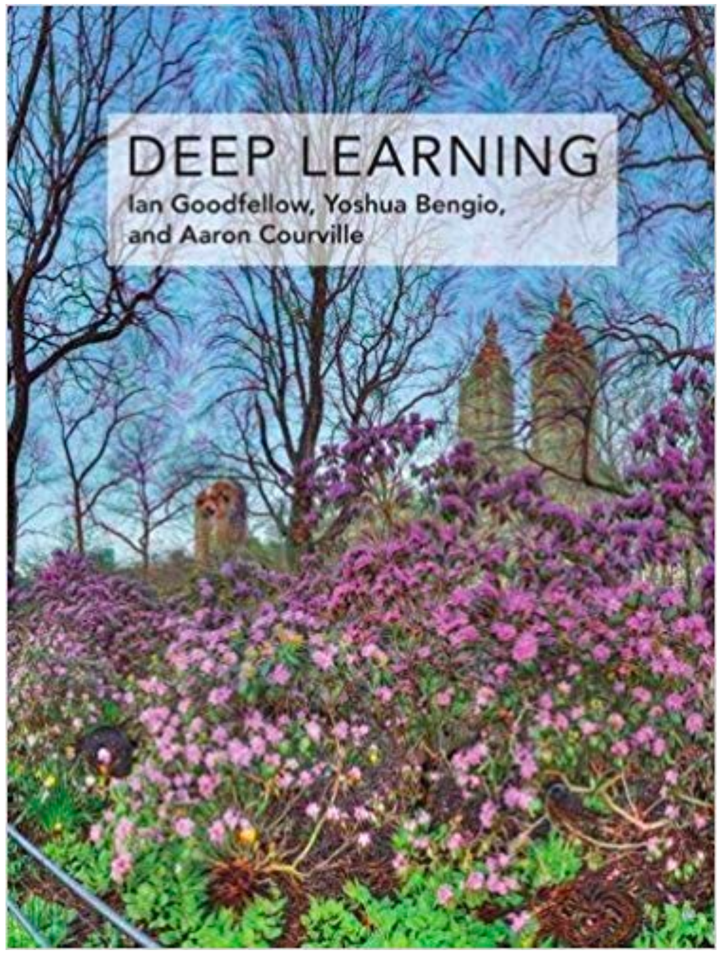
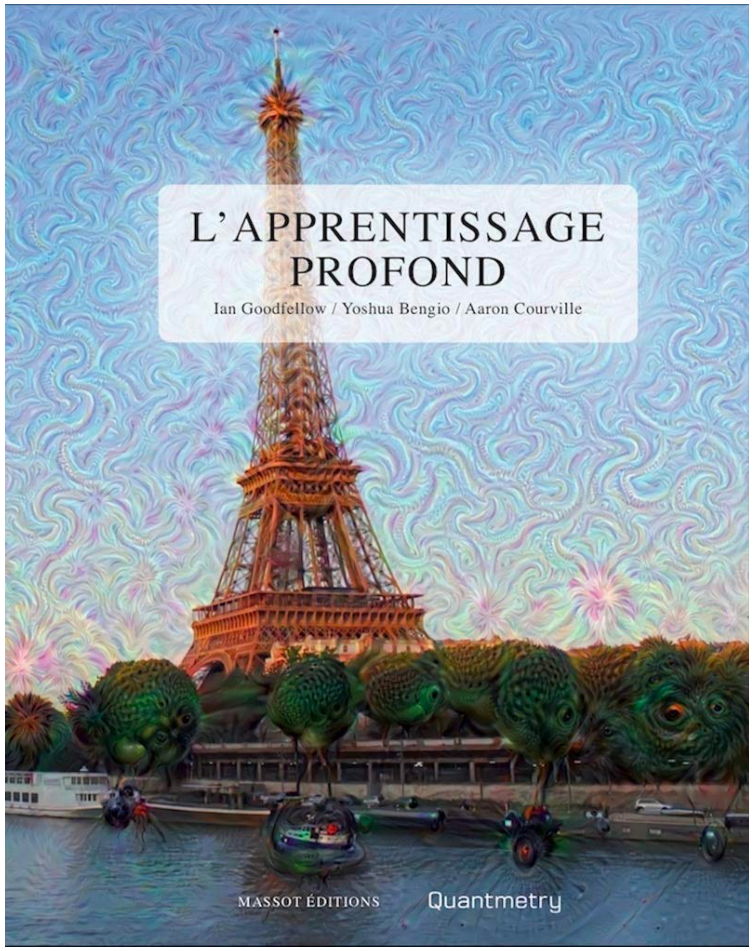
Deep Learning
pour le traitement du langage naturel

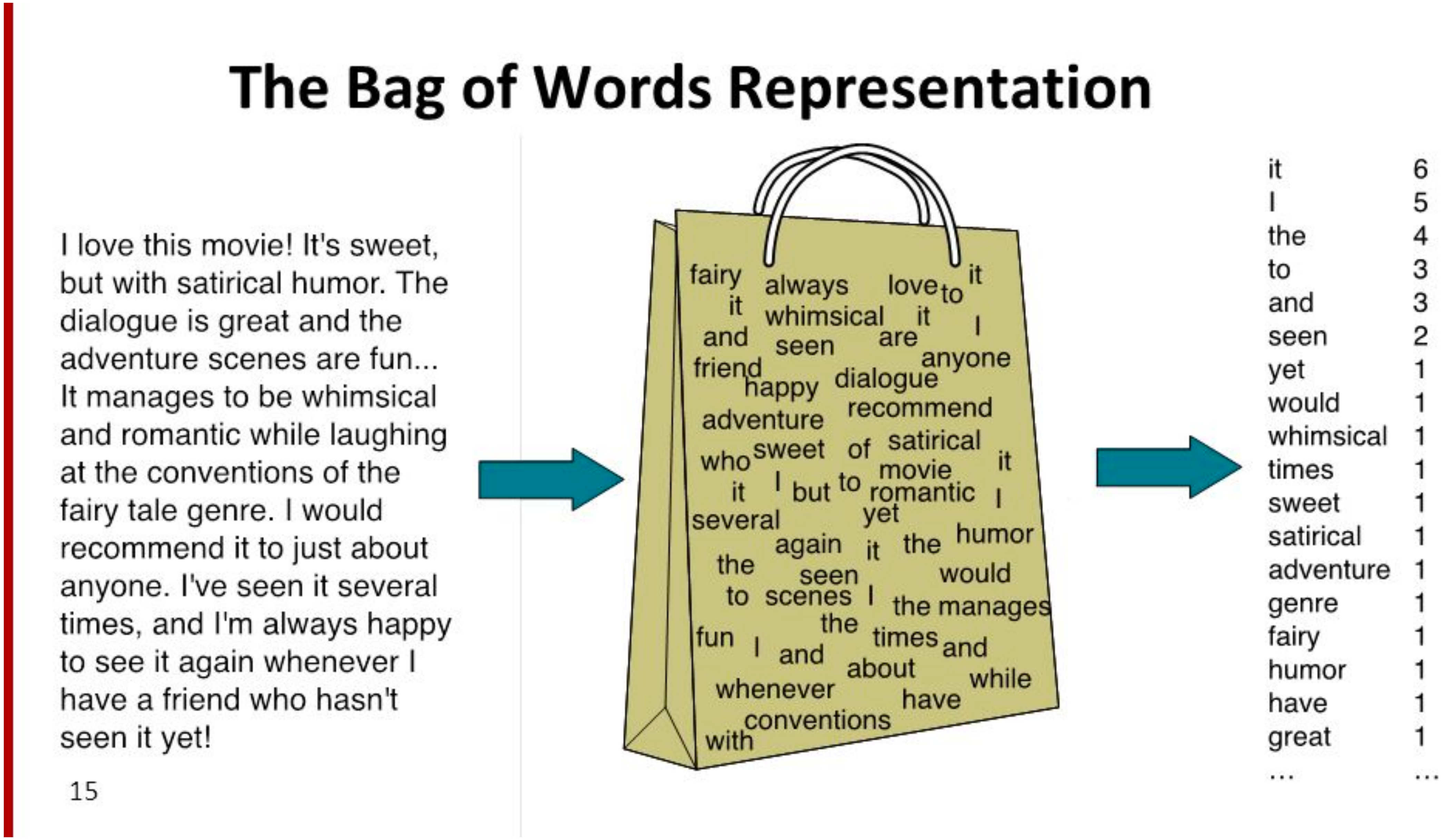

DEEP LEARNING
Ian Goodfellow, Yoshua Bengio, and Aaron Courville


L'APPRENTISSAGE PROFOND
Ian Goodfellow / Yoshua Bengio / Aaron Courville
MASSOT ÉDITIONS
Quantmetry

Emanuela Boros
boros@teklia.com

Christopher Kermorvant
kermorvant@teklia.com

Représentation vectorielle de documents



The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

Représentation vectorielle de documents

Je vous envoie ma nouvelle adresse. Je vous remercie.

Modèle binomial : présence / absence de mot

1	0	1	1	1	1	0	1	1
adresse	commande	envoie	je	ma	nouvelle	réclame	remercie	vous

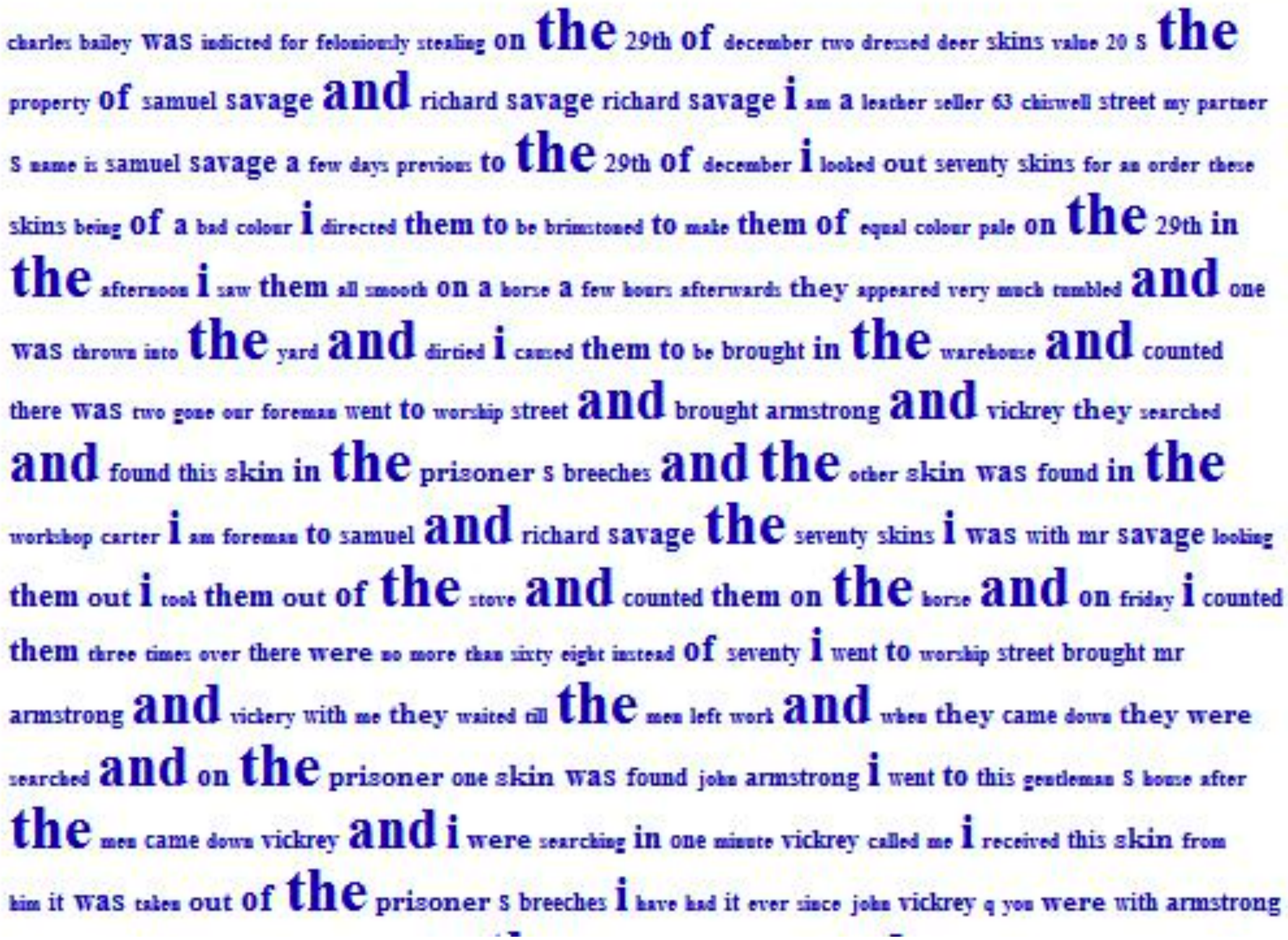
Modèle multinomial : comptage de mots

1	0	1	2	1	1	0	1	2
adresse	commande	envoie	je	ma	nouvelle	réclame	remercie	vous

tf_i (term frequency)

Représentation vectorielle de documents

- Problème du comptage brut : les mots « vides » sont les plus fréquents



charles bailey was indicted for feloniously stealing on the 29th of december two dressed deer skins value 20 s the property of samuel savage and richard savage richard savage i am a leather seller 63 chiswell street my partner s name is samuel savage a few days previous to the 29th of december i looked out seventy skins for an order these skins being of a bad colour i directed them to be brimstoned to make them of equal colour pale on the 29th in the afternoon i saw them all smooth on a horse a few hours afterwards they appeared very much tumbled and one was thrown into the yard and dirtied i caused them to be brought in the warehouse and counted there was two gone our foreman went to worship street and brought armstrong and vickrey they searched and found this skin in the prisoner s breeches and the other skin was found in the workshop carter i am foreman to samuel and richard savage the seventy skins i was with mr savage looking them out i took them out of the stove and counted them on the horse and on friday i counted them three times over there were no more than sixty eight instead of seventy i went to worship street brought mr armstrong and vickery with me they waited till the men left work and when they came down they were searched and on the prisoner one skin was found john armstrong i went to this gentleman s house after the men came down vickrey and i were searching in one minute vickrey called me i received this skin from him it was taken out of the prisoner s breeches i have had it ever since john vickrey q you were with armstrong

Représentation vectorielle de documents

- **Solution 1** : supprimer les mots vides (*black list*)
- **Solution 2** : pénaliser les mots qui apparaissent dans beaucoup de documents

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

idf_i = inverse document frequency

|D| : nombre total de documents dans le corpus.

$|\{d_j : t_i \in d_j\}|$: nombre de document où le mot t_i apparait.

Représentation très utilisée : $\mathrm{tf}_i * \mathrm{idf}_i$

Représentation vectorielle de documents

Comment représenter les similarités de termes ?

L'altitude du Mont Blanc est 4810 mètres
La hauteur du Mont Blanc est de 4810 mètres

Dans une représentation vectorielle classique :

Distance (altitude, hauteur) = Distance (altitude, lavabo)

Aucune prise en compte de la proximité sémantique

Représentation vectorielle des mots

Comment apprendre le sens des mots ?

- Approches par dictionnaires, ontologies
- Approches par corpus

Firth (1957): ”You shall know a word by the company it keeps!”

Apprentissage du sens d’un mot par ses contextes d’usage

Représentation vectorielle des mots

“Tesgüino” ?

1. lac finlandais

2. boisson mexicaine

3. manga japonais

Nida, E. A. 1975. Componential analysis of meaning: An introduction to semantic structures.

Représentation vectorielle des mots

Tesgüino ?

Une bouteille de tesgüno est sur la table.

Le tesgüino est produit en Sierra Madre occidentale au Mexique.

