

Word embedding

UV LO17

Fatma Chamekh

Fatma.chamekh@utc.fr

Objectifs

- Introduction à la sémantique distributionnelle
- Introduction aux word embeddings
- Utilisation de divers outils de développement
- Manipulation des words embeddings (entraîner un modèle, interroger un modèle etc.)

Sémantique Distributionnelle

Sémantique distributionnelle

Des mots apparaissant dans contextes similaires ont des sens proches (Harris, 1954)

"You shall know a word by the company it keep", (Firth, 1957)

Principe de l'approche distributionnelle

(6.1) Ongchoi is delicious sauteed with garlic.

(6.2) Ongchoi is superb over rice.

(6.3) ...ongchoi leaves with salty sauces...

(6.4) ...spinach sauteed with garlic over rice...

(6.5) ...chard stems and leaves are delicious...

(6.6) ...collard greens and other salty leafy greens

Speech and Language Processing, Jurafsky & Martin, 3rd ed. draft

Modèles distributionnels

- Représentation du sens des mots sous forme de vecteurs (= liste de nombres)

Modèles à base de fréquences

- Construits à partir des fréquences des cooccurrences
- Vecteurs peu denses et très gros

	aardvark	...	computer	data	pinch	result	sugar	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Figure 6.5 Co-occurrence vectors for four words, computed from the Brown corpus, showing only six of the dimensions (hand-picked for pedagogical purposes). The vector for the word *digital* is outlined in red. Note that a real vector would have vastly more dimensions and thus be much sparser.

Modèles distributionnels

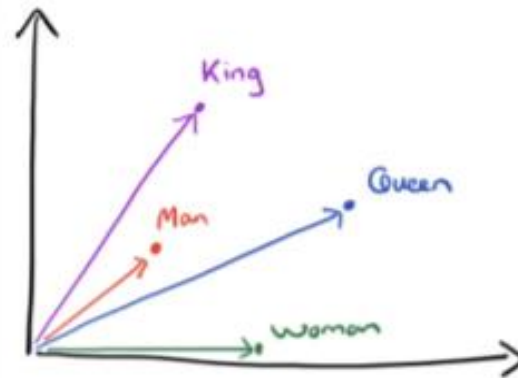
Modèles prédictifs / dense vectors / word embeddings / plongements de mots

- Vecteurs denses avec peu de dimensions
- Plus facile à intégrer dans des modèles de deep learning

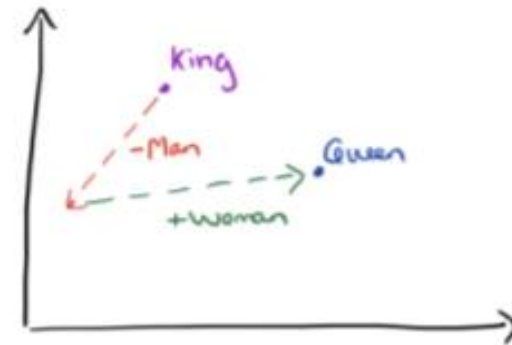


<https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/>

Opérations mathématiques



Word
Vectors



Vector
Composition

<https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/>

- $\text{vecteur}(\text{king}) - \text{vecteur}(\text{man}) + \text{vecteur}(\text{woman}) = \text{vecteur}(\text{queen})$

Analogies

Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (Skip-gram model trained on 783M words with 300 dimensionality).*

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Similarité et voisins

Cosinus

$$\text{cosine}(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|} = \frac{\sum_{i=1}^N v_i w_i}{\sqrt{\sum_{i=1}^N v_i^2} \sqrt{\sum_{i=1}^N w_i^2}}$$

Speech and Language Processing, Jurafsky & Martin, 3rd ed. draft

- Score cosinus allant de 0 (mots très différents) à 1 (mots très similaires).
- Permet de récupérer les voisins d'un mot donné.

