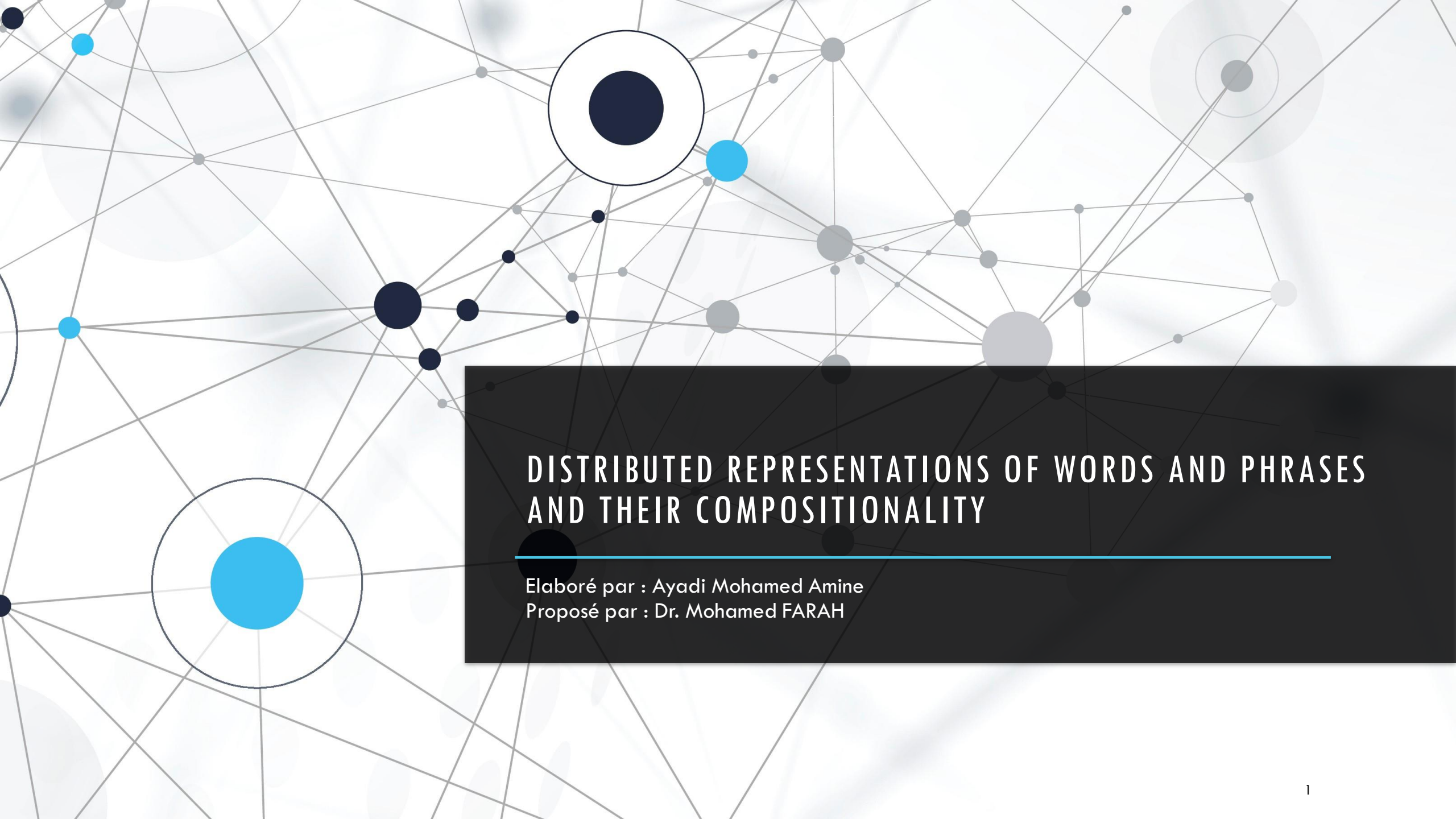


DISTRIBUTED REPRESENTATIONS OF WORDS AND PHRASES AND THEIR COMPOSITIONALITY

Elaboré par : Ayadi Mohamed Amine
Proposé par : Dr. Mohamed FARAH

PLAN

1. Prise en compte du contexte
2. Les algorithmes SKIP-GRAM et CBOW
3. Bibliographie

Représentation de documents en sac de mots (bag of words)

La représentation en sac de mots ne tient pas compte des positions relatives de mots dans les documents.