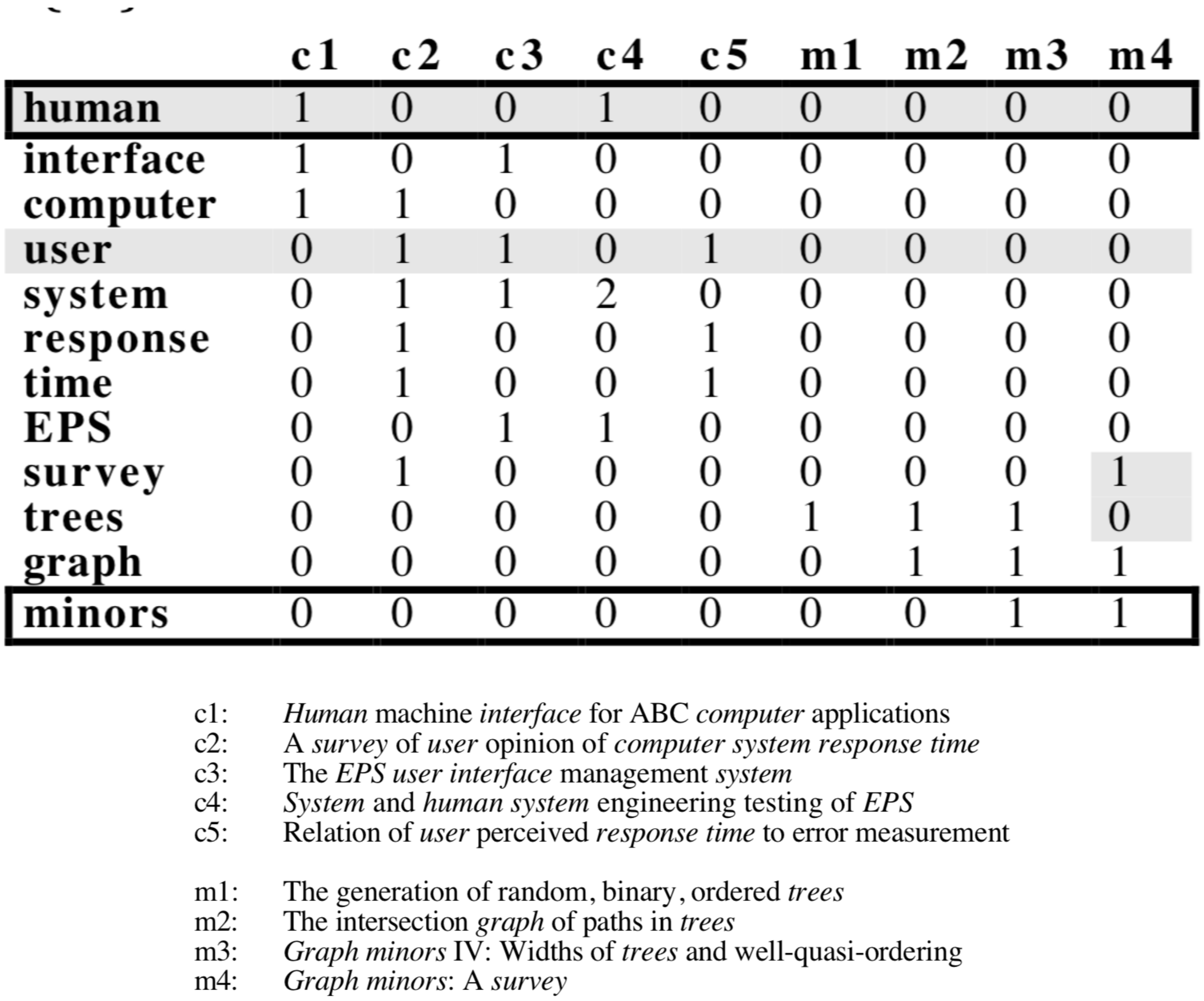
Boire du tesgüino rend ivre.

On fabrique le tesgüono à partir de maïs.

Représentation du sens d’un mot par ses contextes d’usage

Représentation vectorielle des mots

Matrice terme-document : représenter les mots par les documents dans lesquels ils apparaissent



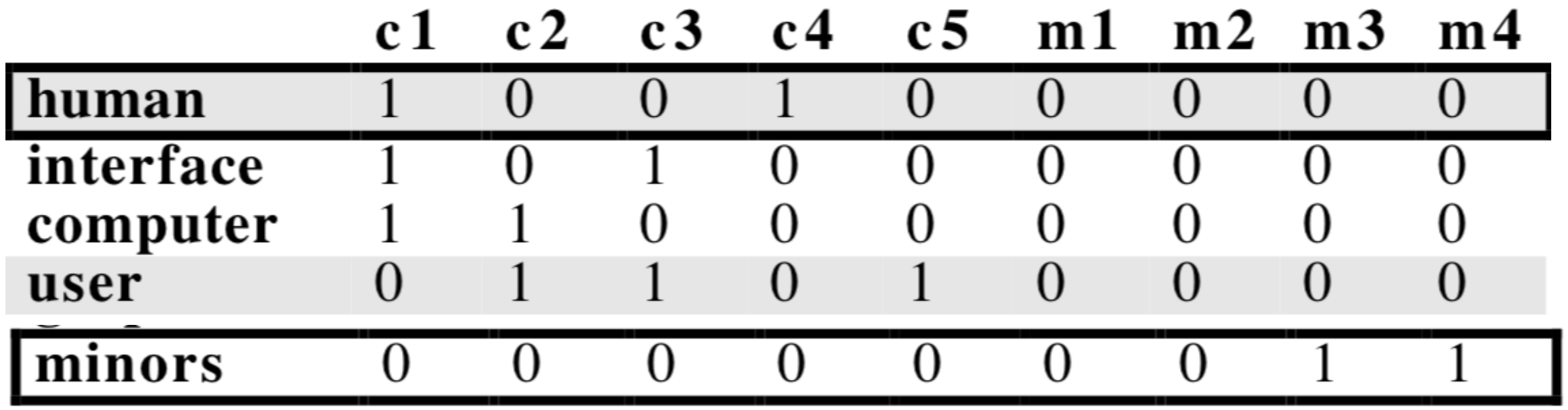
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

c1: *Human* machine *interface* for ABC *computer* applications
c2: A *survey* of *user* opinion of *computer system response time*
c3: The *EPS user interface* management *system*
c4: *System* and *human system* engineering testing of *EPS*
c5: Relation of *user* perceived *response time* to error measurement

m1: The generation of random, binary, ordered *trees*
m2: The intersection *graph* of paths in *trees*
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4: *Graph minors*: A *survey*

Représentation vectorielle des mots

Limite : il faudrait beaucoup de documents pour bien représenter les mots

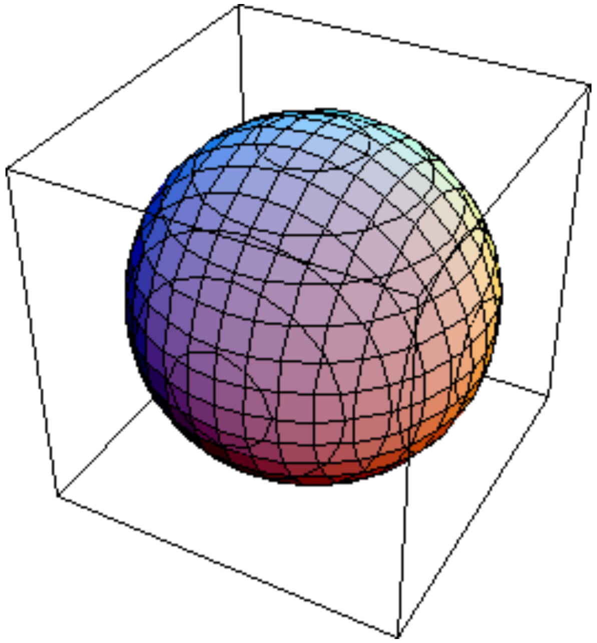


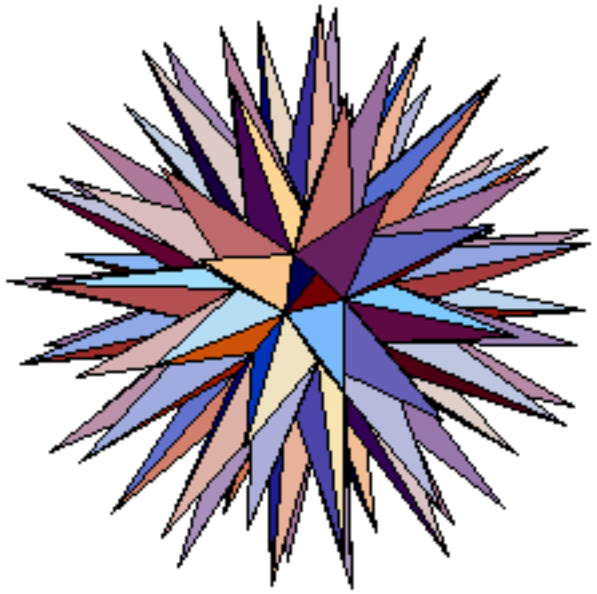
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
minors	0	0	0	0	0	0	0	1	1

- *human* et *user* ne partagent aucun contexte (document)
- *human* et *minors* non plus
- distance (*human,user*) > distance(*human,minors*)

Malédiction de la dimensionalité

- Dans un espace en haute dimension, tous les points sont loins les uns des autres.
- Le nombre de données nécessaires pour couvrir l'espace augmente de manière exponentielle.





Malédiction de la dimensionalité

Exemple : volume d'une sphère de rayon 0.5 inscrite dans un cube de coté 1

dimension 2 :

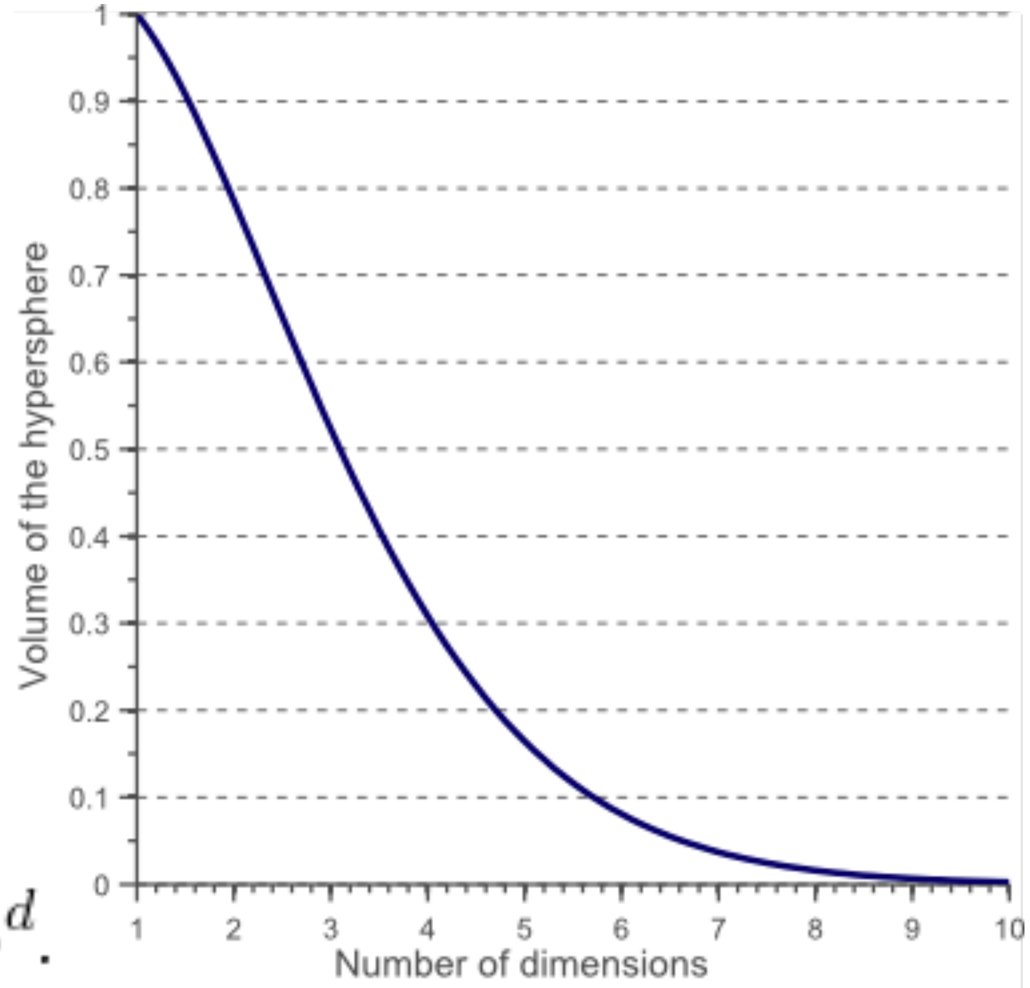
$V_{cube} = 1 \quad V_{sphère} = \pi/4$

dimension 3 :

$V_{cube} = 1 \quad V_{sphère} = \pi/6$

dimension d :

$V_{cube} = 1 \quad V_{sphère} = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} 0.5^d.$

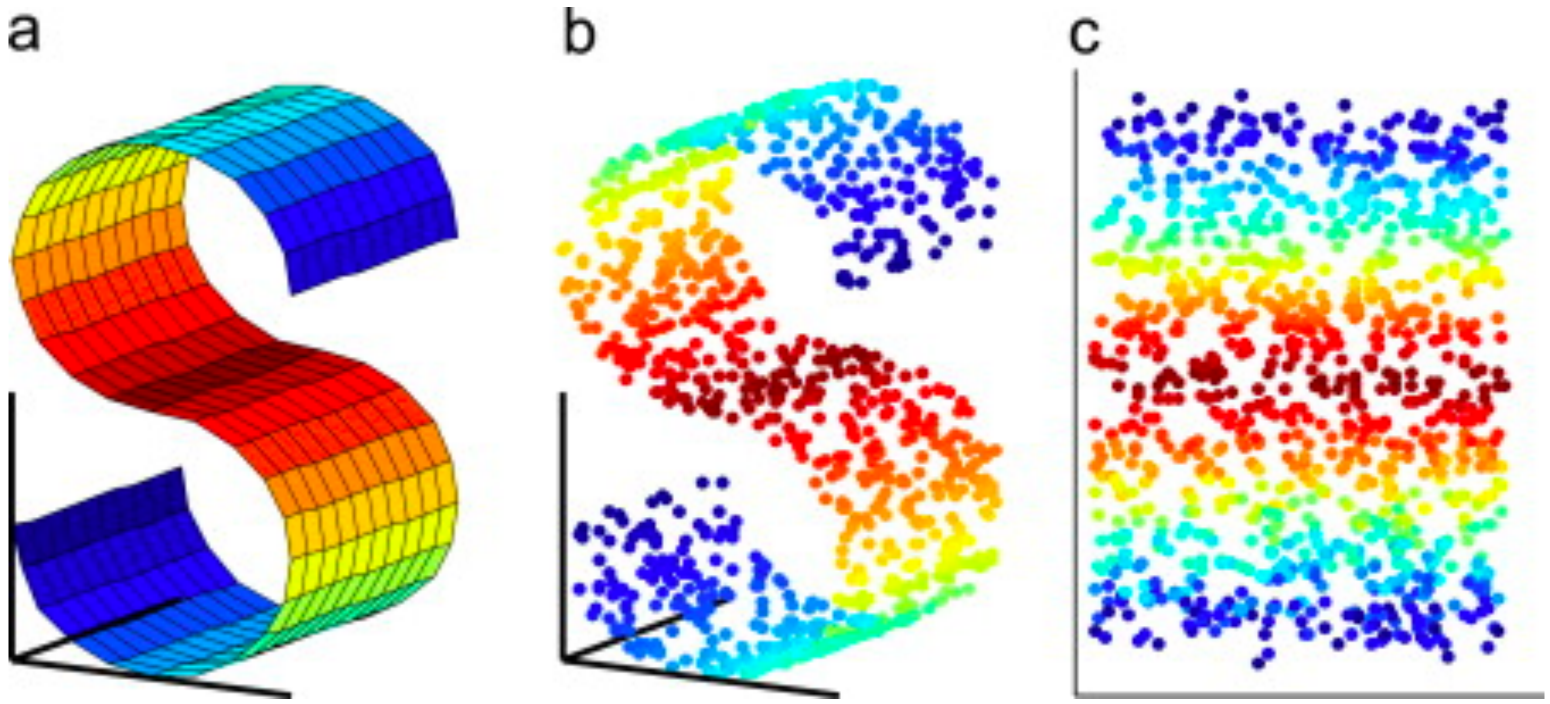


Malédiction de la dimensionalité

- Lorsque la dimension augmente, le volume de la sphère devient négligeable par rapport au volume du cube : tous les points de l’espace sont dans les coins, éloignés les uns des autres.
- Plus la dimension augmente, plus il faut de points pour couvrir l’espace et estimer les modèles