Exemple calcul cosinus

	petit	gris	orange
chat	4	3	1
chien	2	2	0
abricot	1	0	5

Calcul de similarité de chat/chien

$$\cos(chat, chien) = \frac{4 * 2 + 3 * 2 + 1 * 0}{\sqrt{4^2 + 3^2 + 1^2} \sqrt{2^2 + 2^2 + 0^2}} = \frac{14}{\sqrt{26} \sqrt{8}} = 0.97$$

Calcul de similarité de chat/abricot



$$\cos(chat, abricot) = \frac{4 * 1 + 3 * 0 + 1 * 5}{\sqrt{4^2 + 3^2 + 1^2} \sqrt{1^2 + 0^2 + 5^2}} = \frac{9}{\sqrt{26} \sqrt{26}} = 0.34$$

A vous de jouer !

	petit	gris	orange
chat	4	3	1
chien	2	2	0
abricot	1	0	5

Calcul de similarité de chien/abricot

$$\cos(\text{chien}, \text{abricot}) = \frac{2 * 1 + 2 * 0 + 0 * 5}{\sqrt{2^2 + 2^2 + 0^2} \sqrt{1^2 + 0^2 + 5^2}} = \frac{2}{\sqrt{8} \sqrt{26}} = 0.14$$



Insuffisances de la représentation usuelle en « sac de mots »

PRISE EN COMPTE DU CONTEXTE EN TEXT MINING



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



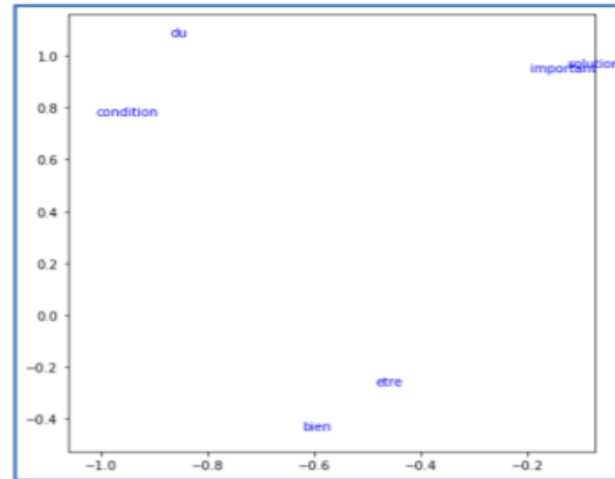
Les termes apparaissant dans des contextes similaires sont proches (au sens de la distance entre les vecteurs de description).



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 ($V = 1$)

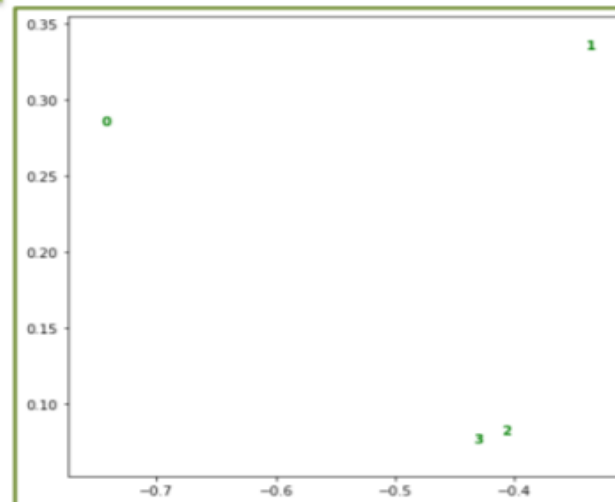
- (0) condition du bien etre
- (1) etre important
- (2) solution bien etre
- (3) important bien etre



Représentation des termes

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

- (A) $K \ll$ 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,...)