(0) condition du bien etre
(1) etre important
(2) solution bien etre
(3) important bien etre

	bien	condition	du	etre	important	solution
(0)	1	1	1	1	0	0
(1)	0	0	0	1	1	0
(2)	1	0	0	1	0	1
(3)	1	0	0	1	1	0

La contextualisation de « bien » par « être » (ou l'inverse d'ailleurs) est importante, ils sont souvent « voisins ». La représentation en sac de mots passe à côté (les algos de machine learning ne verront que la co-occurrence, c'est déjà pas mal - ex. topic modeling)

Idée du prolongement lexical – Word embedding

Idée du prolongement lexical : déterminer une représentation des termes par un vecteur numérique de dimension K (paramétrable), en tenant compte de son contexte (fenêtre de voisinage V dont la taille est paramétrable).

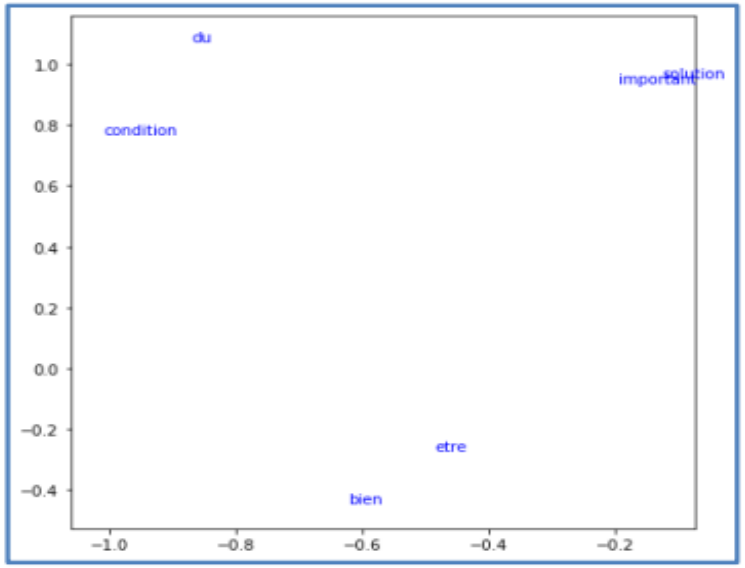
```
(0) condition du bien etre
(1) etre important
(2) solution bien etre
(3) important bien etre
```

Les termes apparaissant dans des contextes similaires sont proches (au sens de la distance entre les vecteurs de description).

A partir de la représentation des termes qui les composent, il est possible de dériver une description numérique (vectorielle) des documents.

Un exemple – Représentation dans le plan (K = 2), fenêtre de voisinage pris en compte (V = 1)

(0) condition du bien etre
(1) etre important
(2) solution bien etre
(3) important bien etre