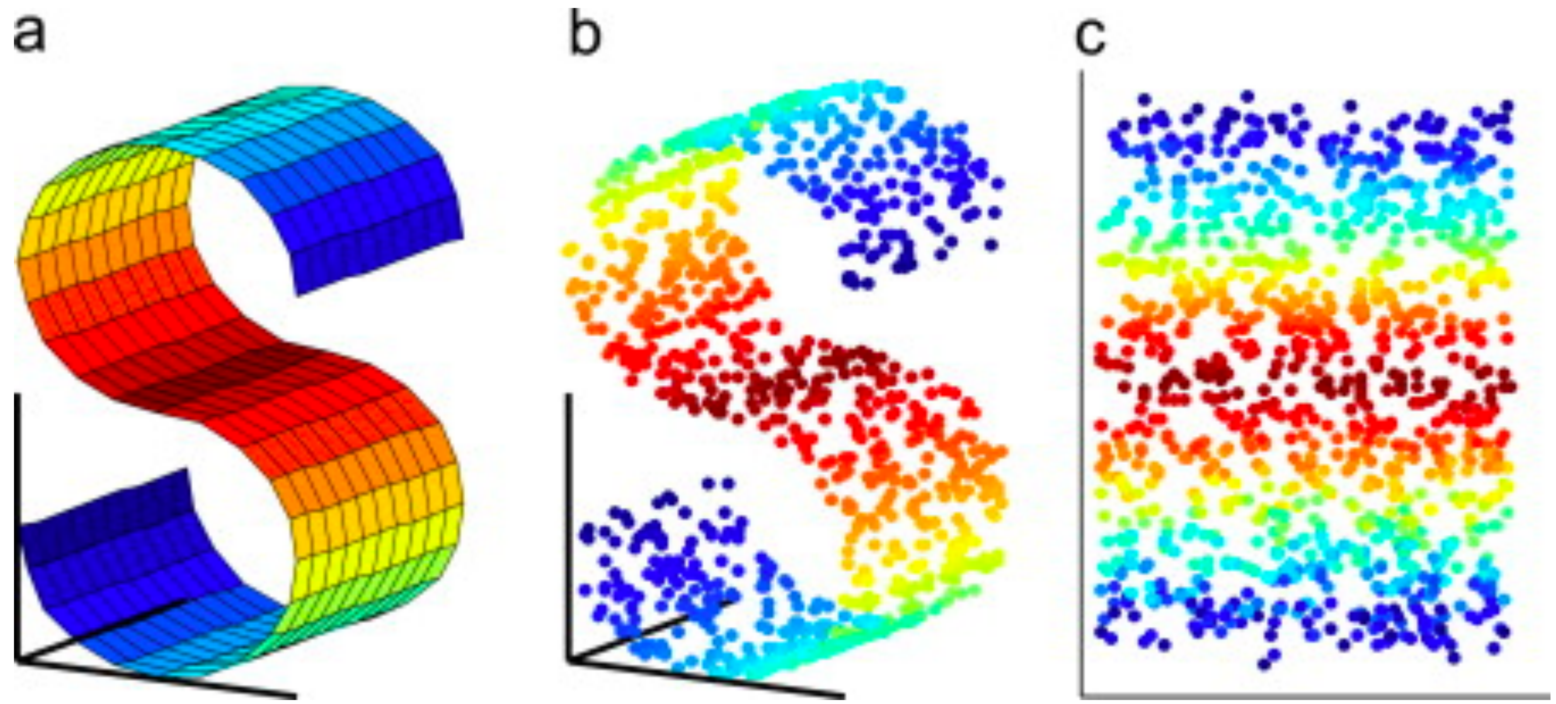
Réduction de dimension

Intuition : en réalité, les points n’occupent pas uniformément tout l’espace



- les points sont représentés en 3 dimensions (figures a. et b.)
- mais en réalité, ils sont disposés sur un sous-espace de dimension 2 (figure c.)

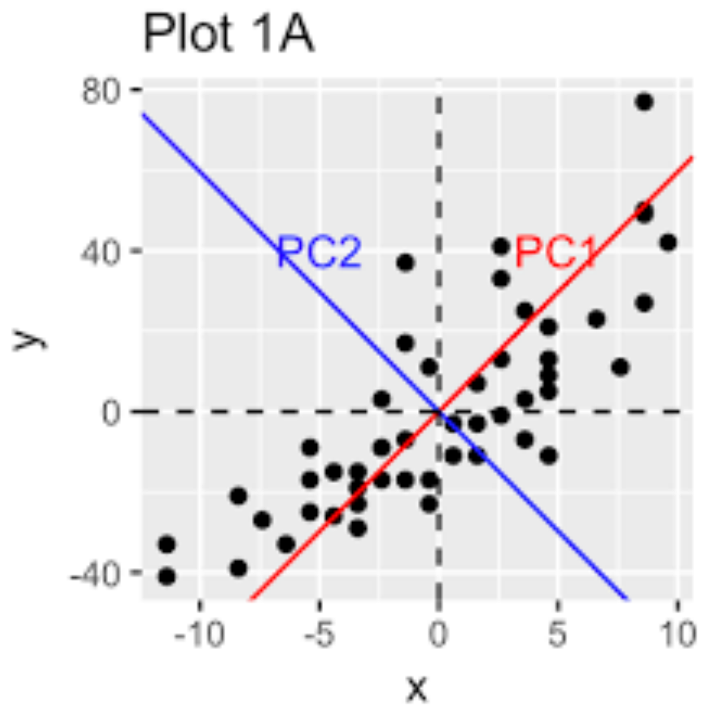
Réduction de dimension



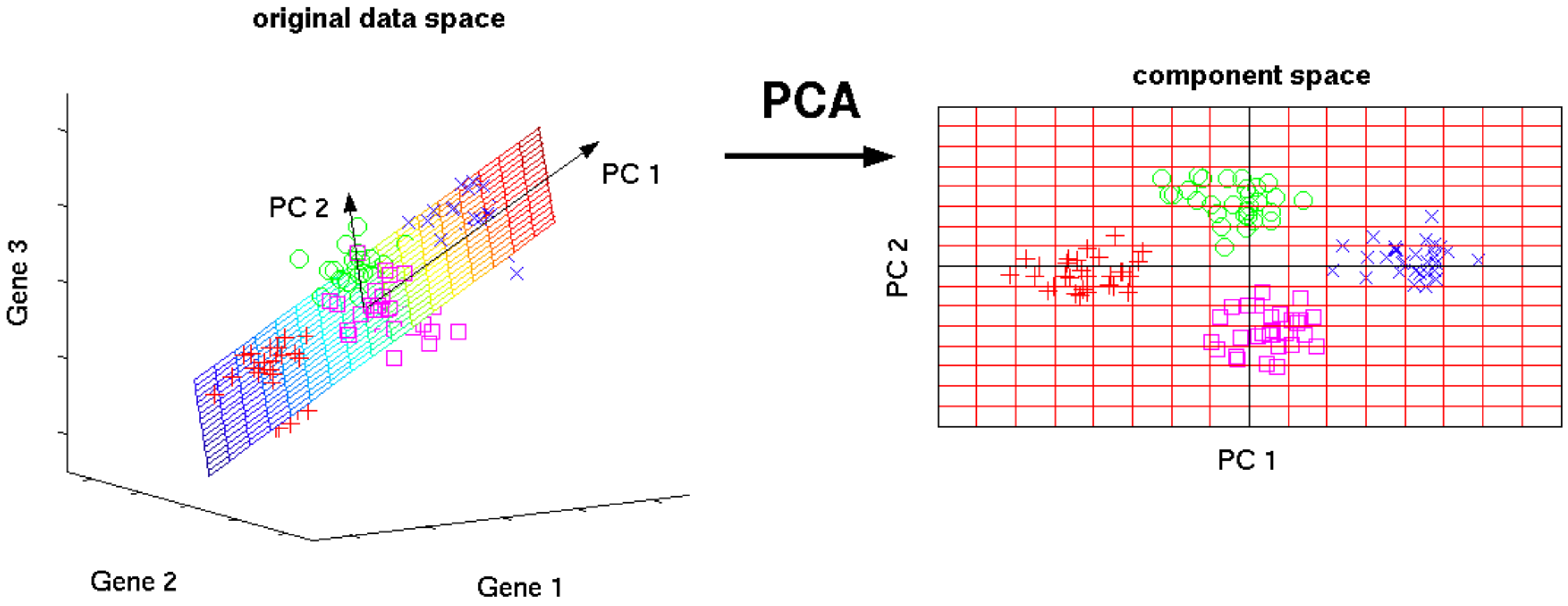
On peut donc conserver les relations entre les points tout en réduisant la dimension de l'espace de représentation

Réduction de dimension

Analyse en composantes principales (PCA)

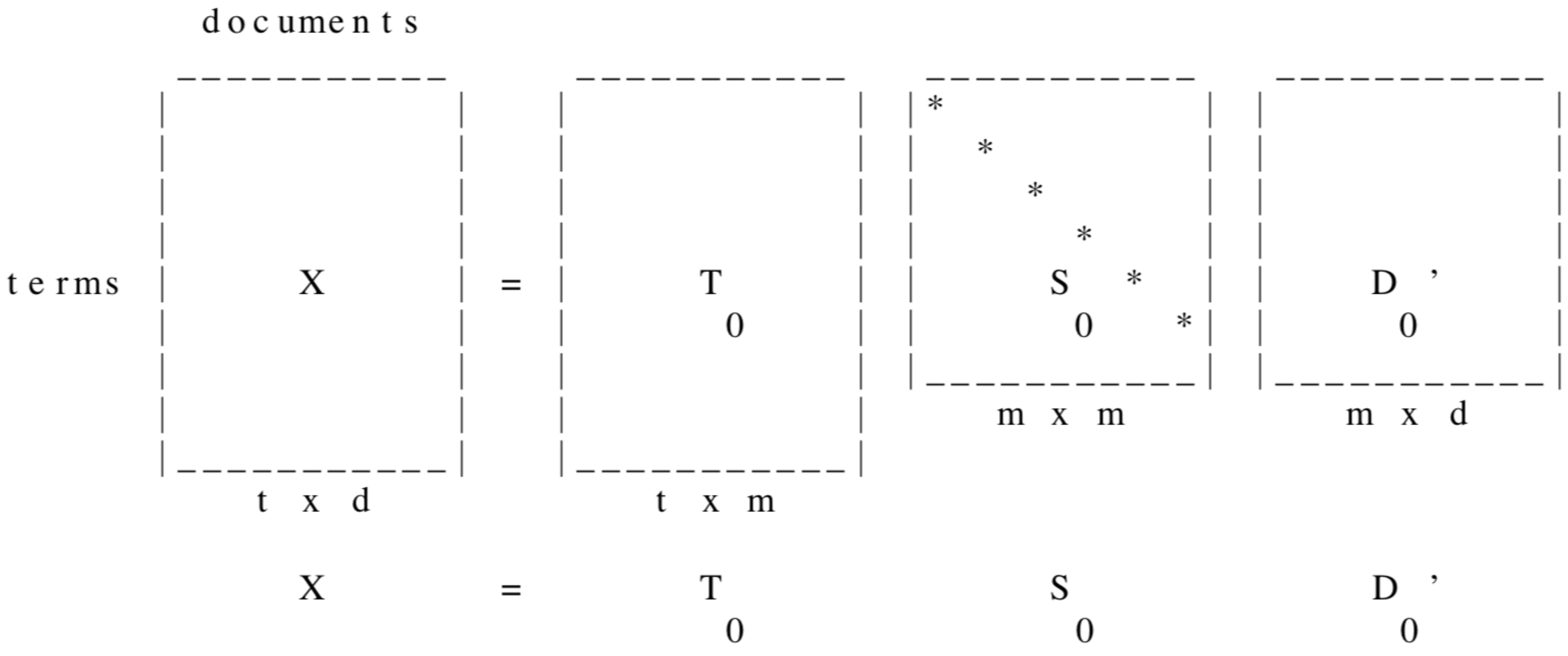


- Recherche des axes des composantes principales
- Sélection des composantes
- Projection des données sur les axes des composantes



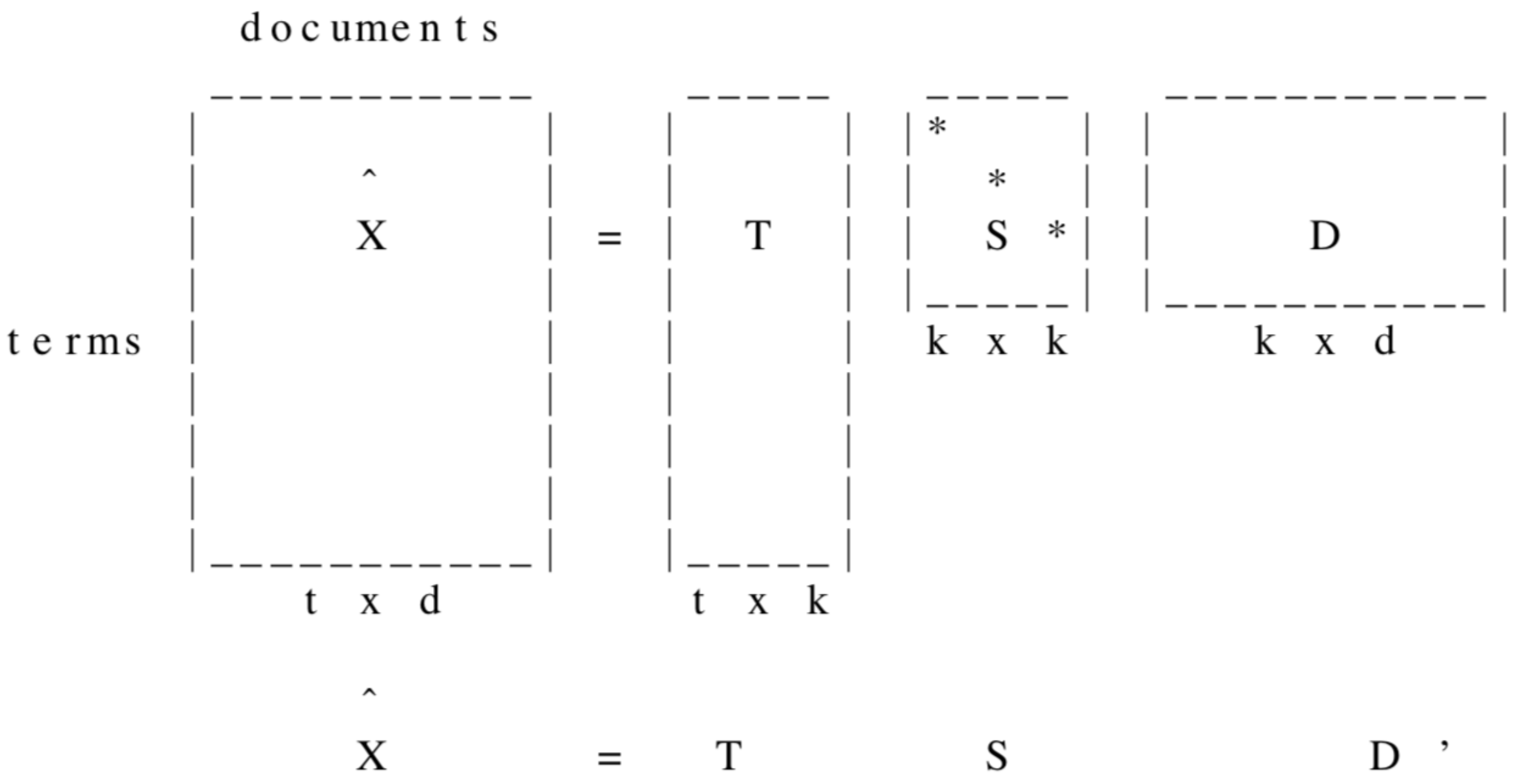
Latent semantic indexing/analysis

- **Singular Value Decomposition :** extension de l'ACP aux matrices non carrés
- **Latent Semantic Indexing** : SVD appliqué à la matrice termes-documents



Deerwester et *al*., Indexing by latent semantic analysis, 1990

Latent semantic indexing/analysis



- Approximation de rang inférieur : k < m (rang de X, k ≈ 300
- LSA/LSI représente le sens d'un mot comme une moyenne pondérée du sens des documents dans lesquel il apparait et en même temps le sens d'un document comme une moyenne pondérée des sens mots qu'il contient.

