# Introduction au Traitement Automatique des Langues

## 6 – Les niveaux de traitement – Le niveau Sémantique

# Introduction au traitement automatique des langues

Contenu de la matière :

1) Introduction Générale
2) Les applications du TAL
3) Les niveaux de traitement - Traitements de «bas niveau»
4) Les niveaux de traitement - Le niveau lexical
5) Les niveaux de traitement - Le niveau syntaxique
6) **Les niveaux de traitement - Le niveau sémantique**
7) Les niveaux de traitement - Le niveau pragmatique

# Plan du cours

1. **Définitions** : Sémantique et Analyse sémantique
2. **Concepts de base**: relations sémantiques, connotation, similarité, proximité, semantic frames, vector semantics & embeddings, mesures de similarité (cosinus).
3. **Représentation vectorielle et techniques** : Term-Document Matrix (Count Vectorizer, Bag-of-Words, N-grams), Term-Term (Co-Occurrence) Matrix, TF-IDF, Word Embeddings, Word2Vec, CBoW, Skip Gram.

# Définitions

- Une phrase comme: ***Le jardin de la porte mange le ciel.***
- Syntaxiquement parfaitement correcte, n'a pas de **sens** dans la plupart des contextes.
- Exemple 2 :

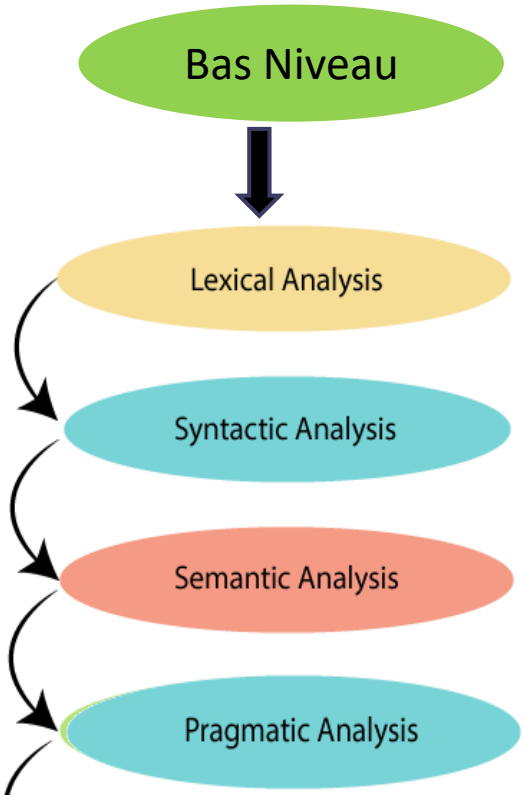
**Exemple de la polysémie en français**

**Je veux boire du café.**
**Je veux aller au café.**
**J'ai récolter du café.**

- Est-ce que le mot "café" veut la même chose dans les trois phrases ?
- Comment peut-on savoir le sens d'un mot dans ce cas ?
- Les sens du mot "café" :
  https://babelnet.org/search?word=caf%C3%A9&lang=FR

# Définitions

- La **sémantique** se préoccupe du **sens/signification** d'un texte.
- La sémantique est une discipline qui à pour objectif la description des significations propres aux langues.
- En TALN, la sémantique peut être définie comme l'étude de sens des mots, des phrases, et des énoncés.
- Analyse sémantique lexicale (sens des mots) et Analyse propositionnelle (sens des phrases).
- Le rôle de l'analyseur sémantique est donc d'attribuer un sens à la phrase structurée par l'analyseur syntaxique.



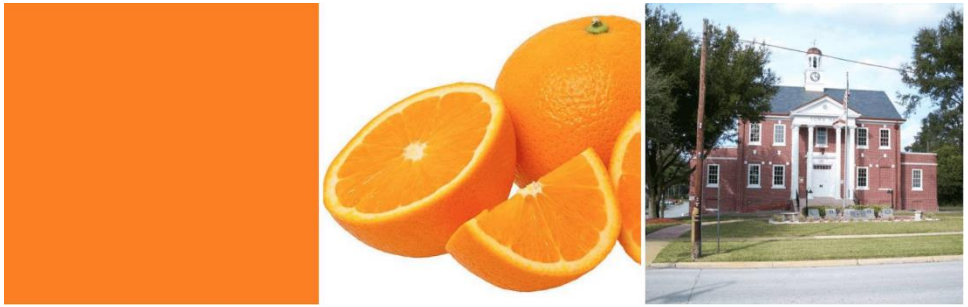
# Définitions

- Comment représenter le sens d'un texte ? – Se concentrer sur les mots.
- Les outils qui opèrent cette analyse sémantique font souvent appel à :

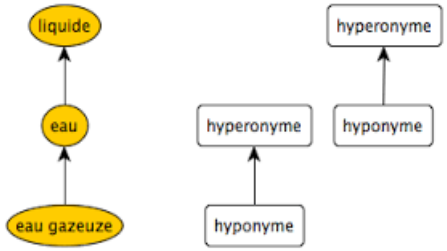
✓ De bases de données lexicales, permettant de classer chaque terme dans une arborescence de concepts pour déterminer les thèmes dominants d'un texte. Ex: WordNet, VerbNet, FrameNet, BabelNet.

✓ Ainsi qu'à des algorithmes complexes (Machine Learning, Deep Learning) permettant d'évaluer les relations entre les différentes idées d'un texte.

✓ Aussi, quelques sous-tâches impliquées, y compris : Word Sense Disambiguation (WSD) et Relationship Extraction.