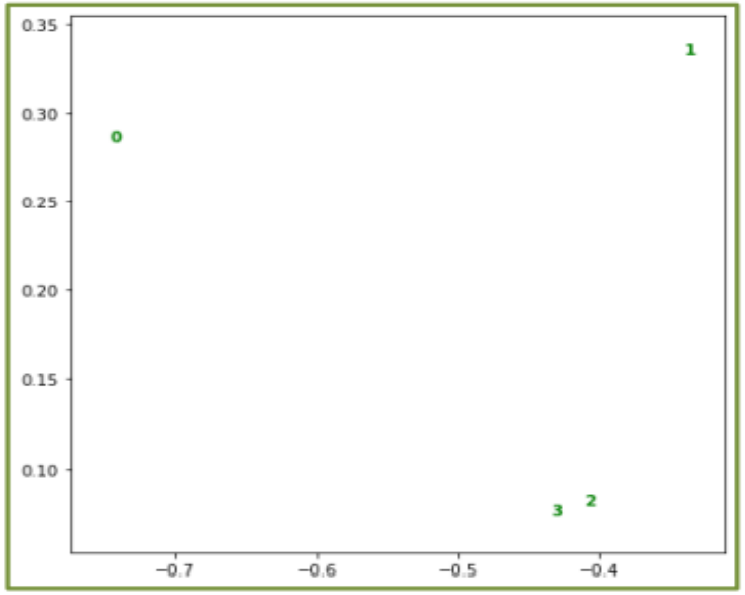


Représentation des termes

Bon, converger sur 4 observations n'est pas évident quoiqu'il en soit

(A) K << taille (dictionnaire), réduction forte de la dimensionnalité

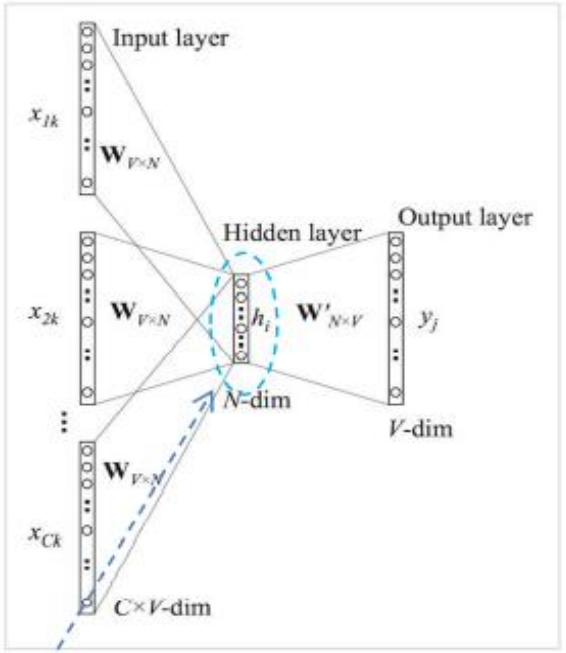
(B) Il est possible d'effectuer des traitements de machine learning à partir de ce nouvel espace de représentation (clustering, classement,...)



Représentation des documents

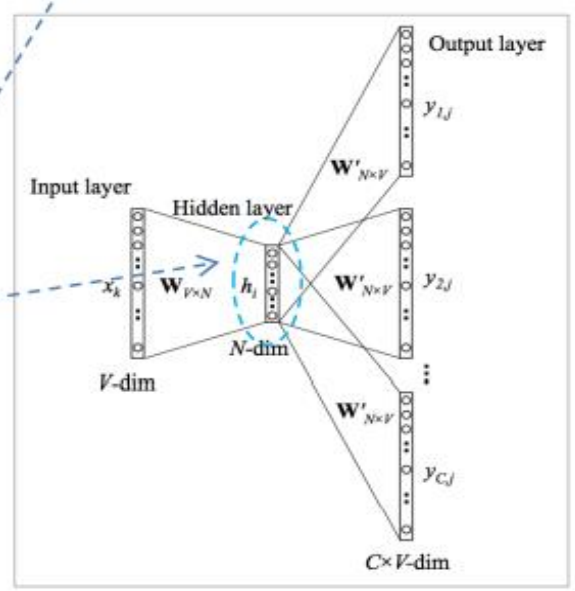
Idée de word2vec

Modéliser les termes en utilisant un réseau de neurones (un perceptron à une couche cachée) avec en entrée le contexte (le voisinage) et en sortie le terme (CBOW) ou inversement (SKIP-GRAM).



CBOW

A la manière des auto-encodeurs, ce sont les descriptions à la sortie de la couche cachée qui nous intéressent (nouvelles coordonnées des termes). Elles constituent la représentation des termes dans un nouvel espace.