Latent semantic indexing/analysis

Limites de LSI :

- Les mots les plus fréquents ont un poids très important dans la matrice de co-occurrence
- LSI modélise la relation statistiques des mots et des documents : difficulté de passage à l'échelle si le nombre de mots ou documents augmente
- LSI ne prend en compte ni l'ordre des mots, ni leur relations syntaxiques ou logiques, ni la morphologie

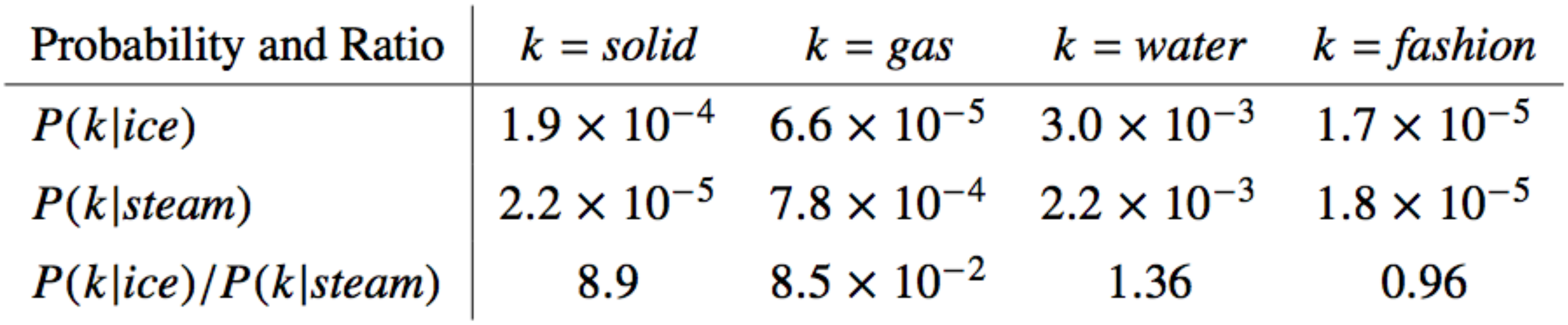
Solutions proposées :

- Normalisation de la matrice basée sur
  - l'entropie ou la corrélation (*COALS, Rohde et al., 2006)*
  - Pointwise mutual information (PPMI, Bullinaria and Levy, 2007)
- …

Glove

Pennington *et al.*, Glove: Global vectors for word representation. 2014

- Prise en compte des co-occurrences des mots pour la création de représentations vectorielles
- **Objectif** : trouver une représentation vectorielle qui conserve les ratios de fréquence de co-occurrence



Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

$$P_{ij} = P(j|i) = \frac{X_{ij}}{Xi}$$

Glove

Apprentissage : minimisation de J

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

V : taille du vocabulaire

f : fonction de pondération

b_i : permet de prendre en compte les différences de fréquence X_i

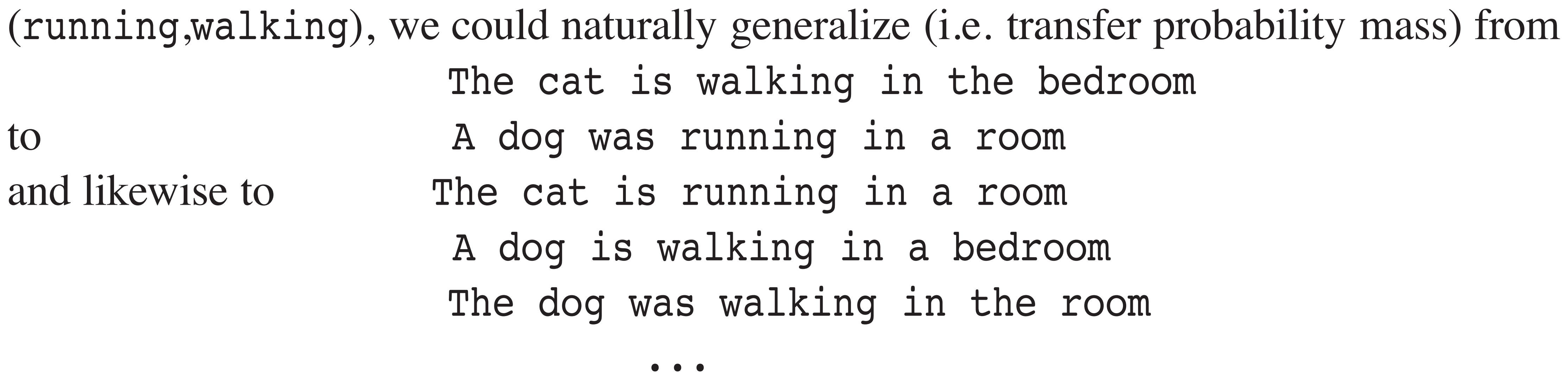
W_i : représentation vectorielle

Recherche des représentation vectorielles des mots qui approchent le mieux $log(X_{ij})$

Optimisation par descente de gradient

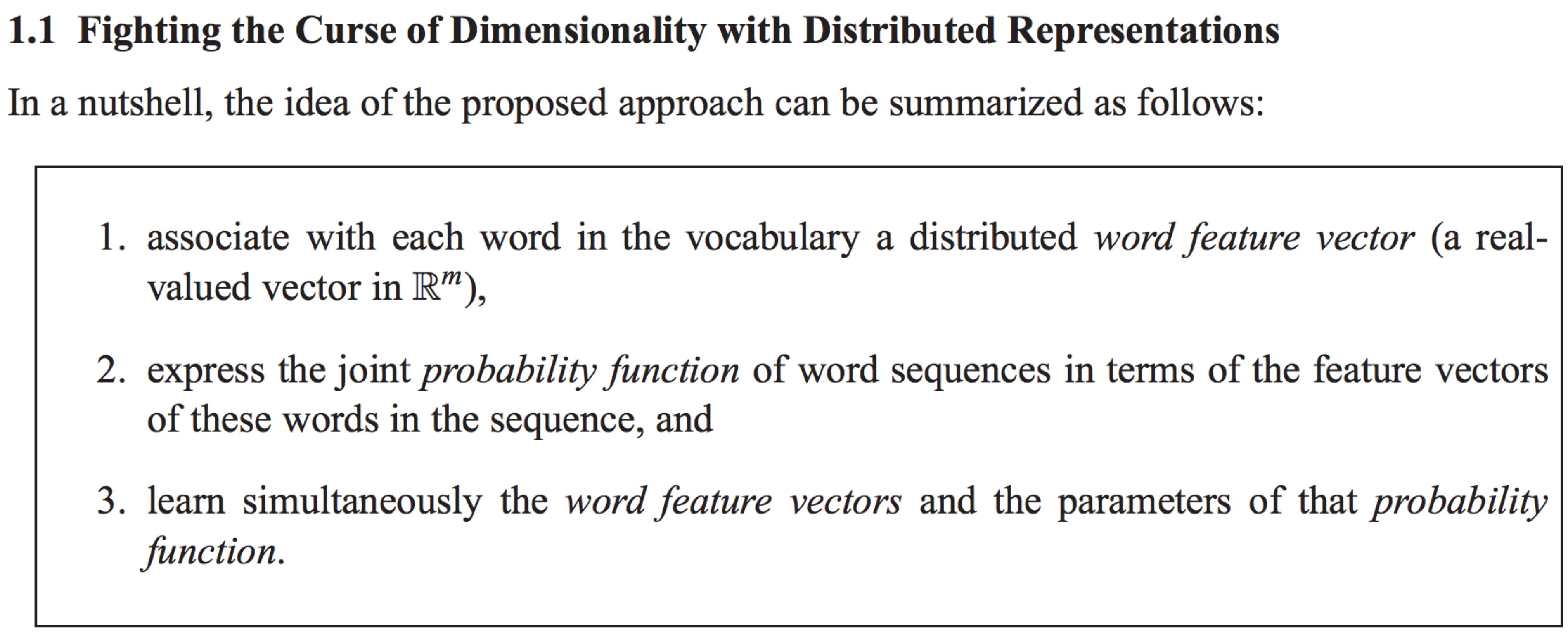
A Neural Probabilistic Language Model
Bengio et al., 2003

Modélisation statistique de la langue



(`running`,`walking`), we could naturally generalize (i.e. transfer probability mass) from

`The cat is walking in the bedroom`
to `A dog was running in a room`
and likewise to `The cat is running in a room`
`A dog is walking in a bedroom`
`The dog was walking in the room`
`...`

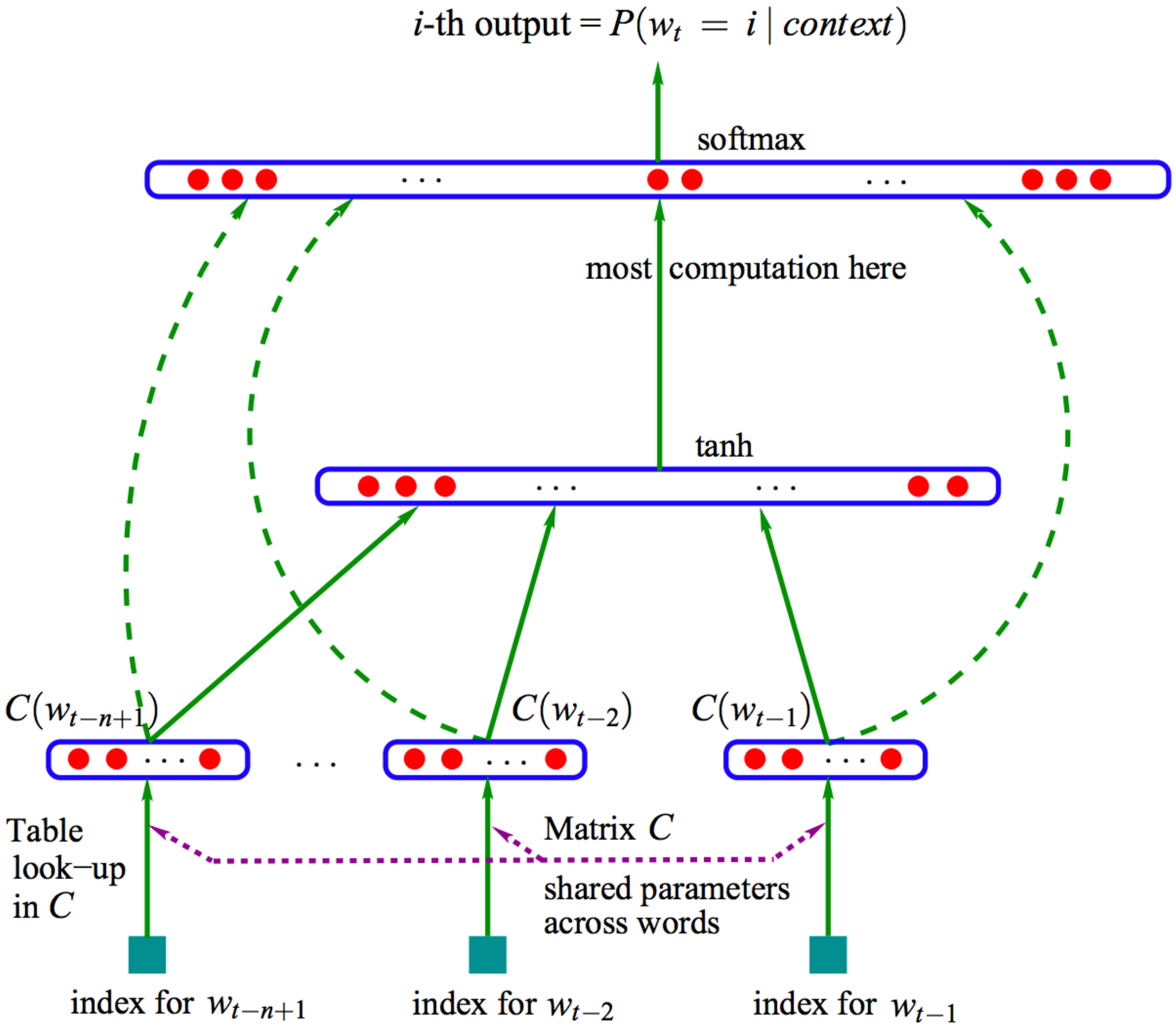


1.1 Fighting the Curse of Dimensionality with Distributed Representations

In a nutshell, the idea of the proposed approach can be summarized as follows:

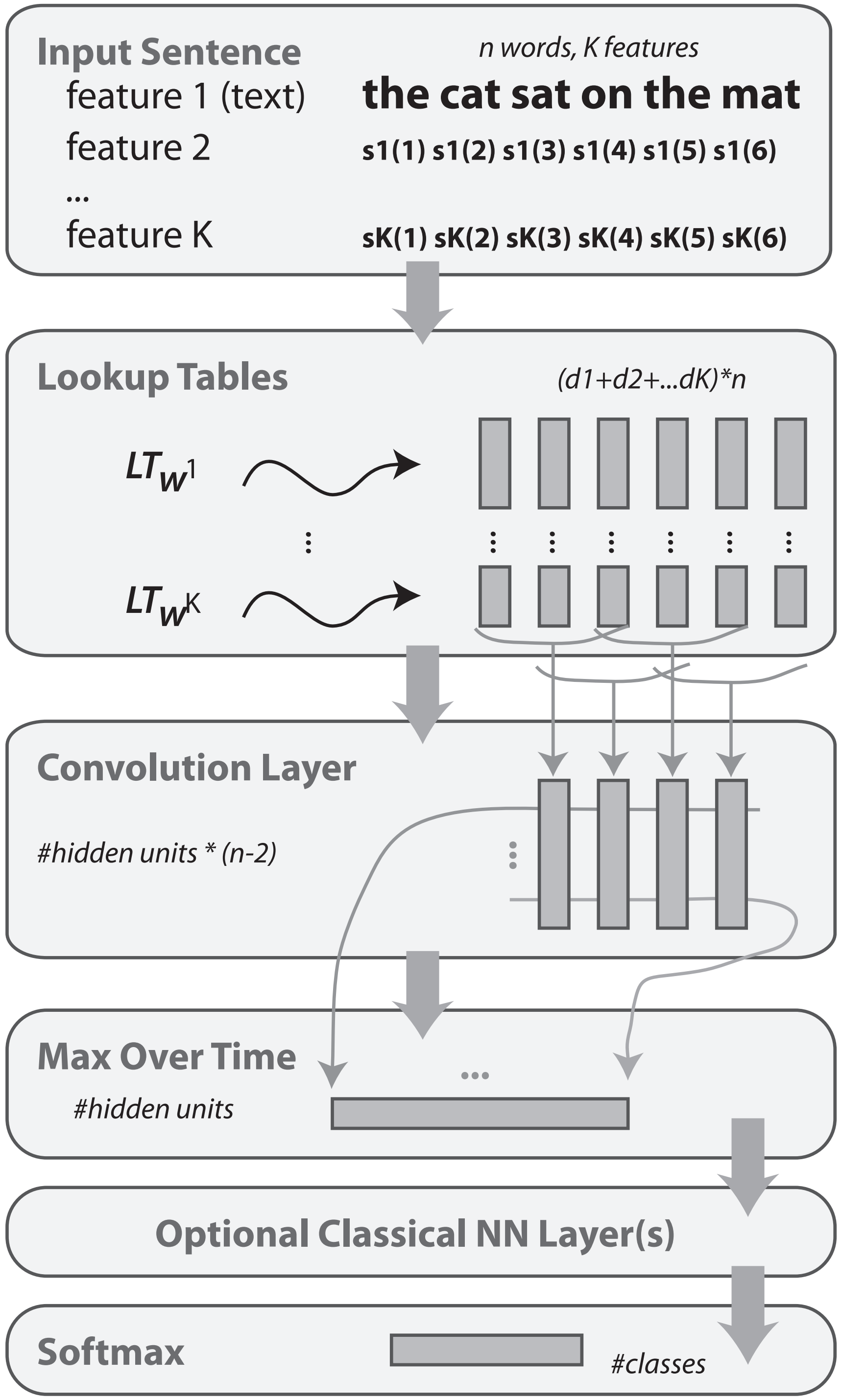
1. associate with each word in the vocabulary a distributed *word feature vector* (a real-valued vector in $\mathbb{R}^m$),
2. express the joint *probability function* of word sequences in terms of the feature vectors of these words in the sequence, and
3. learn simultaneously the *word feature vectors* and the parameters of that *probability function*.