# Concepts de base




**Relations Sémantiques** entre mots

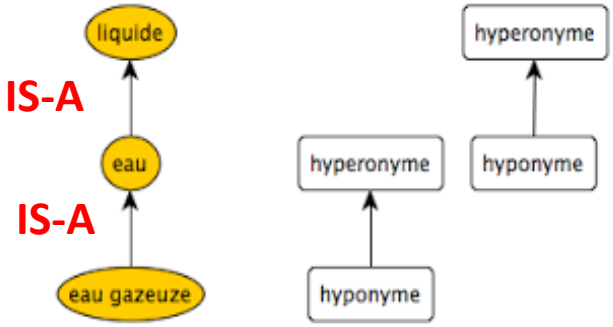
- **Synonyme:** Avoir des sens similaires dans un contexte donné. Si on substitue un mot par un autre dans une phrase sans changer le sens.
- **Antonyme**: avoir des sens opposés dans un contexte donné. Les deux mots doivent exprimer deux valeurs d'une même propriété. Exemple, grand et petit expriment la propriété taille.
- **Hyponyme**: un mot ayant un sens plus spécifique qu'un autre. Exemple, chat, tigre, lion sont l'hyponyme de félin.
- **Hyperonyme**: un mot avec un sens plus générique. Exemple, félin est l'hyperonyme de chat, tigre, lion.
- **Méronyme:** un mot qui désigne une partie d'un autre. Exemple, roue est un méronyme de voiture.

# Concepts de base

**Relations Sémantiques** entre mots

- **Hyponyme**: un mot ayant un sens plus spécifique qu'un autre. Exemple, chat, tigre, lion sont l'hyponyme de félin.
- **Hyperonyme**: un mot avec un sens plus générique. Exemple, félin est l'hyperonyme de chat, tigre, lion. Entraîne un relation **IS-A.**
- **Méronyme:** un mot qui désigne une partie d'un autre. Exemple, roue est un méronyme de voiture.

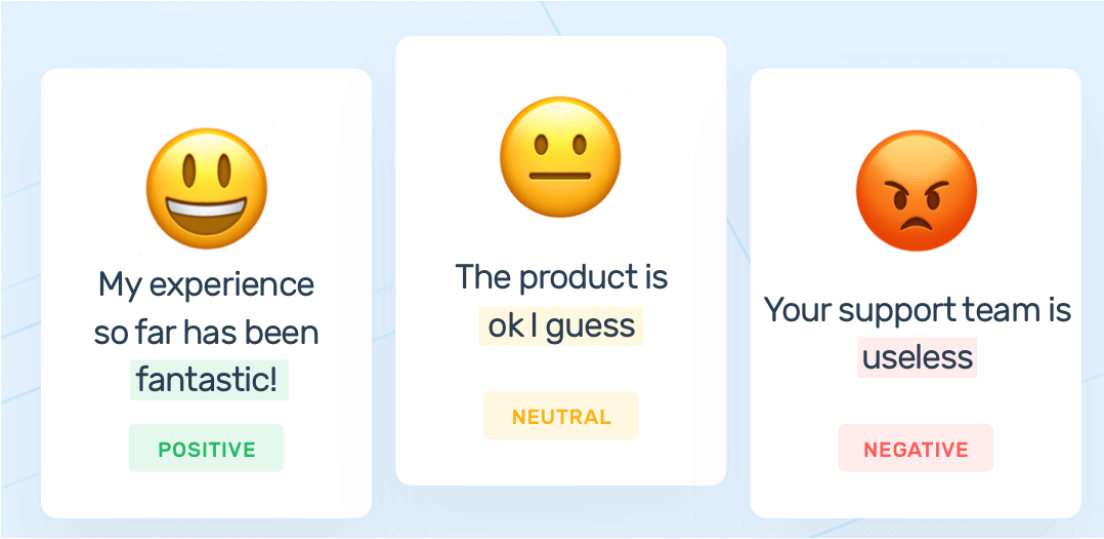
**Les relations taxonomiques (de classification)**



# Concepts de base

**Connotation** des mots

- Les mots ont des significations ou des connotations affectives.
- Liée aux émotions d'un écrivain ou d'un lecteur, sentiments, opinions ou évaluations.
- Par exemple, certains mots ont des connotations positives (heureux) tandis que d'autres ont des connotations négatives (triste). Analyse des sentiments.



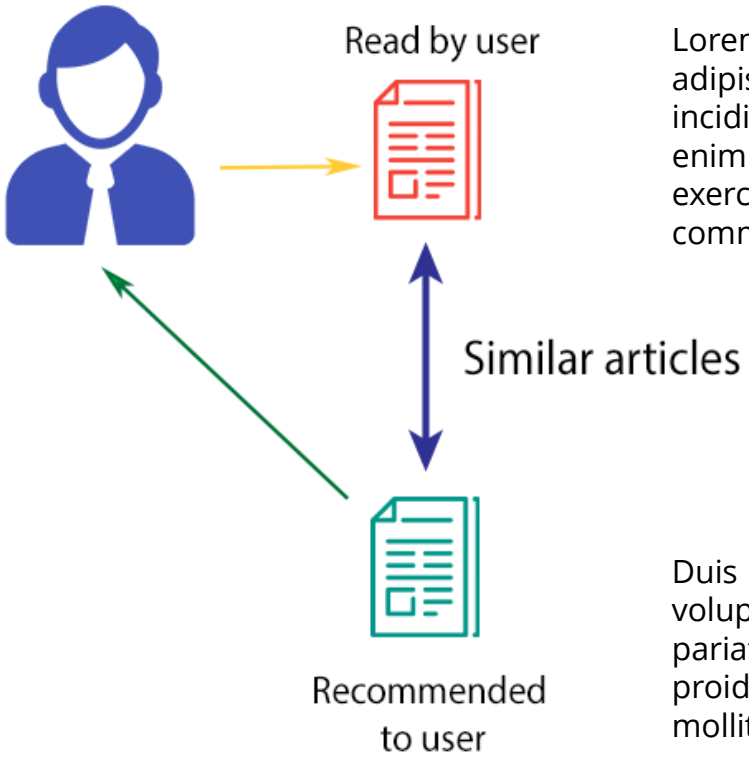
# Concepts de base

**Similarité** entre les mots

- "Words that occur in similar contexts tend to have similar meanings."
- Bien que les mots n'aient pas beaucoup de synonymes, la plupart des mots ont beaucoup de mots similaires.
- Chat n'est pas synonyme de chien, mais chats et chiens le sont des mots identiques (similaires).
- En passant de la synonymie à la similarité, on passe des relations entre les sens des mots (comme la synonymie) aux relations entre les mots (comme la similarité).
- Savoir à quel point deux mots sont similaires peut aider à calculer à quel point le sens de deux expressions ou phrases est similaire.