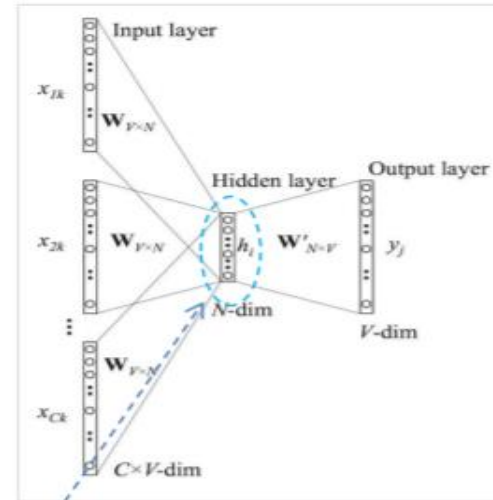


Représentation des documents

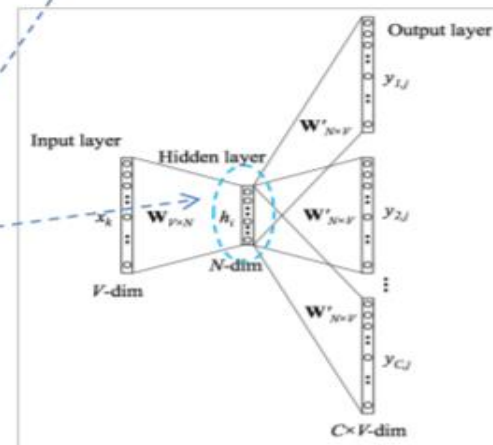
Idée de word2vec

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).

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**.



CBOW

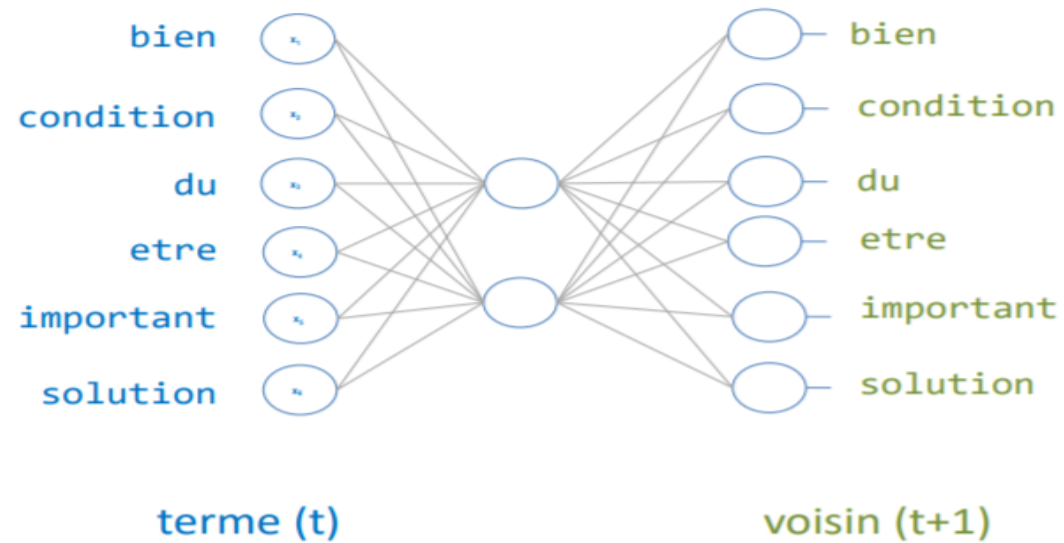


SKIP-GRAM

Modèle SKIP-GRAM

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

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)