A Neural Probabilistic Language Model
Bengio et al., 2003



A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning, ICML 2008

Ronan Colobert et Jason Weston



Apprentissage des représentations (deep learning versus shallow features)

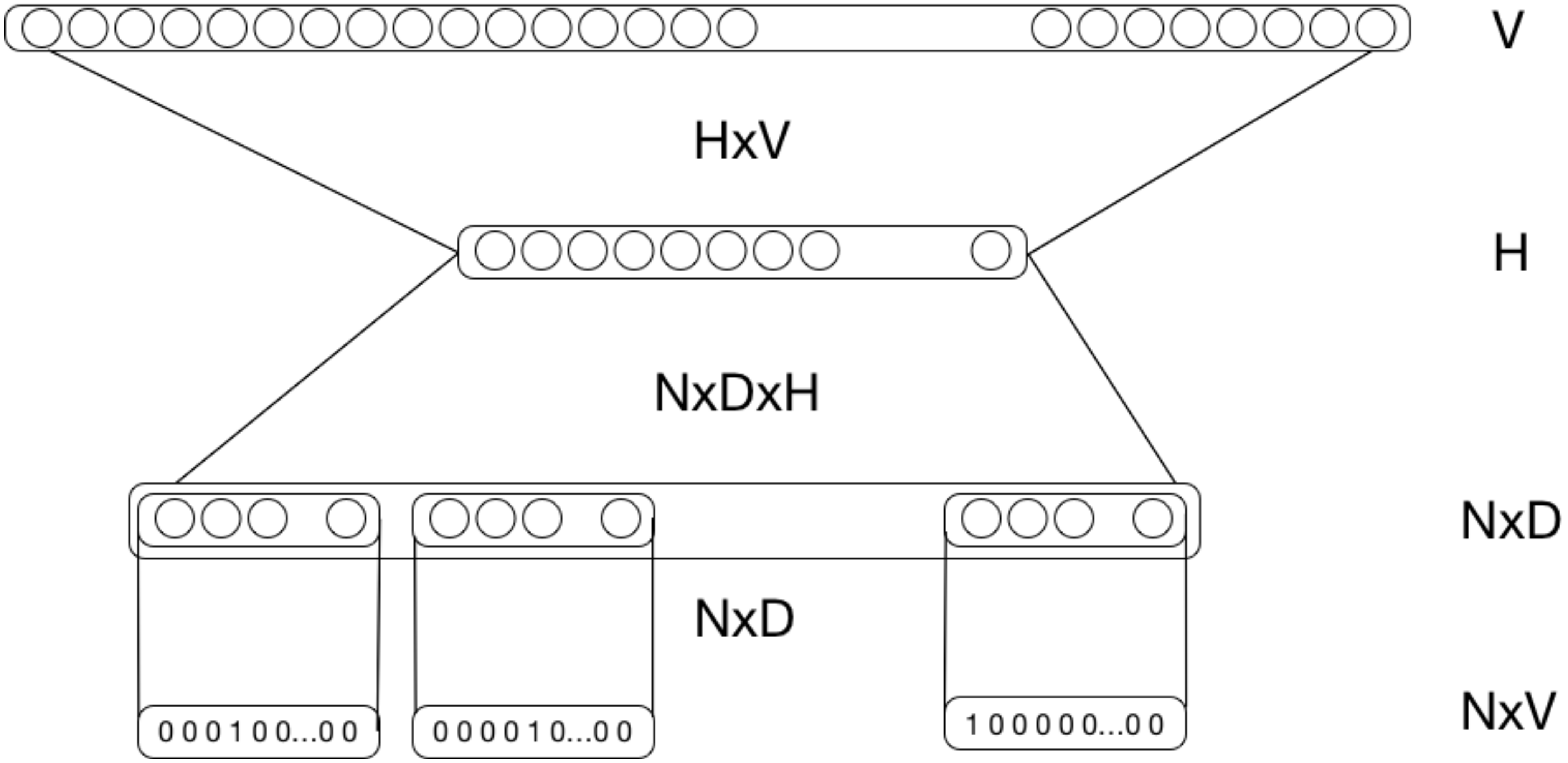
Entrainement end-to-end versus features engineering + classifieur

Pré-entrainement non supervisé (modèle de langue)

Apprentissage supervisé mutli-tâche

2018 International Conference on Machine Learning (ICML) “Test of Time Award”

A Neural Probabilistic Language Model
Bengio et al., 2003



V = 100 000
D = 50 à 200
H = 500 à 1000

HxV est énorme mais il existe des techniques d'accélération

NxDxH reste problématique

Word2Vec

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.

1 Introduction

Many current NLP systems and techniques treat words as atomic units - there is no notion of similarity between words, as these are represented as indices in a vocabulary. This choice has several good reasons - simplicity, robustness and the observation that simple models trained on huge amounts of data outperform complex systems trained on less data. An example is the popular N-gram model used for statistical language modeling - today, it is possible to train N-grams on virtually all available data (trillions of words [3]).

However, the simple techniques are at their limits in many tasks. For example, the amount of relevant in-domain data for automatic speech recognition is limited - the performance is usually dominated by the size of high quality transcribed speech data (often just millions of words). In machine translation, the existing corpora for many languages contain only a few billions of words or less. Thus, there are situations where simple scaling up of the basic techniques will not result in any significant progress, and we have to focus on more advanced techniques.

With progress of machine learning techniques in recent years, it has become possible to train more complex models on much larger data set, and they typically outperform the simple models. Probably the most successful concept is to use distributed representations of words [10]. For example, neural network based language models significantly outperform N-gram models [1, 27, 17].

1.1 Goals of the Paper

The main goal of this paper is to introduce techniques that can be used for learning high-quality word vectors from huge data sets with billions of words, and with millions of words in the vocabulary. As far as we know, none of the previously proposed architectures has been successfully trained on more

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Kai Chen
Greg Corrado
Jeffrey Dean

Proceedings of Workshop at ICLR, 2013.

Word2Vec

- **Objectif** : étant donné un mot $\mathrm{w_t}$ dans un corpus de taille T, prédire les mots $\mathrm{w_c}$ qui peuvent apparaitre dans son contexte :

$$\sum_{t=1}^{T} \sum_{c \in \mathcal{C}_t} \log p(w_c \mid w_t)$$

- Si s est une fonction de similarité entre mots, la probabilité d'un mot $\mathrm{w_c}$ conditionnellement à un autre mot $\mathrm{w_t}$ peut être calculée par :

$$p(w_c \mid w_t) = \frac{e^{s(w_t,\, w_c)}}{\sum_{j=1}^{W} e^{s(w_t,\, j)}}$$

Word2Vec

Le problème avec cette fonction objectif est le terme de normalisation : il nécessite de calculer s sur tous les mots

$$p(w_c \mid w_t) = \frac{e^{s(w_t,\, w_c)}}{\sum_{j=1}^{W} e^{s(w_t,\, j)}}$$

- Word2Vec remplace donc cette fonction objectif par une tâche de classification : prédire si oui ou non un mot apparait dans le contexte d'un mot donné :

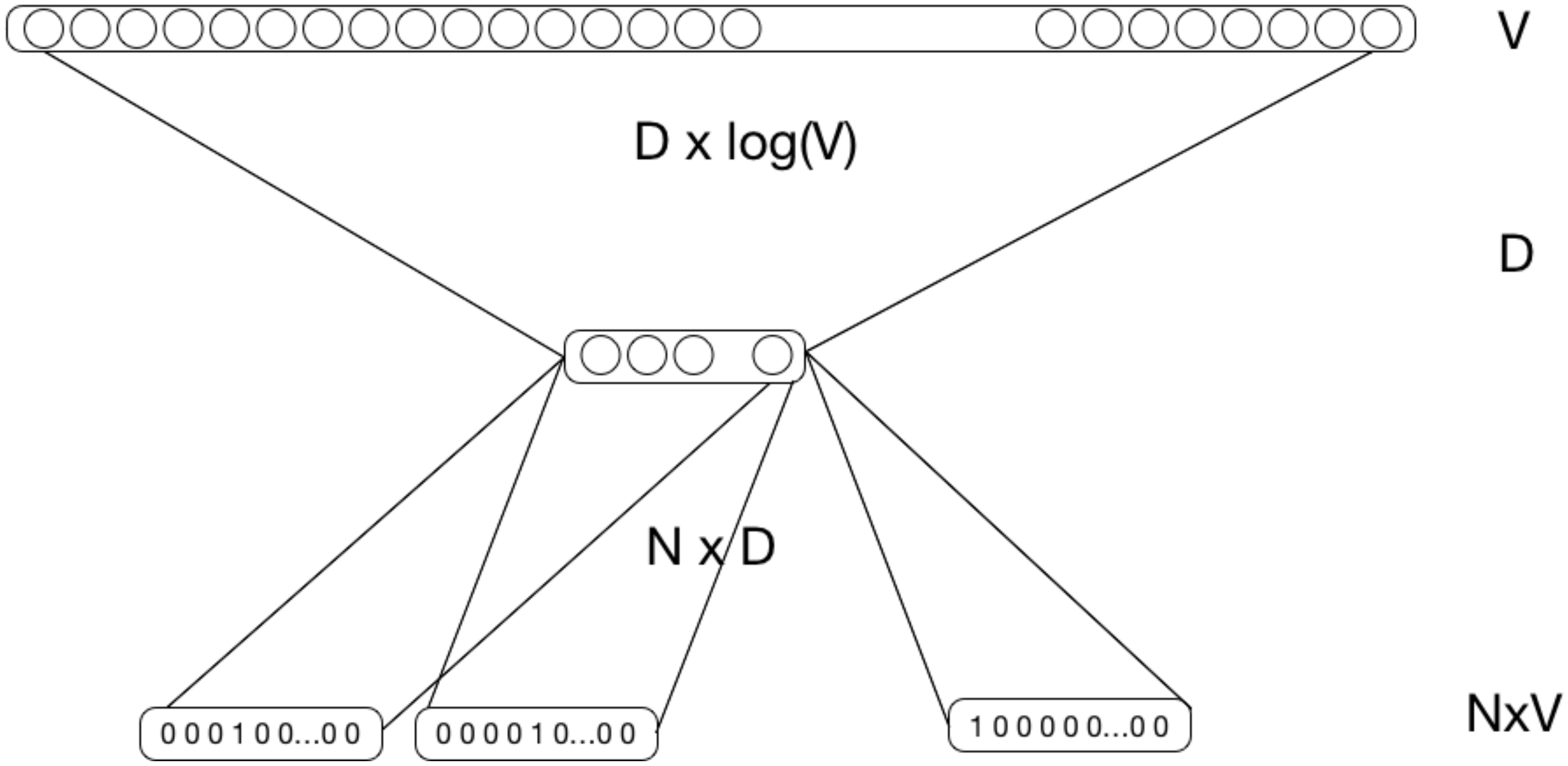
$$\log\left(1 + e^{-s(w_t,\, w_c)}\right) + \sum_{n \in \mathcal{N}_{t,c}} \log\left(1 + e^{s(w_t,\, n)}\right)$$

- Où $\mathcal{N}_{t,c}$ est un ensemble de mots (exemples négatifs) tirés du vocabulaire

Word2Vec

Approche 1 : Continuous Bag of word (CBOW)

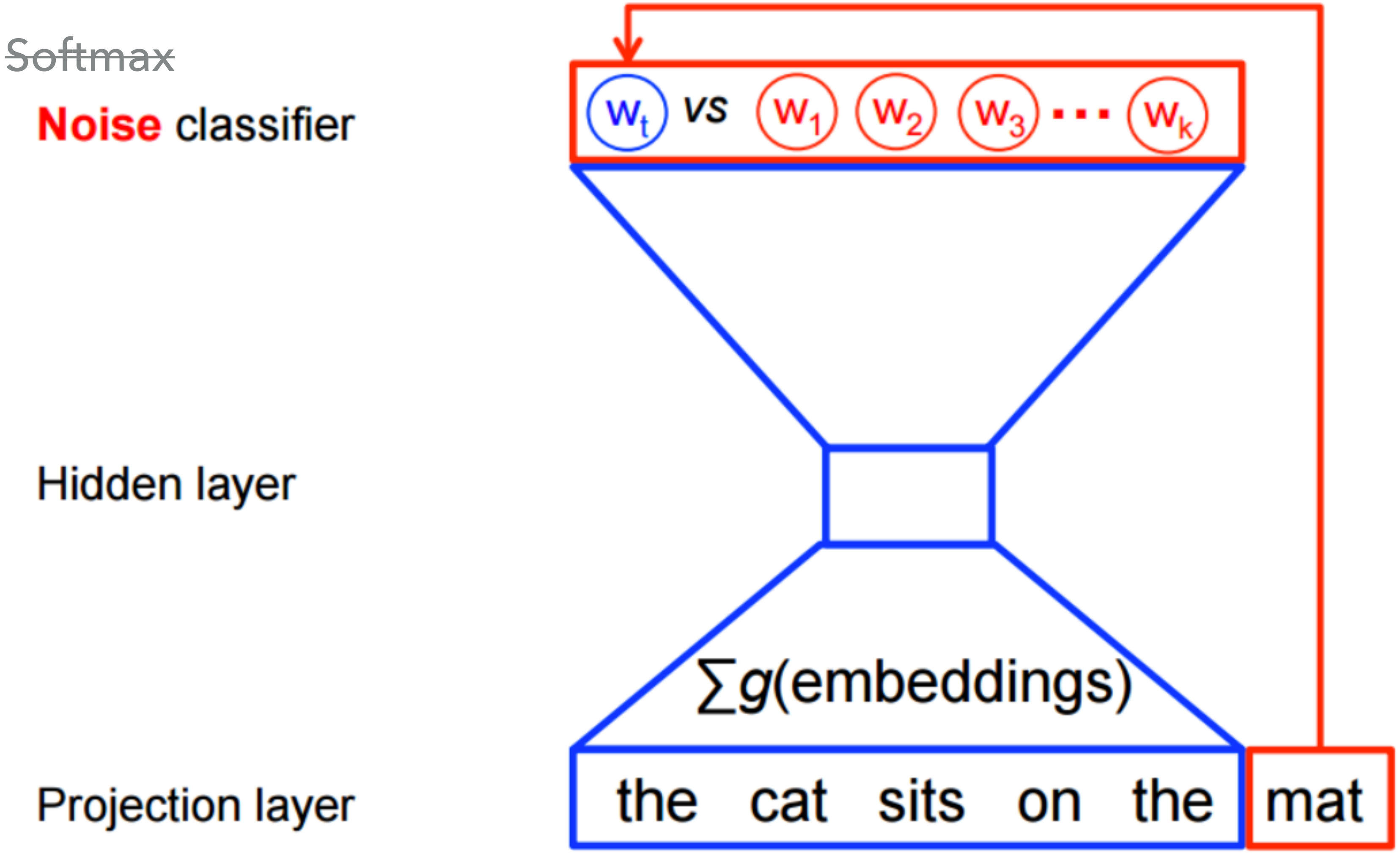
- Supprimer la couche cachée
- Sommer les contextes



Prédiction du mot courant en fonction du contexte droit et gauche

Word2Vec : Continuous Bag of word (CBOW)

Paramètres du modèles :



Exemples négatifs : nombre ?