# Concepts de base

**Similarité** entre les mots




Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Concepts de base

**Proximité** entre les mots

- = Word Relatedness.
- Le sens de deux mots peut être lié autrement que par similarité. Une telle classe de relation est appelée Relatedness.
- Exemple : les mots Café et Tasse, ne sont pas similaires, mais sont liés.
- Un type courant de Relatedness entre les mots est s'ils appartiennent au même champ sémantique: Semantic Field.
- Un champ sémantique est un ensemble de mots qui couvrent un domaine sémantique particulier et entretiennent entre eux des relations structurées.
- Exemple: champ sémantique des hôpitaux (infirmière, chirurgien, scalpel, infirmière, anesthésique, hôpital).

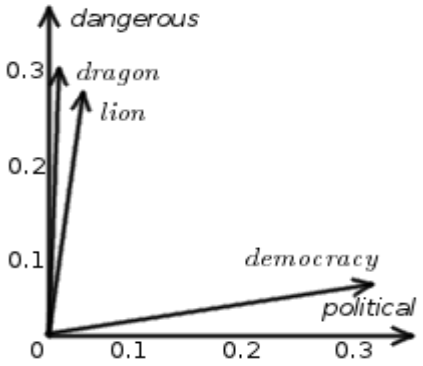
# Concepts de base

## Semantic Frames and Roles

- = Cadre sémantique. Étroitement liée aux Semantic Field.
- Un cadre sémantique est un ensemble de mots qui dénotent des perspectives ou des participants à un type particulier d'événement.
- Exemple: une *transaction commerciale*. Cet événement peut être encodé lexicalement en utilisant des verbes comme *acheter*, *vendre*, *payer*, ou des noms comme *acheteur*, etc.
- Les cadres ont des rôles sémantiques (comme acheteur, vendeur, marchandises, argent), et les mots d'une phrase peuvent prendre ces rôles.
- Savoir que *acheter* et *vendre* ont cette relation permet à un système de savoir qu'une phrase telle que *Sam a acheté le livre à Ling* pourrait être paraphrasée comme *Ling a vendu le livre à Sam*, et que Sam a le rôle de l'acheteur dans le cadre et Ling le vendeur.

# Concepts de base

## Vector semantics




- = Sémantique vectorielle. Est la manière standard de représenter le sens des mots en TALN. => Vectorisation.

- Définit la sémantique et interprète le sens des mots pour expliquer des caractéristiques telles que la similarité des mots.

➢ Les mots qui apparaissent dans des contextes similaires ont tendance à avoir des sens similaires.

➢ Le sens d'un mot est lié à la distribution (occurrence) des mots autour de lui.

- La sémantique vectorielle représente un mot dans un espace vectoriel multidimensionnel.

# Concepts de base

## Vector semantics

- Quel est le sens du mot Ongchoy ?
- Contexte :
  - ✓ Ongchoy sauté à l'ail est délicieux.
  - ✓ Ongchoy est superbe sur du riz.
  - ✓ … Feuilles d'Ongchoy aux sauces salées …
- Sachant :
  - ✓ … épinards sautés à l'ail sur du riz …
  - ✓ … le chou vert et autres légumes-feuilles salés

# Concepts de base

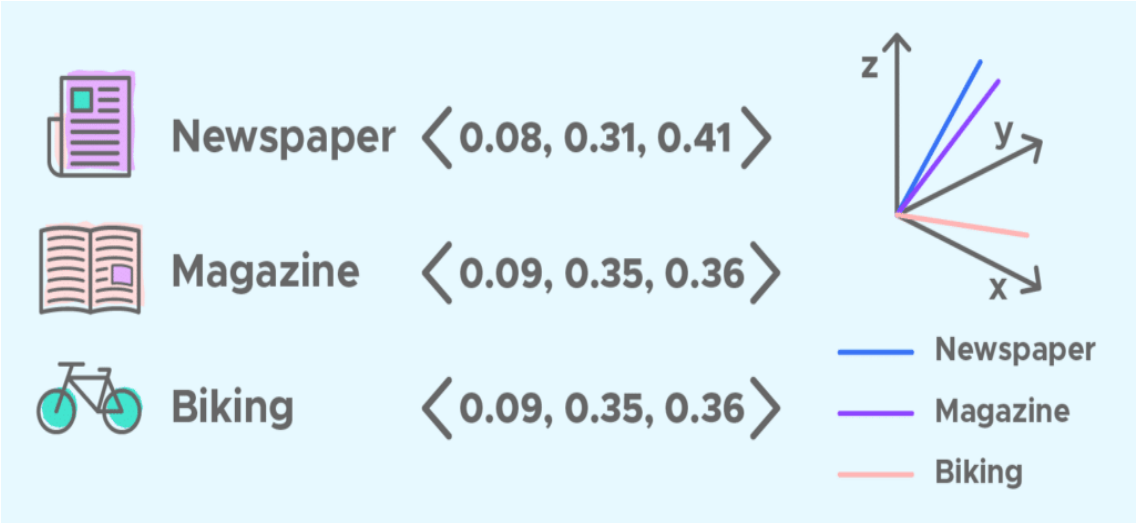
## Vector semantics

- Quel est le sens du mot Ongchoy ?
- Le fait que Ongchoy se produise avec des mots comme riz et ail et délicieux et salé, tout comme des mots comme épinards et chou vert pourrait suggérer que Ongchoy est un vert feuillu semblable à ces autres légumes verts feuillus.
- Nous pouvons faire la même chose automatiquement en comptant simplement les mots dans le contexte d' Ongchoy.
- Cela peut nous aider à découvrir la similarité entre ces mots et le mot Ongchoy.