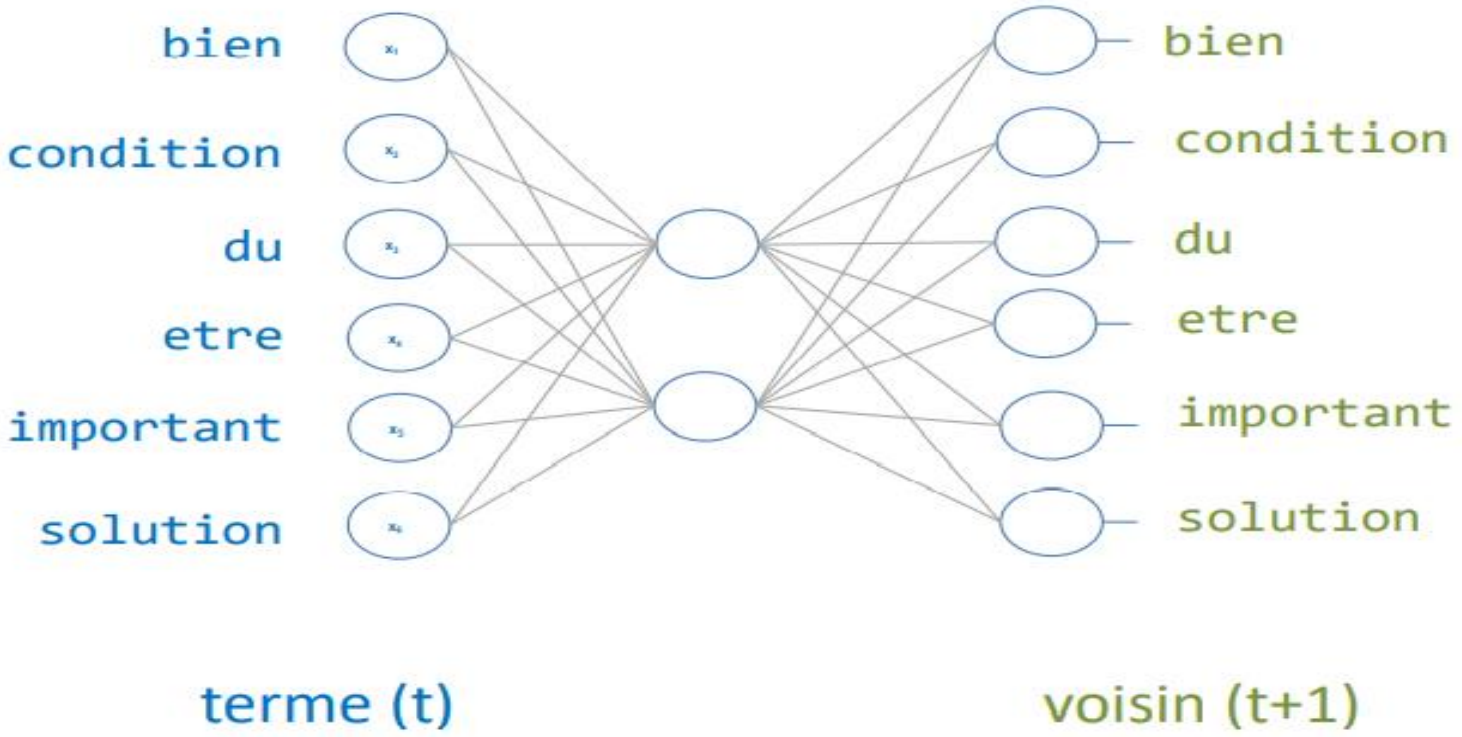


SKIP-GRAM

Modèle SKIP-GRAM

Modéliser les voisinages à partir des termes c.-à-d. P(voisin[s] / terme).

Ex. le voisin immédiat qui succèdent les termes dans les documents



Modèle SKIP-GRAM - Encodage des données tenant compte du voisinage

L'astuce passe par un encodage approprié des données tenant compte du voisinage. Ex. voisinage de taille 1 vers l'avant v_{t+1}

Description BOW (bag of words)

	bien	condition	du	etre	important	solution
condition du bien etre	1	1	1	1	0	0
etre important	0	0	0	1	1	0
solution du bien etre	1	0	0	1	0	1
important bien etre	1	0	0	1	1	0

Entrée (terme t)

Terme	bien	condition	du	etre	important	solution
condition	0	1	0	0	0	0
du	0	0	1	0	0	0
du	0	0	1	0	0	0
bien	1	0	0	0	0	0
bien	1	0	0	0	0	0
bien	1	0	0	0	0	0
etre	0	0	0	1	0	0
important	0	0	0	0	1	0

etc.