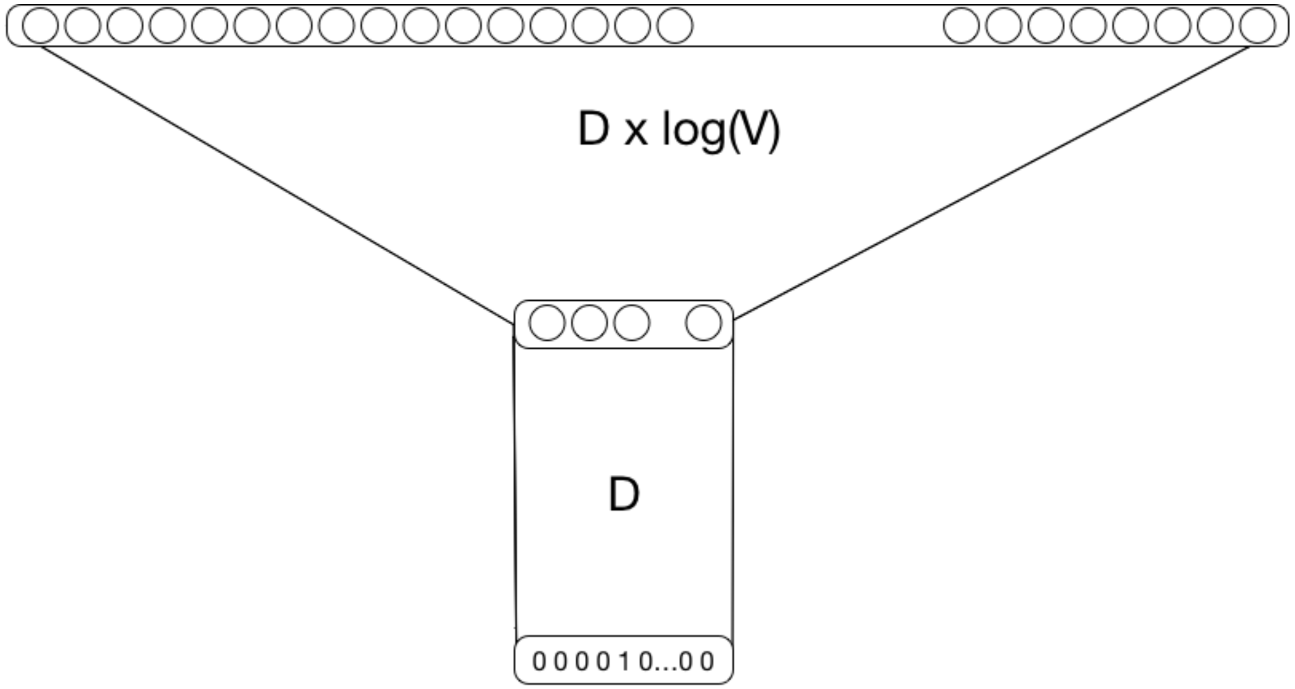
Embedding : taille ?

Contexte : gauche/droite, taille ?

Efficient Estimation of Word Representations in Vector Space

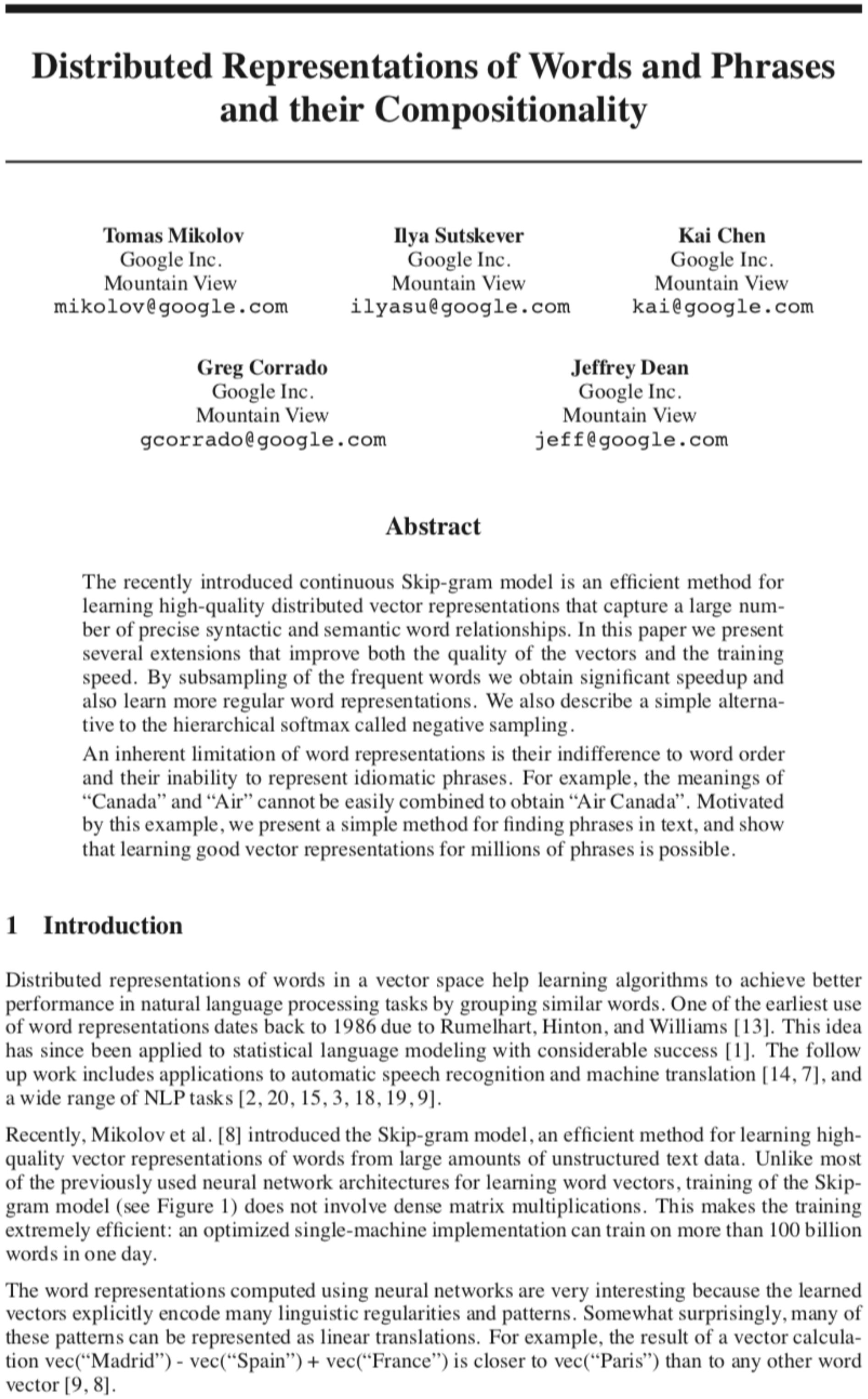
Approche 2 : Continuous Skip-Gram

- Supprimer la couche cachée
- Prédire chaque mot du contexte à partir du mot courant



Prédiction pour C contextes

Word2Vec



Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov
Google Inc.
Mountain View
mikolov@google.com

Ilya Sutskever
Google Inc.
Mountain View
ilyasu@google.com

Kai Chen
Google Inc.
Mountain View
kai@google.com

Greg Corrado
Google Inc.
Mountain View
gcorrado@google.com

Jeffrey Dean
Google Inc.
Mountain View
jeff@google.com

Abstract

The recently introduced continuous Skip-gram model is an efficient method for learning high-quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships. In this paper we present several extensions that improve both the quality of the vectors and the training speed. By subsampling of the frequent words we obtain significant speedup and also learn more regular word representations. We also describe a simple alternative to the hierarchical softmax called negative sampling.

An inherent limitation of word representations is their indifference to word order and their inability to represent idiomatic phrases. For example, the meanings of "Canada" and "Air" cannot be easily combined to obtain "Air Canada". Motivated by this example, we present a simple method for finding phrases in text, and show that learning good vector representations for millions of phrases is possible.

1 Introduction

Distributed representations of words in a vector space help learning algorithms to achieve better performance in natural language processing tasks by grouping similar words. One of the earliest use of word representations dates back to 1986 due to Rumelhart, Hinton, and Williams [13]. This idea has since been applied to statistical language modeling with considerable success [1]. The follow up work includes applications to automatic speech recognition and machine translation [14, 7], and a wide range of NLP tasks [2, 20, 15, 3, 18, 19, 9].

Recently, Mikolov et al. [8] introduced the Skip-gram model, an efficient method for learning high-quality vector representations of words from large amounts of unstructured text data. Unlike most of the previously used neural network architectures for learning word vectors, training of the Skip-gram model (see Figure 1) does not involve dense matrix multiplications. This makes the training extremely efficient: an optimized single-machine implementation can train on more than 100 billion words in one day.

The word representations computed using neural networks are very interesting because the learned vectors explicitly encode many linguistic regularities and patterns. Somewhat surprisingly, many of these patterns can be represented as linear translations. For example, the result of a vector calculation vec("Madrid") - vec("Spain") + vec("France") is closer to vec("Paris") than to any other word vector [9, 8].

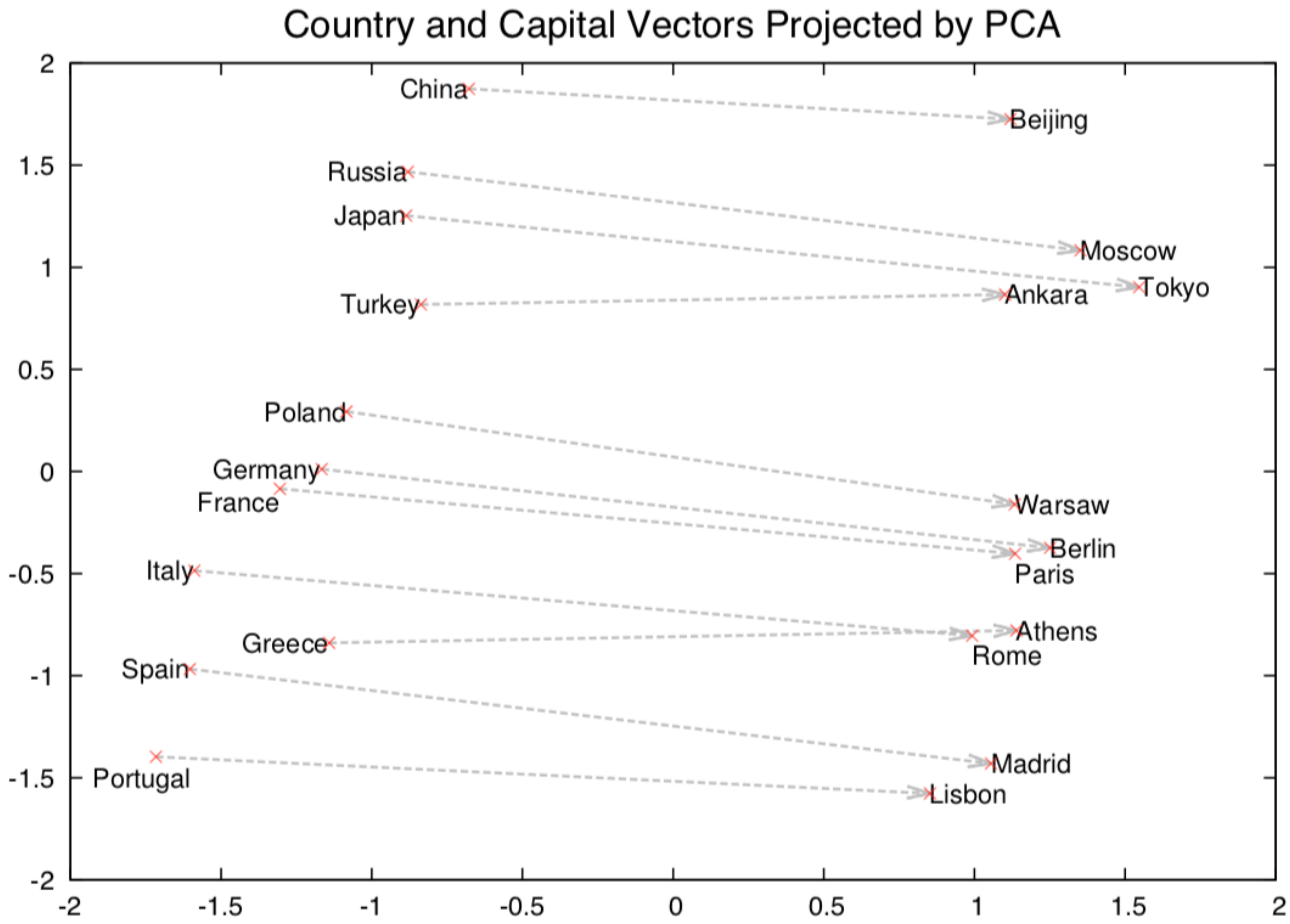
Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov
Ilya Sutskever
Kai Chen
Greg Corrado
Jeffrey Dean