# Concepts de base

**Vector semantics & Embeddings**

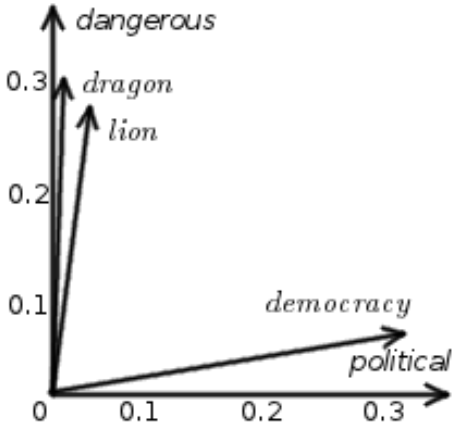
- Le modèle vectoriel est également appelé Embeddings, en raison du fait que le mot est intégré dans un espace vectoriel particulier.
- La représentation d'un mot sous forme d'un vecteur est appelé Embedding.
- Les mots apparaissant dans des contextes similaires possèdent des vecteurs correspondants qui sont relativement proches. => Distance entre mots.



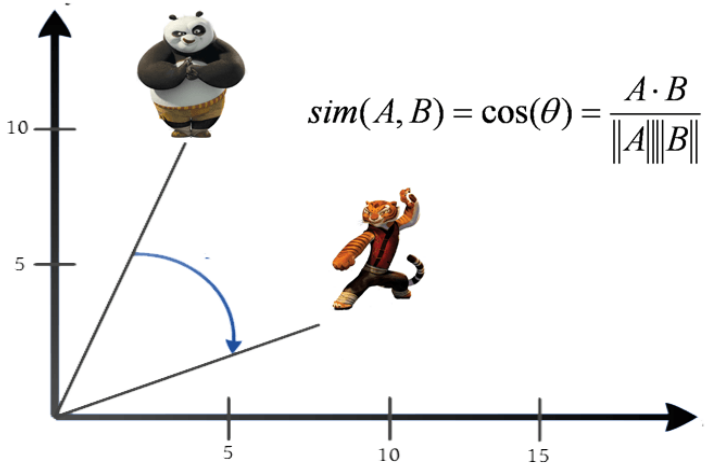
# Concepts de base

## Mesures de Similarité

- Mesure la distance entre les mots, les phrases, les énoncés, etc.
- Similarité distributionnelle et similarité sémantique.
- Mesures de distribution : TF-IDF, OKAPI BM25, LIKEY, etc.
- Mesures de similarité : Cosinus, Jaccard, Levenshtein, Manhattan, WordNet, conceptuelle ontologie, etc.




Cosine Similarity



# Représentation vectorielle

La **vectorisation du texte** est le processus de conversion de texte en vecteurs numériques. Il peut y avoir différentes représentations numériques vectorielles du même texte.

- Les modèles vectoriels ou distributionnels sont généralement basés sur une matrice de co-occurrence, une manière de représenter la fréquence à laquelle les mots apparaissent ensemble simultanément.

- Deux matrices de co-occurrence populaires : la matrice terme-document et la matrice terme-terme.

  - Terme-document : On représente un mot (ligne) par les documents (colonne) qui le contiennent (ou l'inverse).

  - Terme-terme : On représente un mot par d'autres mots.

- Techniques alternatives nouvelles qui prennent en compte les relations entre les mots: la conceptualisation et le plongement lexical (word embeddings).

# Représentation vectorielle

La **vectorisation du texte** est le processus de conversion de texte en vecteurs numériques. Il peut y avoir différentes représentations numériques vectorielles du même texte.

- Types:

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

Ex : One-Hot, N-grams, BoW, TF-IDF, PMI, Count Vectorizer, co-occurrence matrix, etc.

**New Age Techniques**
**Prediction / Neural Network based vectorization approach**

Ex : Word2Vec, CBoW, Skip Gram, Glove, FastText, ELMo, BERT, XLNet, etc.


Fig. Evolution of NLP Models

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## One-Hot Encoding

- La représentation vectorielle la plus classique des mots.
- La méthode consiste à représenter chaque mot du vocabulaire sous forme de vecteur binaire de dimension = taille du vocabulaire, qui a toutes ses valeurs nulles (=0) à l'exception de l'index du mot (=1).
- Cet encodage ne capture pas les relations entre les différents mots. Par conséquent, il ne transmet pas d'informations sur le contexte.

***Example:***

***Doc. 1***: *They are playing football.*
***Doc. 2***: *They are playing cricket.*

***Vocab.***: *[They, are, playing, football, cricket]*

*Onehot('football') = <0, 0, 0, 0, 1>*

| They | are | playing | cricket | football |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## One-Hot Encoding

Cet encodage ne capture pas les relations entre les différents mots. Par conséquent, il ne transmet pas d'informations sur le contexte.