Sortie (voisin t + 1)

Terme	bien	condition	du	etre	important	solution
du	0	0	1	0	0	0
bien	1	0	0	0	0	0
bien	1	0	0	0	0	0
etre	0	0	0	1	0	0
etre	0	0	0	1	0	0
etre	0	0	0	1	0	0
important	0	0	0	0	1	0
bien	1	0	0	0	0	0

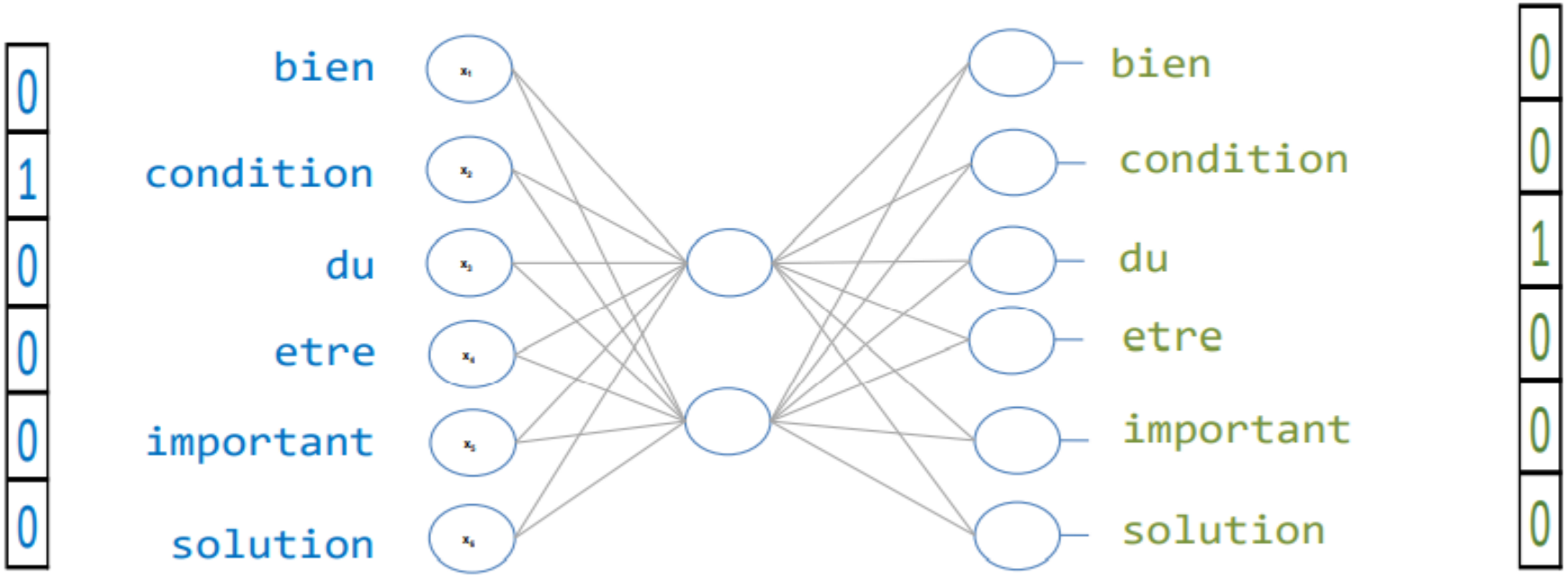
etc.

Ce sont ces données (pondération forcément binaire) que l'on présentera au réseau.

On est dans un (une sorte de) schéma d'apprentissage supervisé multi-cibles

Modèle SKIP-GRAM (Mikolov et al., 2013)

Exemple d'une observation présentée au réseau : (condition → du)



terme (t)

voisin (t+1)

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

SKIP-GRAM – Prise en compte du voisinage (t-1) et (t+1)

Double tableau pour la sortie

maintenant : voisinages (t-1) et (t+1)

	bien	condition	du	etre	important	solution
condition du bien etre	1	1	1	1	0	0
etre important	0	0	0	1	1	0
solution du bien etre	1	0	0	1	0	1
important bien etre	1	0	0	1	1	0

Entrée (terme t)

Terme	bien	condition	du	etre	important	solution
du	0	0	1	0	0	0
du	0	0	1	0	0	0
bien	1	0	0	0	0	0
bien	1	0	0	0	0	0
bien	1	0	0	0	0	0

etc.

Sortie (voisin t -1)

Terme	bien	condition	du	etre	important	solution
condition	0	1	0	0	0	0
solution	0	0	0	0	0	1
du	0	0	1	0	0	0
du	0	0	1	0	0	0
important	0	0	0	0	1	0

etc.