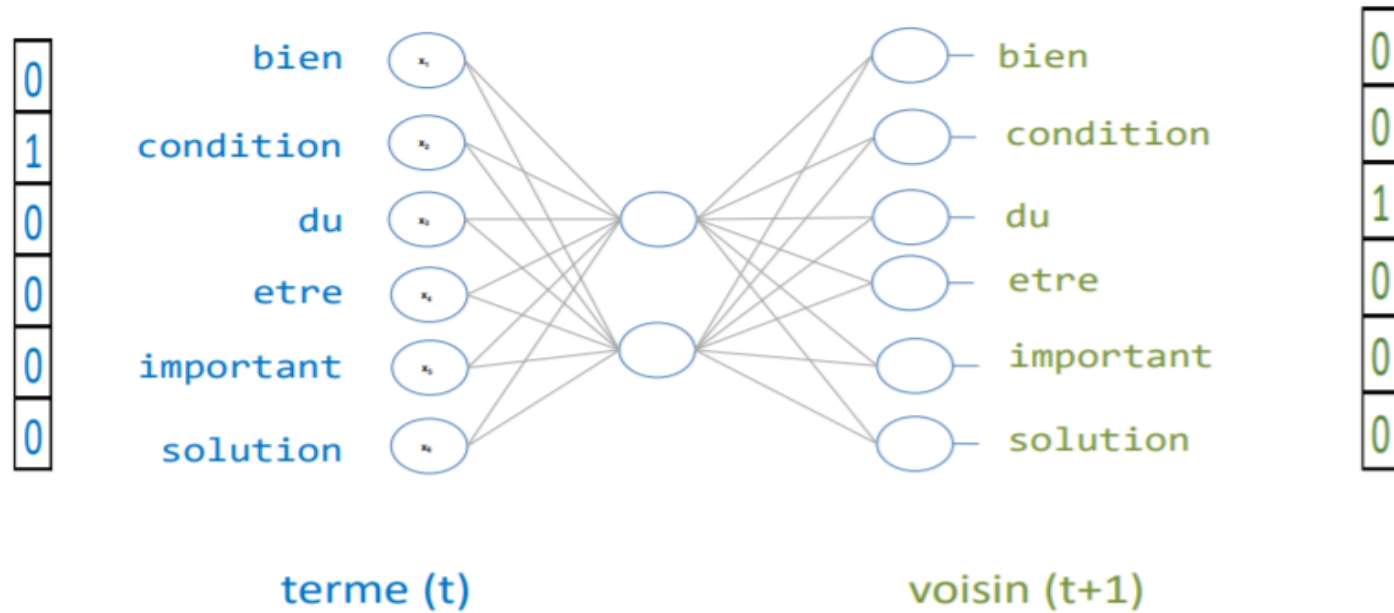
Terme	bien	condition	du	etre	important	solution
du	0	0	1	0	0	0
bien	1	0	0	0	0	0
bien	1	0	0	0	0	0
etre	0	0	0	1	0	0
etre	0	0	0	1	0	0
etre	0	0	0	1	0	0
important	0	0	0	0	1	0
bien	1	0	0	0	0	0

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



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



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.