Dim = |V| (v is the size of vocabulary)

- motel 


- hotel 


If you search for [Seattle motel] key word, we want the search engine to match web page containing "Seattle hotel"

Similarity(motal, hotel) = 0

motel 

T

hotel 

= 0

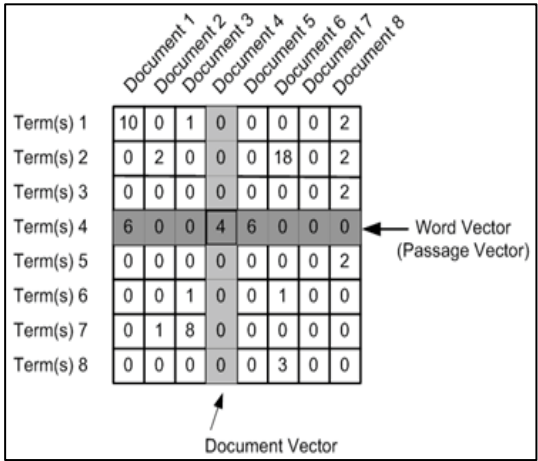
If we do inner product with the above vectors, we can not find out similarity between words

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice de Co-occurrence Terme-Document

- **Count Vectorizer, Bag-of-Words (BoW),** Co-Occurrence matrix.
- Elle a d'abord été défini dans le cadre du modèle d'espace vectoriel (**Vector Space Model**) en recherche d'informations.
- Il crée une **matrice** d'occurrence **terme-document**, qui est un ensemble de valeurs indiquant si un mot particulier apparaît dans le document.
- Chaque ligne de la matrice représente un mot du vocabulaire et chaque colonne représente un document du corpus. Ou l'inverse.
- Les cellules de la matrice indiquent la fréquence du mot dans un document, également appelée fréquence de terme.



# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice Terme-Document

- **Count Vectorizer, Bag-of-Words (BoW)**

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

**Figure 6.2** The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice Terme-Document

- **Count Vectorizer, Bag-of-Words (BoW)**

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

**Figure 6.2** The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

**Figure 6.3** The term-document matrix for four words in four Shakespeare plays. The red boxes show that each document is represented as a column vector of length four.

En rouge, les vecteur de chaque document = l'espace vectoriel

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice Terme-Document

- **Count Vectorizer, Bag-of-Words (BoW)**

Deux documents similaires auront tendance à avoir des mots similaires, et si deux documents ont des mots similaires, leurs vecteurs auront tendance à être similaires.

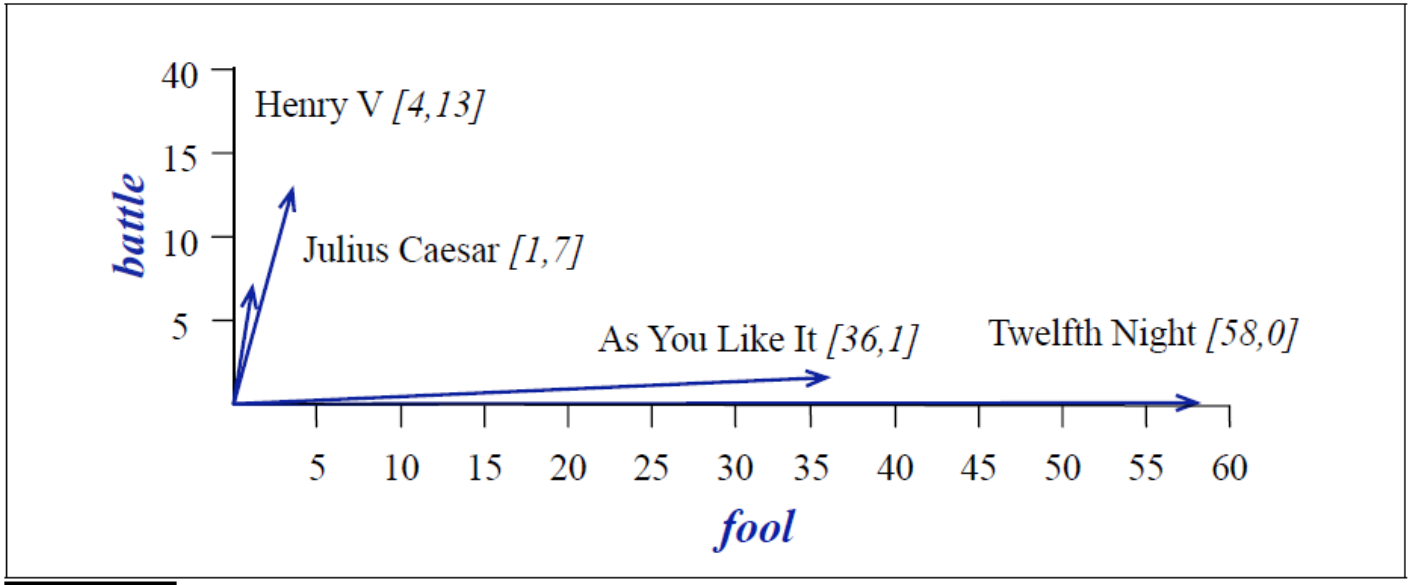



**Figure 6.4** A spatial visualization of the document vectors for the four Shakespeare play documents, showing just two of the dimensions, corresponding to the words *battle* and *fool*. The comedies have high values for the *fool* dimension and low values for the *battle* dimension.

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice Terme-Document

- **Count Vectorizer, Bag-of-Words (BoW)**

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

**Figure 6.2** The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

**Figure 6.5** The term-document matrix for four words in four Shakespeare plays. The red boxes show that each word is represented as a row vector of length four.