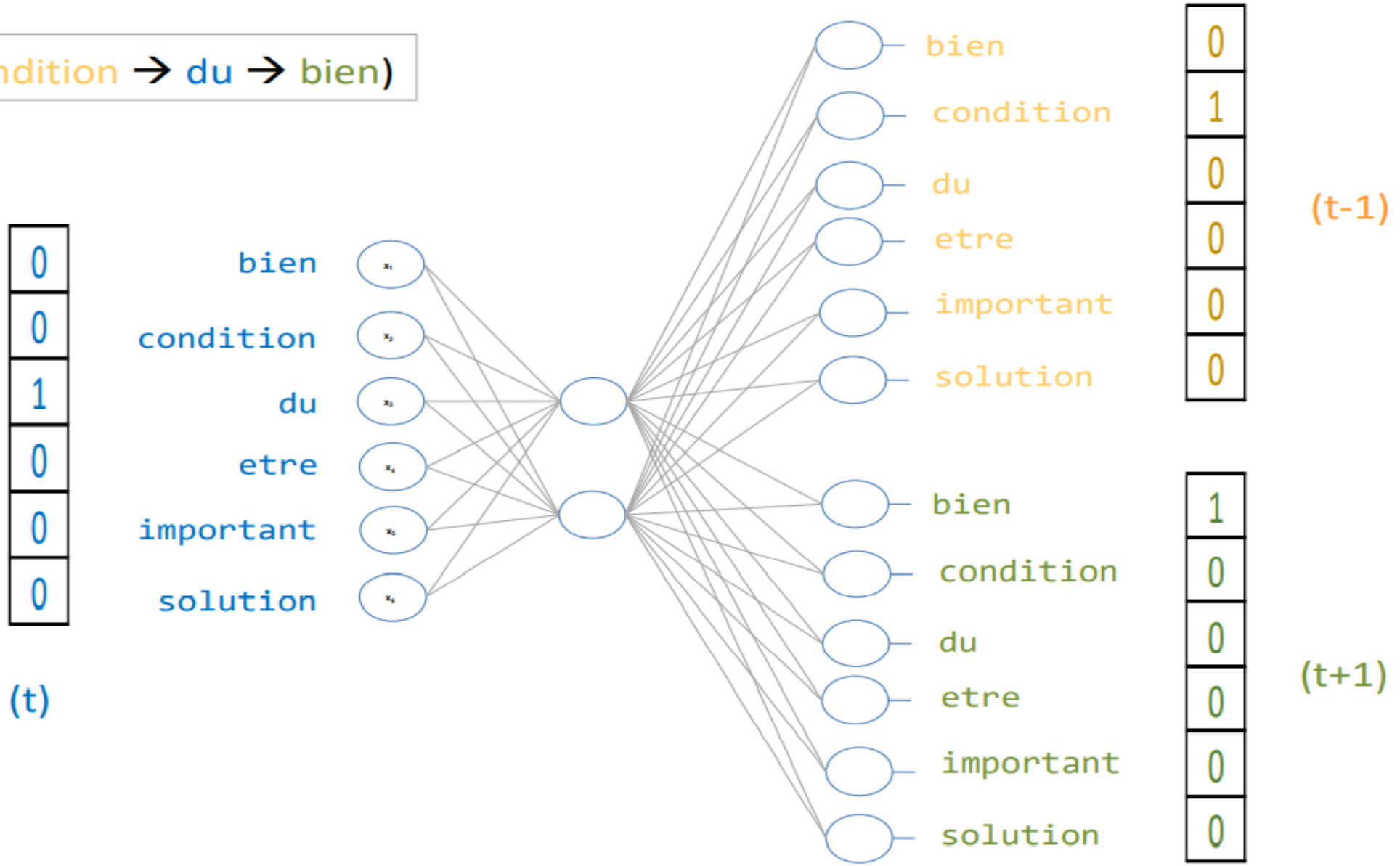
Sortie (voisin t + 1)

Terme	bien	condition	du	etre	important	solution
bien	1	0	0	0	0	0
bien	1	0	0	0	0	0
etre	0	0	0	1	0	0
etre	0	0	0	1	0	0
etre	0	0	0	1	0	0

etc.

