

<https://www.kaggle.com/datasets/yazansalameh/news-category-dataset-v2>