En rouge, les vecteur de chaque terme = l'espace vectoriel

# Représentation vectorielle

**Traditional Techniques**
Frequency-based or Statistical based vectorization approach

## Matrice Terme-Document

- **Count Vectorizer, Bag-of-Words (BoW)**

```
this burger is very tasty and affordable.
this burger is not tasty and is affordable.
this burger is very very delicious.
```

Docs

```
words: ["and", "affordable.", "delicious.",  "is", "not", "burger", "tasty", "this", "very"]
```

Terms

| | and | affordable | delicious | is | not | pasta | tasty | this | very |
|---|---|---|---|---|---|---|---|---|---|
| this pasta is very tasty and affordable. | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| this pasta is not tasty and is affordable | 1 | 1 | 0 | 2 | 1 | 1 | 1 | 1 | 0 |
| this pasta is very very delicious. | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 2 |

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice Terme-Document

- **N-grams vectorization**

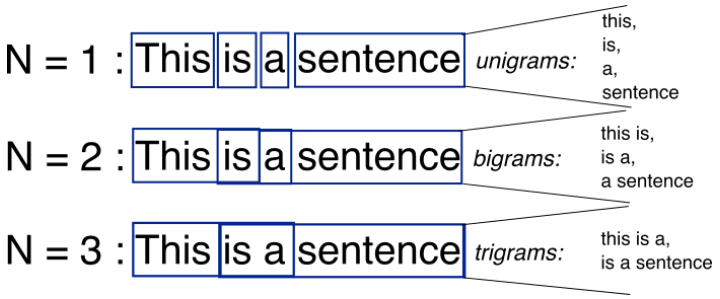
- Semblable à la technique Count Vectorizer, une matrice terme-document est générée et chaque cellule représente la fréquence.

- Les colonnes représentent toutes les colonnes de mots adjacents de longueur n.

- Count vectorizer est un cas particulier de N-Gram où **n**=1.

- Les N-grammes considèrent la séquence de n mots dans le texte ; où n est (1,2,3.. ) comme 1-gramme, 2-gramme. pour la paire de tokens. Contrairement à BoW, il maintient l'ordre des mots.

| Text | N-gram |
|---|---|
| Data | 1-gram |
| Great information | 2-gram |
| I am fine | 3-gram |
| Nice to meet you | 4-gram |



# Représentation vectorielle

**Traditional Techniques**
Frequency-based or Statistical based vectorization approach

## Matrice Terme-Document

- **N-grams vectorization**

**This is Big Data AI Book**

| | | | | | | |
|---|---|---|---|---|---|---|
| N=1 | *Uni-Gram* | This | Is | Big | Data | AI | Book |
| N=2 | *Bi-Gram* | This is | Is Big | Big Data | Data AI | AI Book | |
| N=3 | *Tri-Gram* | This is Big | Is Big Data | Big Data AI | Data AI Book | | |

# Représentation vectorielle

**Traditional Techniques**
Frequency-based or Statistical based vectorization approach

## Matrice Terme-Document

- **N-grams vectorization**

Bi-grams (2-grams) vectorization

| |
|---|
| good movie |
| not a good movie |
| did not like |

| good movie | movie | did not | a | ... |
|---|---|---|---|---|
| 1 | 1 | 0 | 0 | ... |
| 1 | 1 | 0 | 1 | ... |
| 0 | 0 | 1 | 0 | ... |

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice de Co-occurrence Terme-Terme

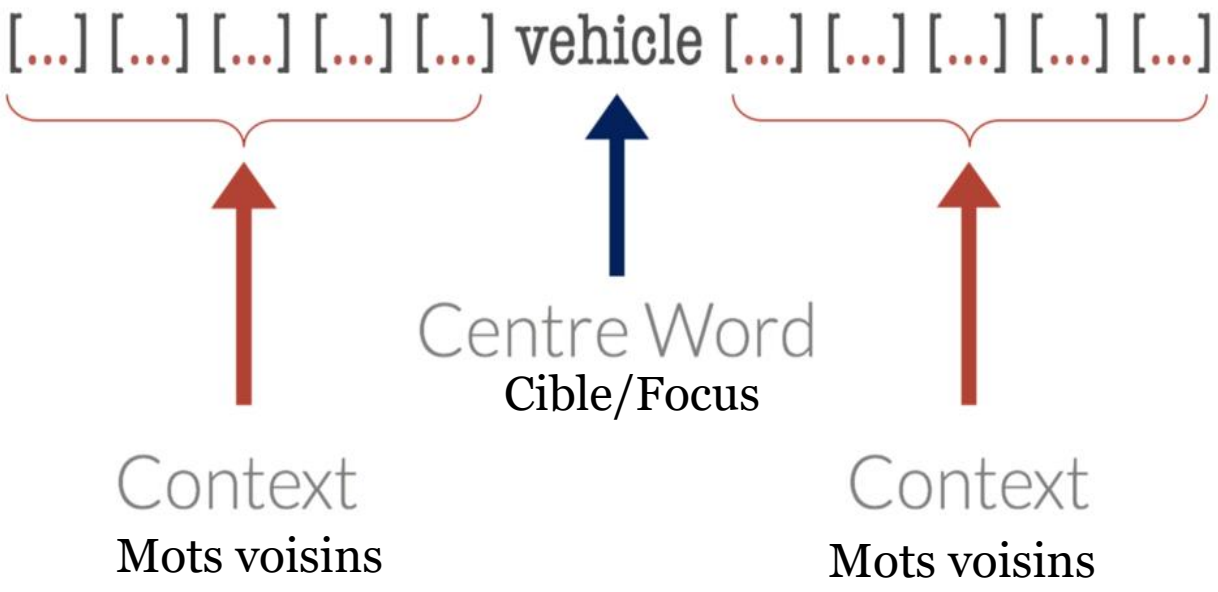
- Matrice Mot-Mot, matrice Terme-Contexte, Co-Occurrence matrix.
- Lignes et colonnes représentent les termes.
- On peut représenter un mot par rapport aux autres mots du vocabulaire (= contexte) en utilisant la co-occurrence.
- Chaque cellule enregistre la fréquence que le mot de ligne (cible) et le mot de colonne (contexte) co-apparaissent dans un certain contexte dans un corpus d'entrainement.
- La co-occurrence est calculée par rapport aux fenêtres autour du mot. La fenêtre peut être par exemple 4 mots avant et 4 mots après.

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice Terme-Terme

- La co-occurrence est calculée par rapport aux fenêtres autour du mot. La fenêtre peut être par exemple 4 mots avant et 4 mots après.