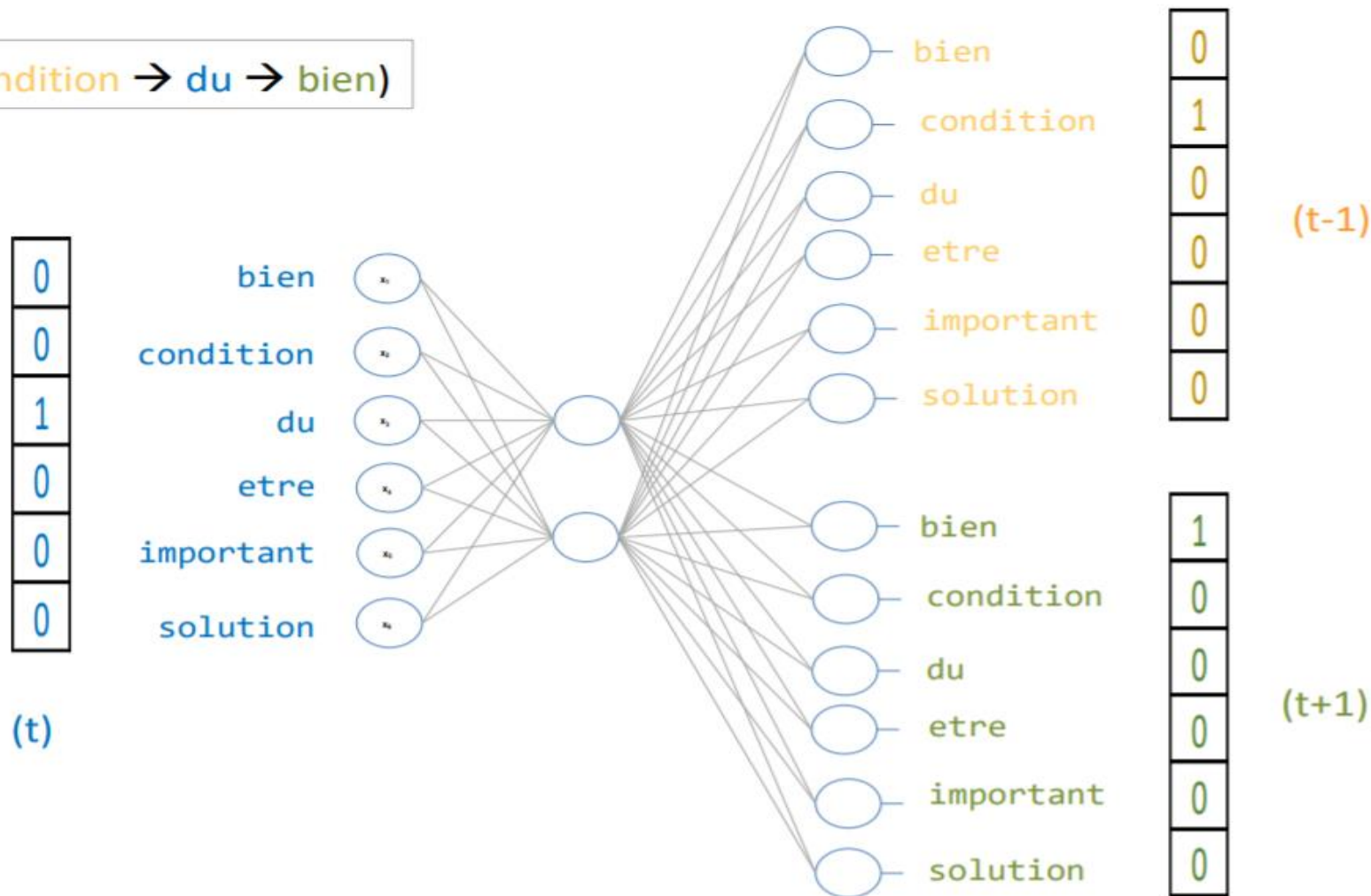
Sortie (voisin t + 1)

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.

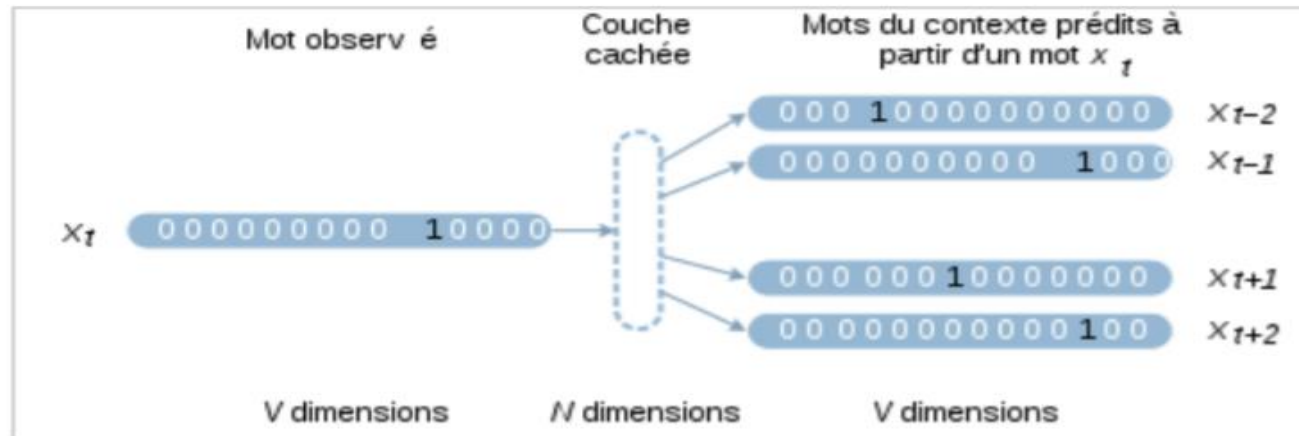
SKIP-GRAM – Prise en compte du voisinage (t-1) et (t+1) – Structure du réseau

Ex. (condition → du → bien)