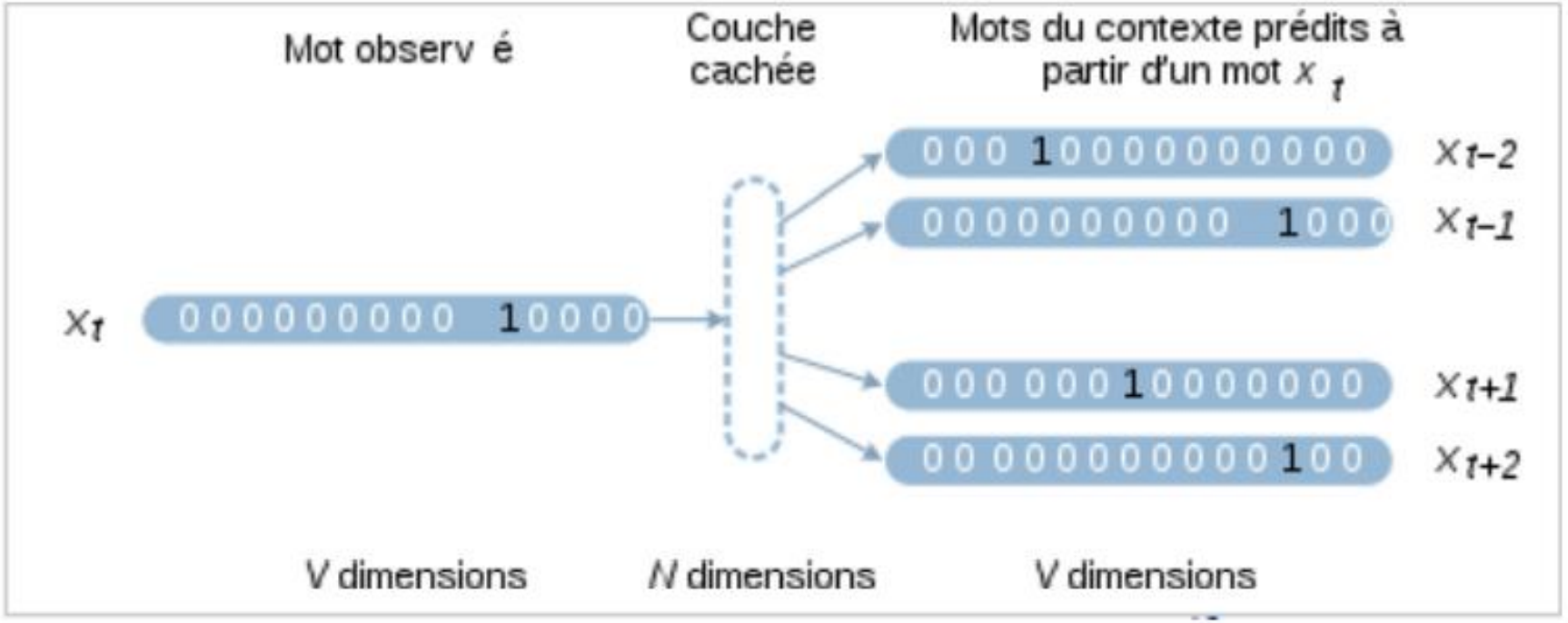
SKIP-GRAM – Prise en compte du voisinage (t-1) et (t+1) – Structure du réseau

Ex. (condition → du → bien)