# Représentation vectorielle

**Traditional Techniques**
Frequency-based or Statistical based vectorization approach

## Matrice Terme-Terme

- La co-occurrence est calculée par rapport aux fenêtres autour du mot. La fenêtre peut être par exemple 4 mots avant et 4 mots après.

: Center Word
: Context Word

c=0 The cute cat jumps over the lazy dog.

c=1 The cute cat jumps over the lazy dog.

c=2 The cute cat jumps over the lazy dog.

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice Terme-Terme

- La co-occurrence peut être calculée par rapport aux documents, aux phrases ou des fenêtres autour du mot. La fenêtre peut être 4 mots avant et 4 mots après.
- Exemple: Fenêtre 2-2 / Context Window of 2

```
Sentence: Quick Brown Fox Jump Over The Lazy Dog
```

Quick Brown Fox Jump Over The Lazy Dog

The green words are a 2 (around) context window for the word 'Fox' and only these green words are used for calculating the co-occurrence. Let us see the context window for the word 'Over'.

Quick Brown Fox Jump Over The Lazy Dog

# Représentation vectorielle

**Traditional Techniques**
Frequency-based or Statistical based vectorization approach

## Matrice Terme-Terme

- Example corpus:
  - I like deep learning.
  - I like NLP.
  - I enjoy flying.

*Fenêtre 1-1 / Context Window of length 1*

*Vector('like') = <2, 0, 0, 1, 0, 1, 0>*

| counts | I | like | enjoy | deep | learning | NLP | flying |
|---|---|---|---|---|---|---|---|
| I | 0 | 2 | 1 | 0 | 0 | 0 | 0 |
| like | 2 | 0 | 0 | 1 | 0 | 1 | 0 |
| enjoy | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| deep | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| learning | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| NLP | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| flying | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice Terme-Terme

- Example corpus:

**- He is not lazy. He is intelligent. He is smart.**

*Fenêtre 2-2 / Context Window of length 2*

| | He | is | not | lazy | intelligent | smart |
|---|---|---|---|---|---|---|
| He | 0 | 4 | 2 | 1 | 2 | 1 |
| is | 4 | 0 | 1 | 2 | 2 | 1 |
| not | 2 | 1 | 0 | 1 | 0 | 0 |
| lazy | 1 | 2 | 1 | 0 | 0 | 0 |
| intelligent | 2 | 2 | 0 | 0 | 0 | 0 |
| smart | 1 | 1 | 0 | 0 | 0 | 0 |

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice Terme-Terme

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| He | is | not | lazy | He | is | intelligent | He | is | smart |
| He | is | not | lazy | He | is | intelligent | He | is | smart |
| He | is | not | lazy | He | is | intelligent | He | is | smart |
| He | is | not | lazy | He | is | intelligent | He | is | smart |

| | He | is | not | lazy | intelligent | smart |
|---|---|---|---|---|---|---|
| He | 0 | 4 | 2 | 1 | 2 | 1 |
| is | 4 | 0 | 1 | 2 | 2 | 1 |
| not | 2 | 1 | 0 | 1 | 0 | 0 |
| lazy | 1 | 2 | 1 | 0 | 0 | 0 |
| intelligent | 2 | 2 | 0 | 0 | 0 | 0 |
| smart | 1 | 1 | 0 | 0 | 0 | 0 |

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Matrice Terme-Terme

- On peut calculer la similarité entre deux mots:

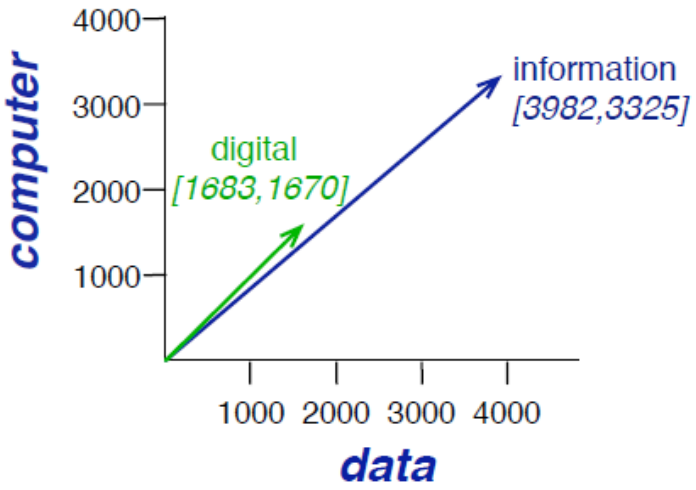


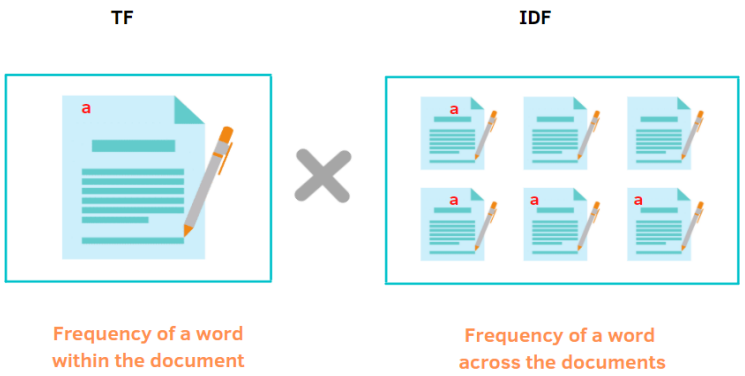

**Figure 6.7** A spatial visualization of word vectors for *digital* and *information*, showing just two of the dimensions, corresponding to the words *data* and *computer*.