NIPS, 2013

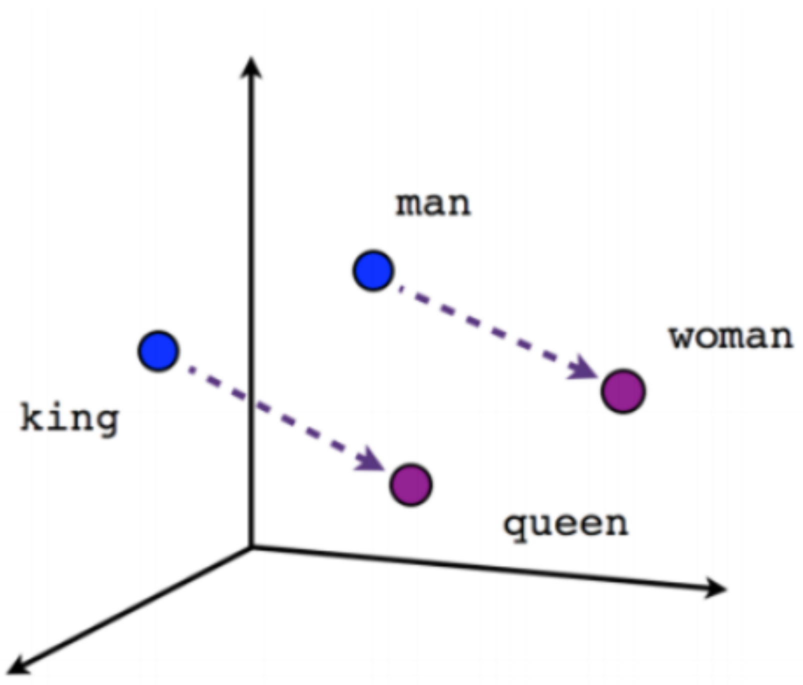
Word2Vec

Relations sémantiques et géométriques

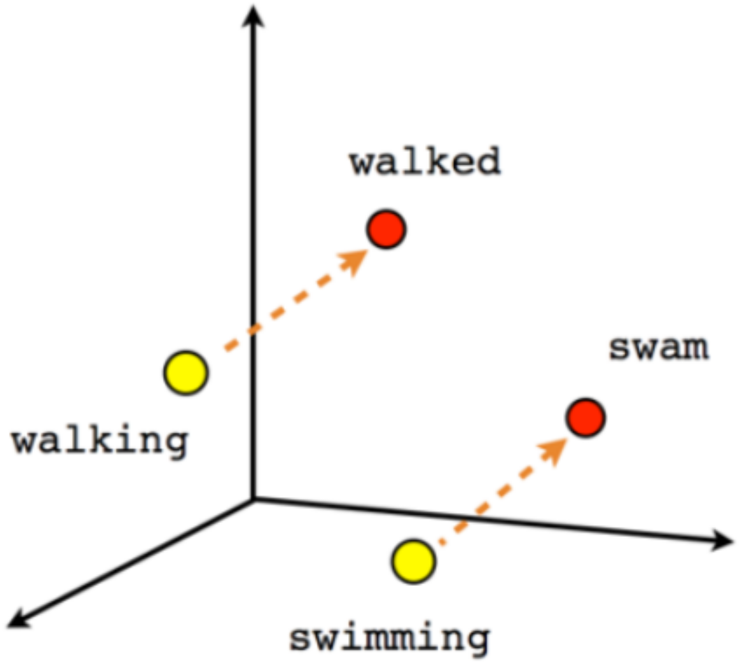


Word2Vec

Relations sémantiques et géométriques



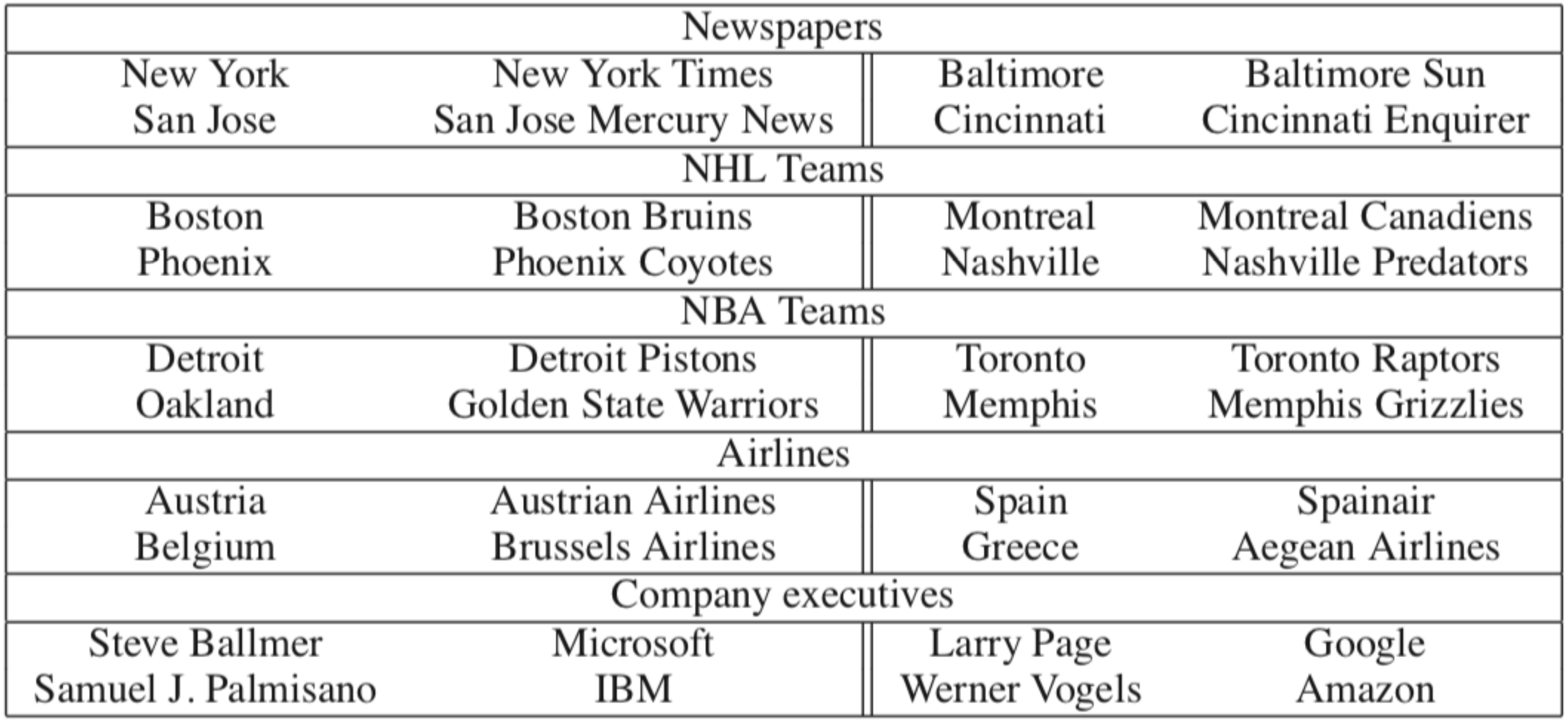
Male-Female



Verb tense

Word2Vec

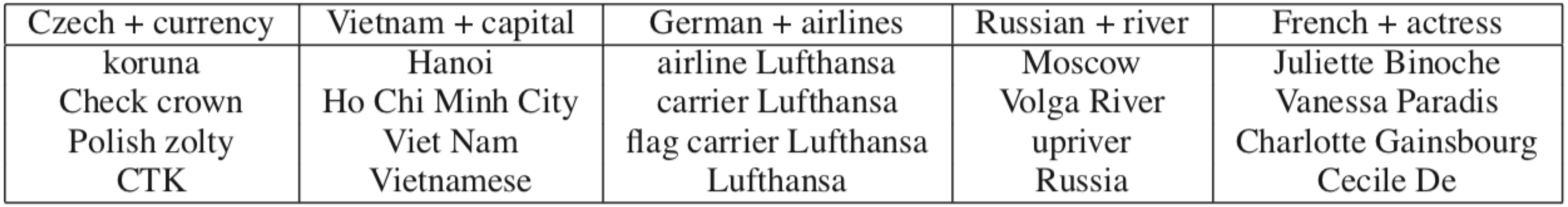
Raisonnement par analogie



Newspapers			
New York	New York Times	Baltimore	Baltimore Sun
San Jose	San Jose Mercury News	Cincinnati	Cincinnati Enquirer
NHL Teams			
Boston	Boston Bruins	Montreal	Montreal Canadiens
Phoenix	Phoenix Coyotes	Nashville	Nashville Predators
NBA Teams			
Detroit	Detroit Pistons	Toronto	Toronto Raptors
Oakland	Golden State Warriors	Memphis	Memphis Grizzlies
Airlines			
Austria	Austrian Airlines	Spain	Spainair
Belgium	Brussels Airlines	Greece	Aegean Airlines
Company executives			
Steve Ballmer	Microsoft	Larry Page	Google
Samuel J. Palmisano	IBM	Werner Vogels	Amazon

Word2Vec

Arithmétique vectorielle et sémantique



Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

Word2Vec

Limites :

- Pas de représentation pour les mots inconnus : mots rares, nom propres, néologismes, fautes d'orthographe, argot, jargon, erreur de reconnaissance (OCR, parole)
- Pas de paramètres partagés pour les différentes formes fléchies d'un mot

mange / mangerai

cheval /chevaux

Fast Text

Enriching Word Vectors with Subword Information

Piotr Bojanowski[*] and **Edouard Grave**[*] and **Armand Joulin** and **Tomas Mikolov**
Facebook AI Research
`{bojanowski,egrave,ajoulin,tmikolov}@fb.com`

Abstract

Continuous word representations, trained on large unlabeled corpora are useful for many natural language processing tasks. Popular models that learn such representations ignore the morphology of words, by assigning a distinct vector to each word. This is a limitation, especially for languages with large vocabularies and many rare words. In this paper, we propose a new approach based on the skipgram model, where each word is represented as a bag of character n-grams. A vector representation is associated to each character n-gram; words being represented as the sum of these representations. Our method is fast, allowing to train models on large corpora quickly and allows us to compute word representations for words that did not appear in the training data. We evaluate our word representations on nine different languages, both on word similarity and analogy tasks. By comparing to recently proposed morphological word representations, we show that our vectors achieve state-of-the-art performance on these tasks.

1 Introduction

Learning continuous representations of words has a long history in natural language processing (Rumelhart et al., 1988). These representations are typically derived from large unlabeled corpora using co-occurrence statistics (Deerwester et al., 1990; Schütze, 1992; Lund and Burgess, 1996). A large body of work, known as distributional semantics, has studied the properties of these methods (Turney et al., 2010; Baroni and Lenci, 2010). In the neural network community, Collobert and Weston (2008) proposed to learn word embeddings using a feedforward neural network, by predicting a word based on the two words on the left and two words on the right. More recently, Mikolov et al. (2013b) proposed simple log-bilinear models to learn continuous representations of words on very large corpora efficiently.

Most of these techniques represent each word of the vocabulary by a distinct vector, without parameter sharing. In particular, they ignore the internal structure of words, which is an important limitation for morphologically rich languages, such as Turkish or Finnish. For example, in French or Spanish, most verbs have more than forty different inflected forms, while the Finnish language has fifteen cases for nouns. These languages contain many word forms that occur rarely (or not at all) in the training corpus, making it difficult to learn good word representations. Because many word formations follow rules, it is possible to improve vector representations for morphologically rich languages by using character level information.

In this paper, we propose to learn representations for character n-grams, and to represent words as the sum of the n-gram vectors. Our main contribution is to introduce an extension of the continuous skipgram model (Mikolov et al., 2013b), which takes into account subword information. We evaluate this model on nine languages exhibiting different morphologies, showing the benefit of our approach.

[*] The two first authors contributed equally.

Enriching Word Vectors with Subword Information

Piotr Bojanowski
Edouard Grave
Armand Joulin
Tomas Mikolov

Transactions of the Association for Computational Linguistics, 2013

Fast Text

« Our main contribution is to introduce an extension of the continuous skip- gram model (Mikolov et al., 2013b), which takes into account subword information»

« Learn representationsfor character n-grams, and to represent words as the sum of the n-gram vectors »

Fonction à minimiser :

$$\sum_{t=1}^{T} \left[\sum_{c \in \mathcal{C}_t} \ell(s(w_t,\ w_c)) + \sum_{n \in \mathcal{N}_{t,c}} \ell(-s(w_t,\ n)) \right]$$

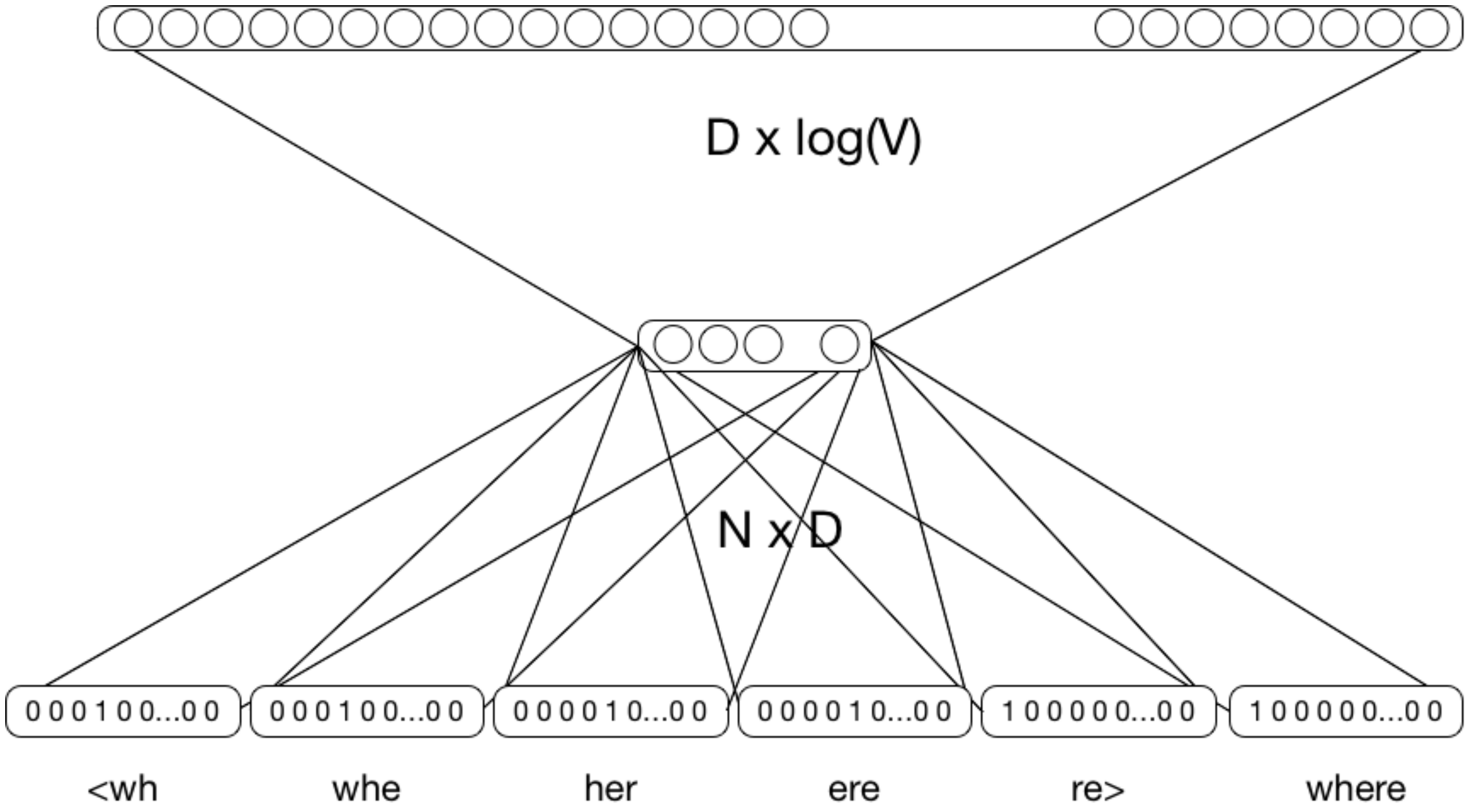
$$\ell : x \mapsto \log(1 + e^{-x})$$

$\mathcal{N}_{t,c}$: ensemble d'exemples négatifs tirés du vocabulaire

C_t : mots du contexte du mot cible *t*

Fast Text

Décomposition en ngrams :



Prédiction pour C contextes

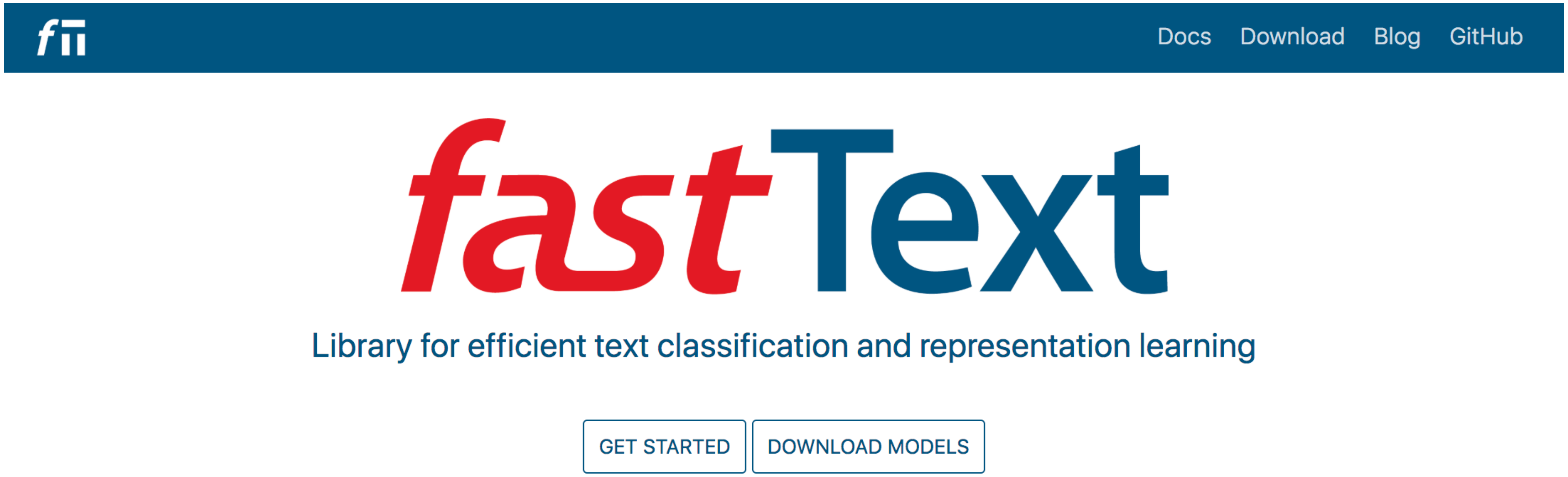
Paramètres :