Modèle SKIP-GRAM – Voisinage plus étendu

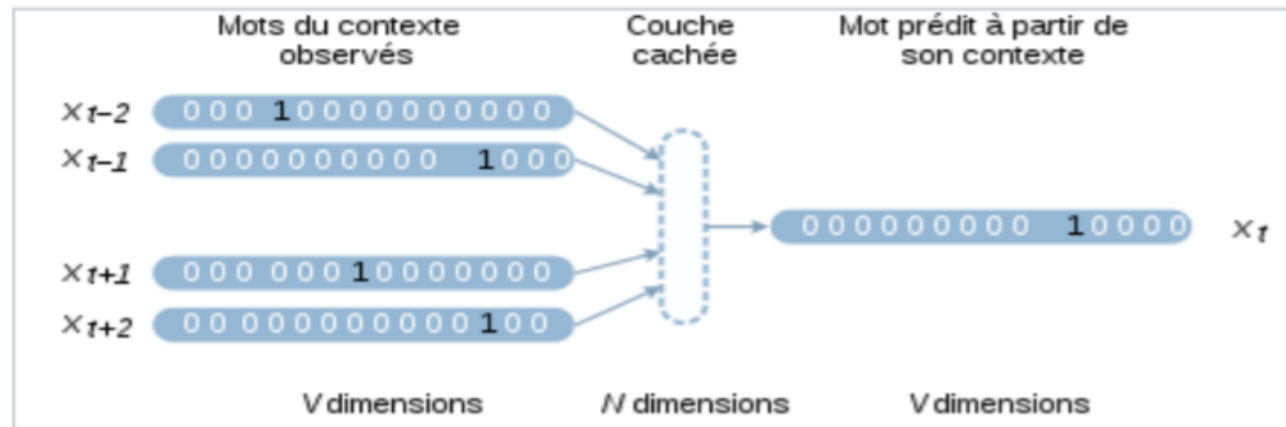
Il est possible de prendre un voisinage plus étendu ($V = 2$ ou plus). Attention simplement à la dilution de l'information.



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 $P(\text{terme} / \text{voisin}[s])$.



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

Word2Vec – Quelques éléments techniques

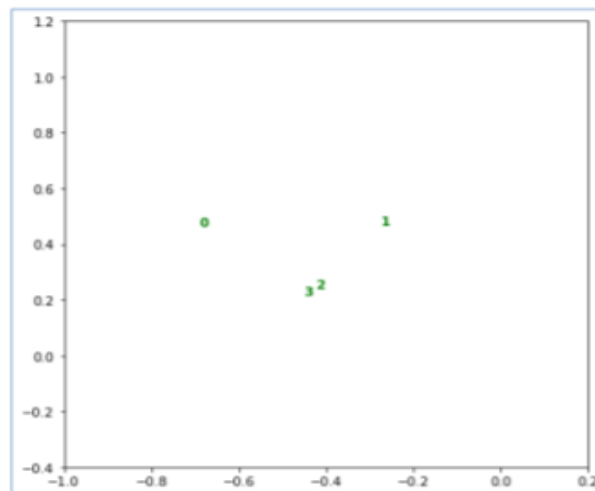
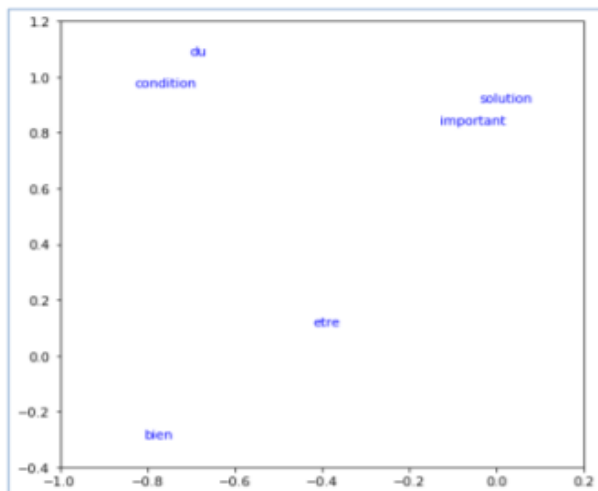
- La fonction de transfert pour la couche centrale est linéaire
- Pour la couche de sortie, la fonction de transfert est [softmax](#)
- La « [negative log-likelihood](#) » fait office de fonction de perte (à la place du classique MSE – mean squared error). En conjonction avec softmax, le calcul du gradient en est [largement simplifiée](#) lors de l'optimisation
- Dicit la [documentation de H2O](#) (fameux package de Deep Learning) : skip-gram donne plus de poids aux voisins proches, elle produit de meilleurs résultats pour les termes peu fréquents