Modèle SKIP-GRAM – Voisinage plus étendu

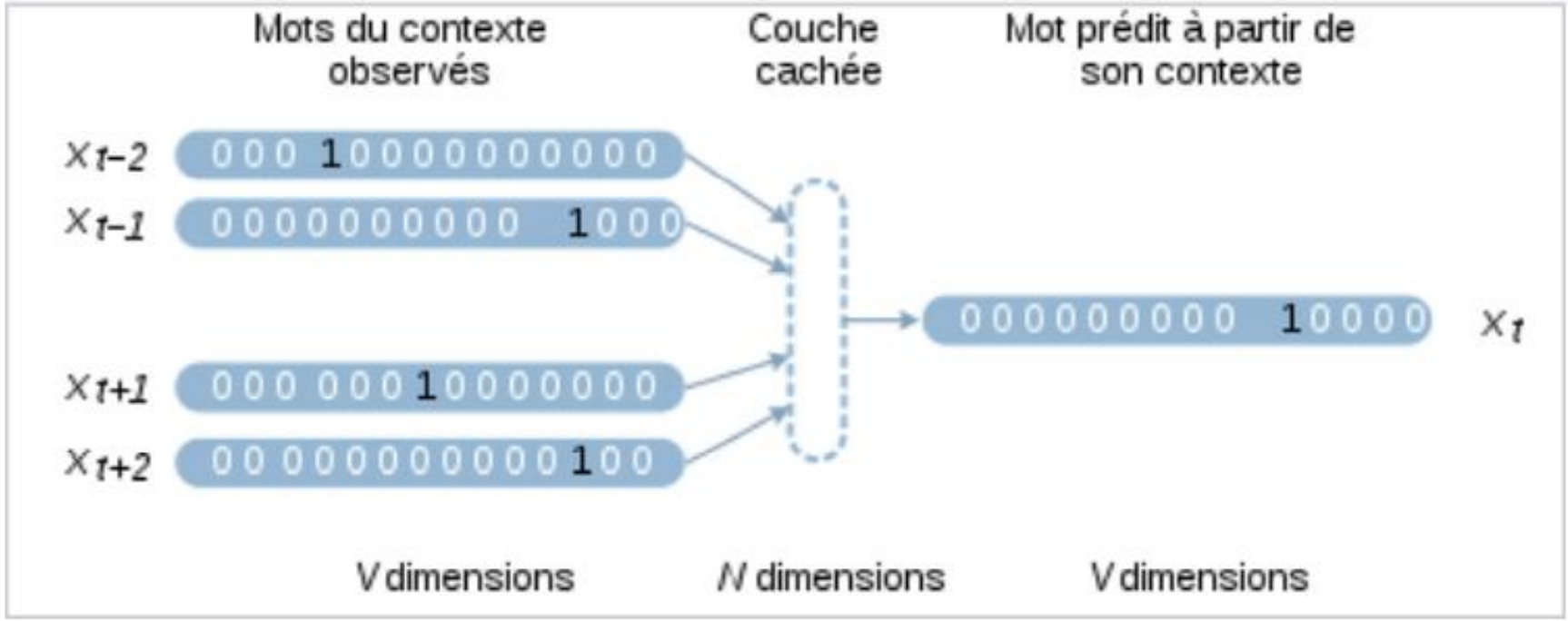
Il est possible de prendre un voisinage plus étendu (V = 2 ou plus). Attention simplement à la dilution de l'information.



https://fr.wikipedia.org/wiki/Word_embedding

Modèle CBOW – Continuous Bag-of-Words

La problématique est inversée : on s'appuie sur le voisinage (le contexte) pour apprendre les termes. On modélise P(terme / voisin[s]).



https://fr.wikipedia.org/wiki/Word_embedding

Word2Vec – Quelques éléments techniques

- La fonction de transfert pour la couche centrale est linéaire
- Pour la couche de sortie, la fonction de transfert est softmax
- La « negative log-likelihood » fait office de fonction de perte (à la place du classique MSE – mean squared error). En conjonction avec softmax, le calcul du gradient en est largement simplifiée lors de l'optimisation
- Dixit la documentation de H20 (fameux package de Deep Learning) : skip-gram donne plus de poids aux voisins proches, elle produit de meilleurs résultats pour les termes peu fréquents

Conclusion – word2vec

• « word2vec » est un technique de « word embedding » basée sur un algorithme de deep learning.

• 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 dist. cosinus par ex.).

• De la description des termes, nous pouvons dériver une description des documents, toujours dans un espace de dimension K. Possibilité d'appliquer des méthodes de machine learning par la suite (ex. catégorisation de documents).

• K << taille du dictionnaire : nous sommes bien dans la réduction de la dimensionnalité (par rapport à la représentation « bag-of-words » par ex.).

• Il existe des modèles pré-entrainés sur des documents (qui font référence, ex. Wikipedia ; en très grande quantité) que l'on peut directement appliquer sur nos données (ex. Google Word2Vec ; Wikipedia2Vec)

Références - Bibliographie

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space." In Proceedings of Workshop at ICLR. (Sep 2013)

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. "Distributed Representations of Words and Phrases and their Compositionality." In Proceedings of NIPS. (Oct 2013)

H20, "Word2Vec", http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/word2vec.html