- Dimension de l’embedding : 300 / Taille du contexte : entre 1 et 5 (aléatoire)
- Nombre d’exemples négatifs : 5, tirés selon la racine carré de leur fréquence

Fast Text

Les clés du succès :

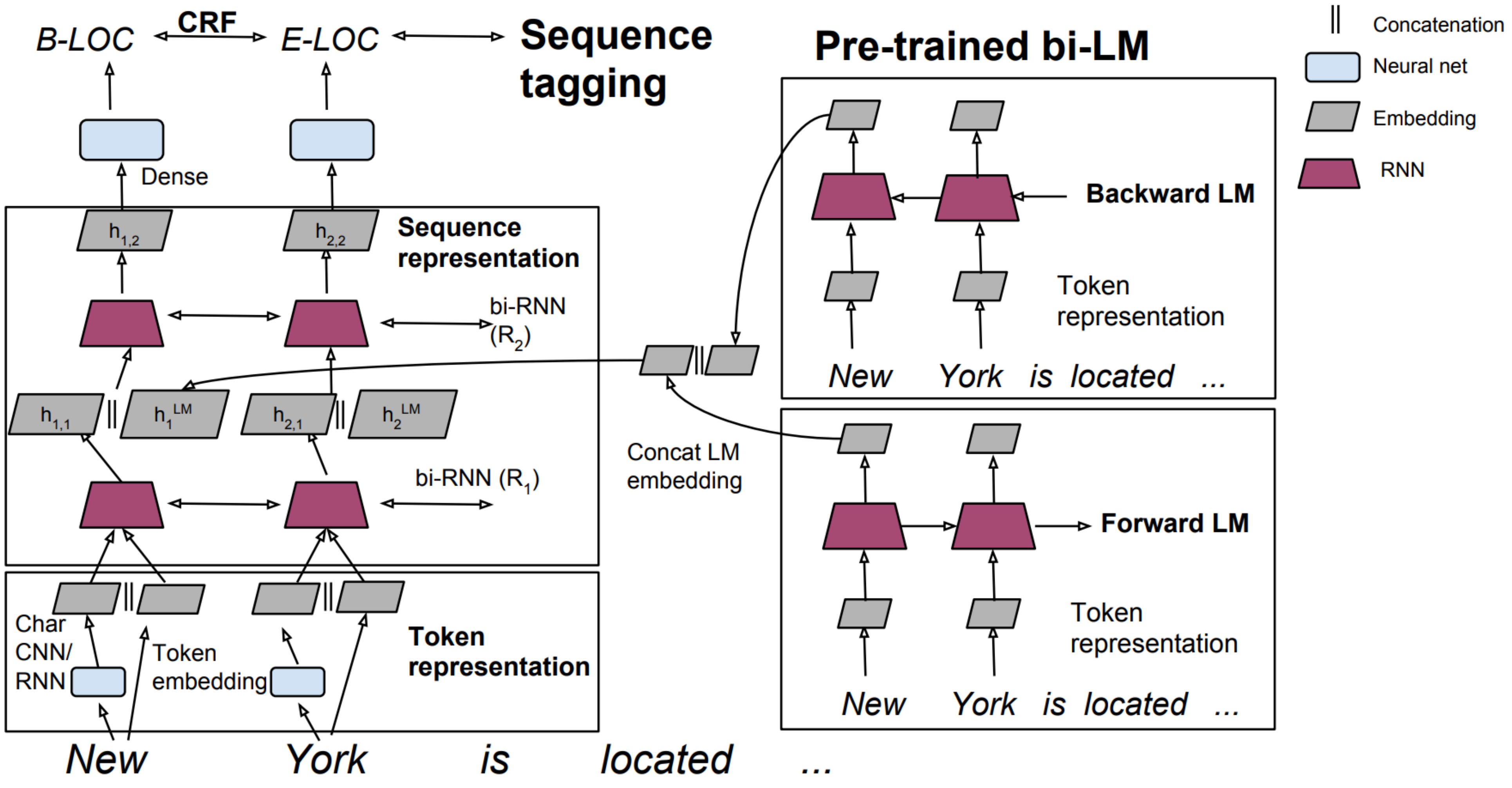
- Code open source (C++, python), documenté et rapide
- Modèles pré-entrainés disponibles en 157 langues
- … et un bonne communication



Library for efficient text classification and representation learning

Word embeddings modèles plus complexes

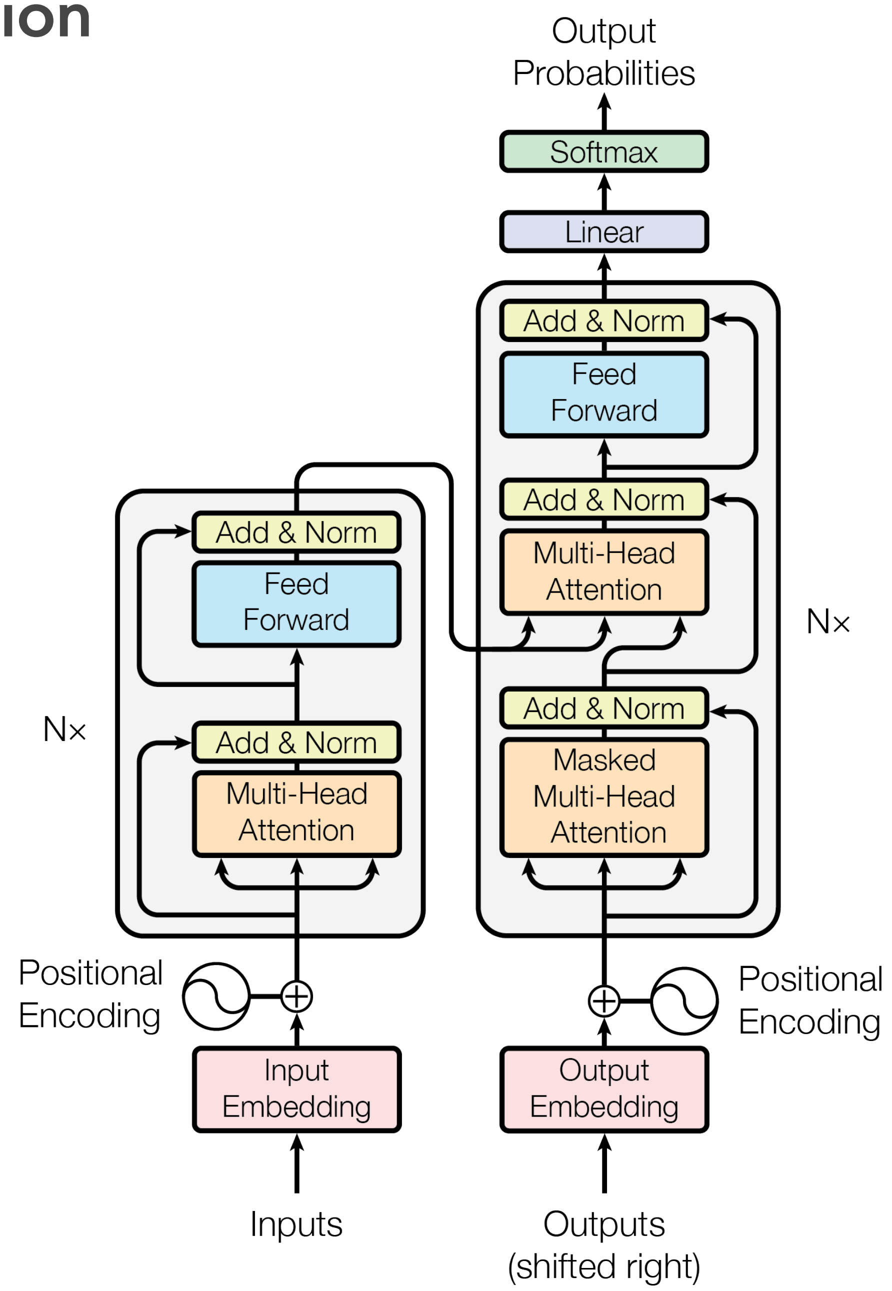
Avec réseaux de neurones récurrents



- Semi-supervised sequence tagging with bidirectional language models,Peters *et al.,* ACL 2017
- Deep Contextualized word representation, Peters *et al.*, NAACL 2018

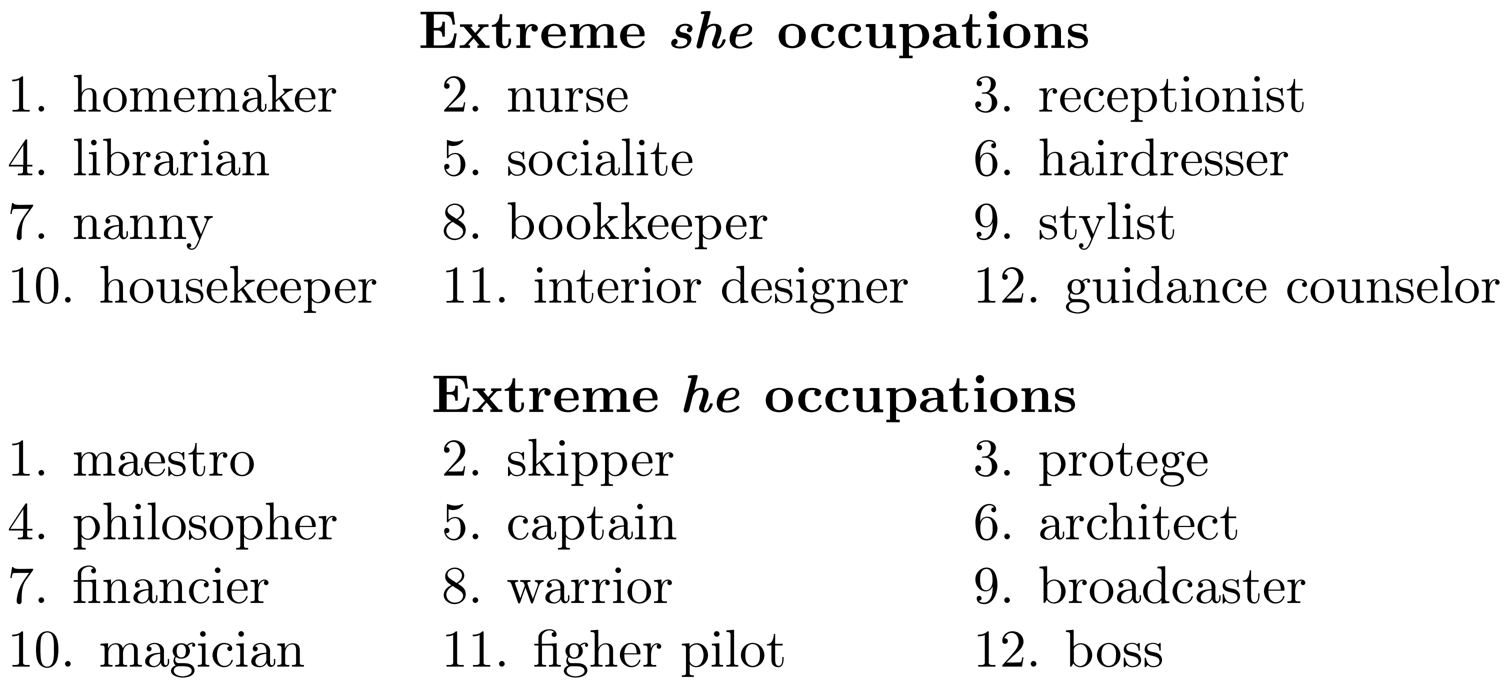
Word embeddings modèles plus complexes

Avec des modèles d'attention



- Attention is all you need, Vaswani *et al.*, NIPS 2017

Word embeddings : problème de biais



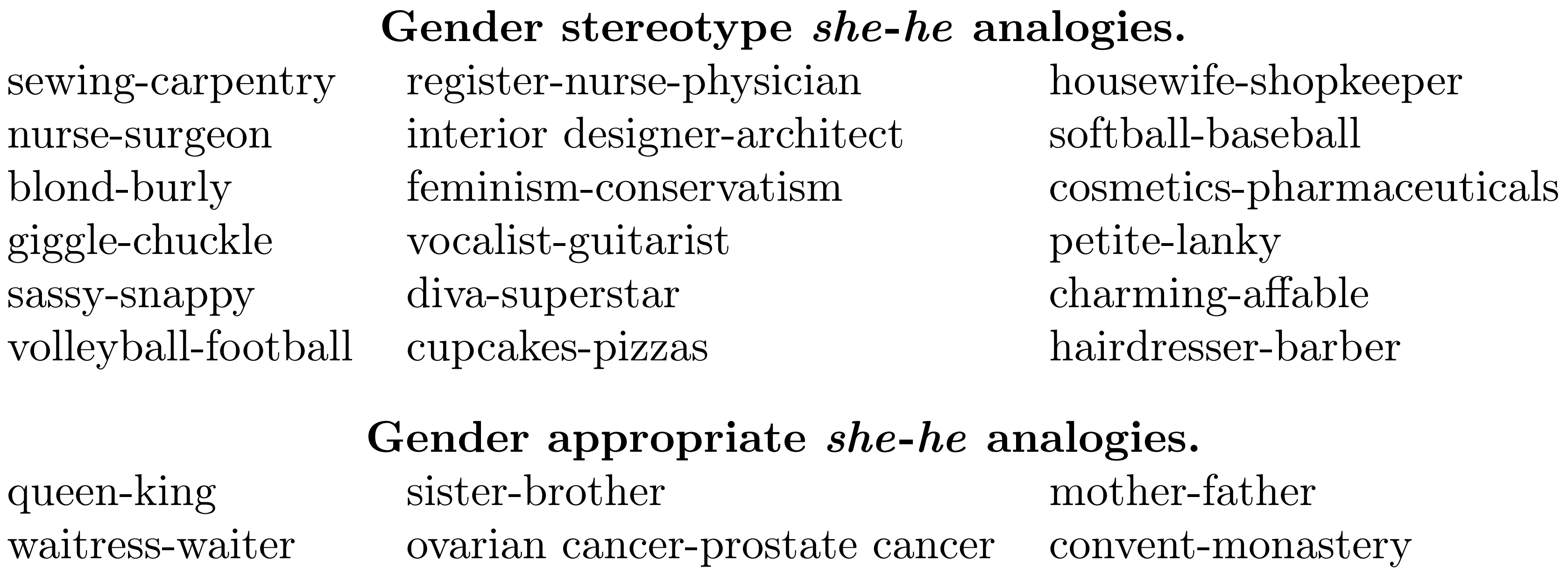
Extreme *she* occupations

1. homemaker
2. nurse
3. receptionist
4. librarian
5. socialite
6. hairdresser
7. nanny
8. bookkeeper
9. stylist
10. housekeeper
11. interior designer
12. guidance counselor

Extreme *he* occupations

1. maestro
2. skipper
3. protege
4. philosopher
5. captain
6. architect
7. financier
8. warrior
9. broadcaster
10. magician
11. figher pilot
12. boss

Biais de position sur l'axe homme-femmec



Gender stereotype *she*-*he* analogies.

sewing-carpentry
nurse-surgeon
blond-burly
giggle-chuckle
sassy-snappy
volleyball-football

register-nurse-physician
interior designer-architect
feminism-conservatism
vocalist-guitarist
diva-superstar
cupcakes-pizzas

housewife-shopkeeper
softball-baseball
cosmetics-pharmaceuticals
petite-lanky
charming-affable
hairdresser-barber

Gender appropriate *she*-*he* analogies.

queen-king
waitress-waiter

sister-brother
ovarian cancer-prostate cancer

mother-father
convent-monastery

Biais d'analogie

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings, Bolukbasi *et al*. NIPS 2016