Représentation des documents dans l'espace de dimension K

Disposer d'une représentation des documents dans le nouvel espace est indispensable pour pouvoir appliquer les algorithmes de machine learning (ex. catégorisation de documents, etc.)

```
<document>
<sujet>acq</sujet>
<texte>
Resdel Industries Inc said it has agreed to acquire San/Bar Corp in a share-for-share
exchange, after San/Bar distributes all shgares of its Break-Free Corp subsidiary.
</texte>
</document>
<document>
<sujet>acq</sujet>
<texte>
Warburg, Pincus Capital Co L.P., an investment partnership, said it told representatives
of Symbion Inc it would not increase the 3.50-dlr-per-share cash price it has offered for
the company.
</texte>
</document>
```

Comment passer de la représentation des termes à celle des documents (composés de termes) ?



Représentation des documents

Solution simple : calculer la moyenne des coordonnées (barycentre) des termes qui composent le document.

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

	Word	V1	V2
0	du	-0.703333	1.077643
1	condition	-0.828805	0.966393
2	solution	-0.040079	0.911001
3	important	-0.127894	0.830322
4	bien	-0.808076	-0.295098
5	etre	-0.419983	0.109700



	C1	C2
(0)	-0.690049	0.464660
(1)	-0.273938	0.470011
(2)	-0.422713	0.241868
(3)	-0.451984	0.214975

Calcul des
coordonnées du
document n°1



$$\left\{ \begin{array}{l} C1(1) = \frac{-0.419983 + (-0.127894)}{2} = -0.273938 \\ C2(1) = \frac{0.109700 + 0.830322}{2} = 0.470011 \end{array} \right.$$

Si (K = 2), une représentation simultanée dans le plan est possible.



Conclusion

- 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).
- $K \ll$ 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-entraînés 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](#))



Entraîner des word embeddings avec gensim

gensim - Fonctions utiles

Récupérer les voisins sémantiques d'un mot

```
model.wv.most_similar("cat")
```

```
model.wv.most_similar("cat", topn=5)
```

Rang d'un voisin donné pour un mot donné

```
model.wv.rank("cat", "dog")
```

Calcul du score cosinus entre 2 mots

```
model.wv.similarity("cat", "dog")
```

Analogies

```
model.wv.most_similar(positive=["woman", "king"],  
negative=["man"])
```

References

- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching Word Vectors with Subword Information. In *Emnlp*, volume 91, pages 28–29.
- Gaume, B., Tanguy, L., Fabre, C., Ho-Dac, L.-M., Pierrejean, B., Hathout, N., Farinas, J., Piquier, J., Danet, L., Péran, P., De Boissezon, X., and Jucla, M. (2018). Automatic analysis of word association data from the Evolex psycholinguistic tasks using computational lexical semantic similarity measures. In *13th International Workshop on Natural Language Processing and Cognitive Science (NLPCS)*, Krakow, Poland.
- Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2016). Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Acl 2016*, pages 1489–1501.
- Jurafsky, D. and Martin, J. H. (2018). Vector Semantics. In *Speech and Language Processing*, volume Draft of September 23, 2018.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. In *ICLR Workshop*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. In *arXiv:1310.4546 [cs, stat]*. arXiv: 1310.4546.
- Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.
- Tissier, J., Gravier, C., and Habrard, A. (2017). Dict2vec: Learning Word Embeddings using Lexical Dictionaries. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Copenhagen, Denmark.



**Certaines slides sont reprises du cours de Mon
collègue Ricco Rakotomalala**