# Représentation vectorielle

**Traditional Techniques**
Frequency-based or Statistical based vectorization approach

## TF-IDF

- Term Frequency – Inverse Document Frequency
- TF-IDF donne une mesure qui tient compte de l'importance d'un mot en fonction de sa fréquence d'apparition dans un document et un corpus.
- Distinguer entre les mots qui apparaissent fréquemment et les mots rares. TF-idf donne plus de poids aux mots rares et moins de poids aux mots fréquents.
- TF-idf pénalise les mots fréquents qui apparaissent fréquemment dans un document/corpus comme "le", "est", mais attribue un poids plus important aux mots moins fréquents ou rares.



# Représentation vectorielle

**Traditional Techniques**
Frequency-based or Statistical based vectorization approach

## TF-IDF

$$TFIDF_{t,d,D} = TF_{t,d} \times IDF_{t,D}$$

| $TFIDF_{t,d,D}$ | $TF_{t,d}$ | $IDF_{t,D}$ |
|---|---|---|
| Importance d'un terme t dans un document d | Fréquence d'un terme t dans un document d | Importance du terme t dans l'ensemble des documents D |

**TF(t, d)** = nombre de fois que le terme **t** apparait dans le document **d** / nombre de termes dans le document **d**

**IDF(t, D)** = log (Nombre total de documents / Nombre de documents contenant le terme t)

# Représentation vectorielle

**Traditional Techniques**
Frequency-based or Statistical based vectorization approach

## TF-IDF

$$TFIDF_{t,d,D} = TF_{t,d} \times IDF_{t,D}$$

- $TFIDF_{t,d,D}$ : Importance d'un terme t dans un document d
- $TF_{t,d}$ : Fréquence d'un terme t dans un document d
- $IDF_{t,D}$ : Importance du terme t dans l'ensemble des documents D

TF-IDF weighted term-document matrix:


$$W_{x,y} = tf_{x,y^*} \log\left(\frac{N}{df_x}\right)$$

$W_{x,y}$= Word x within document y

$tf_{x,y}$= frequency of x in y

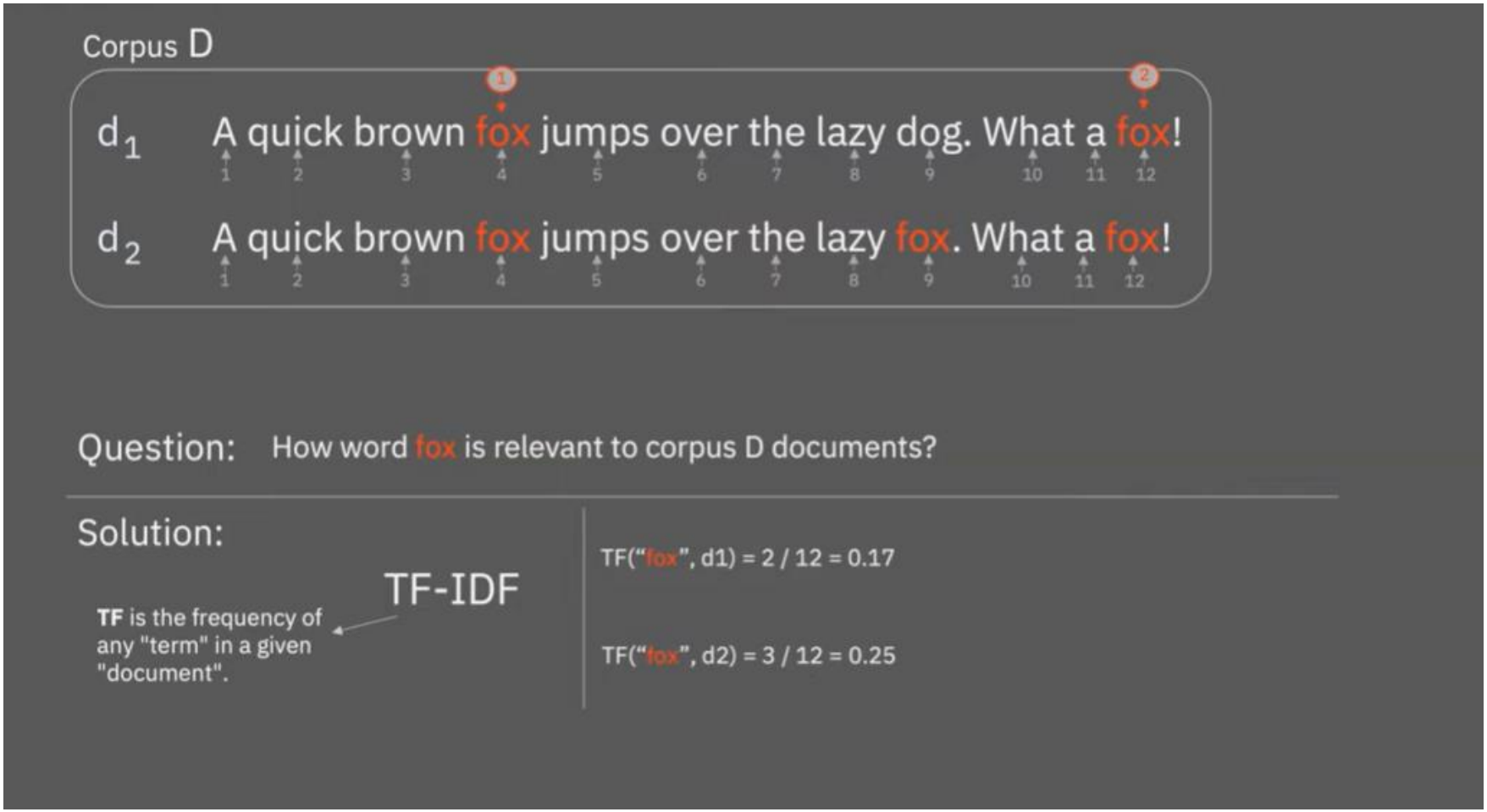
$df_x$= number of documents containing x

N=total number of documents

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## TF-IDF

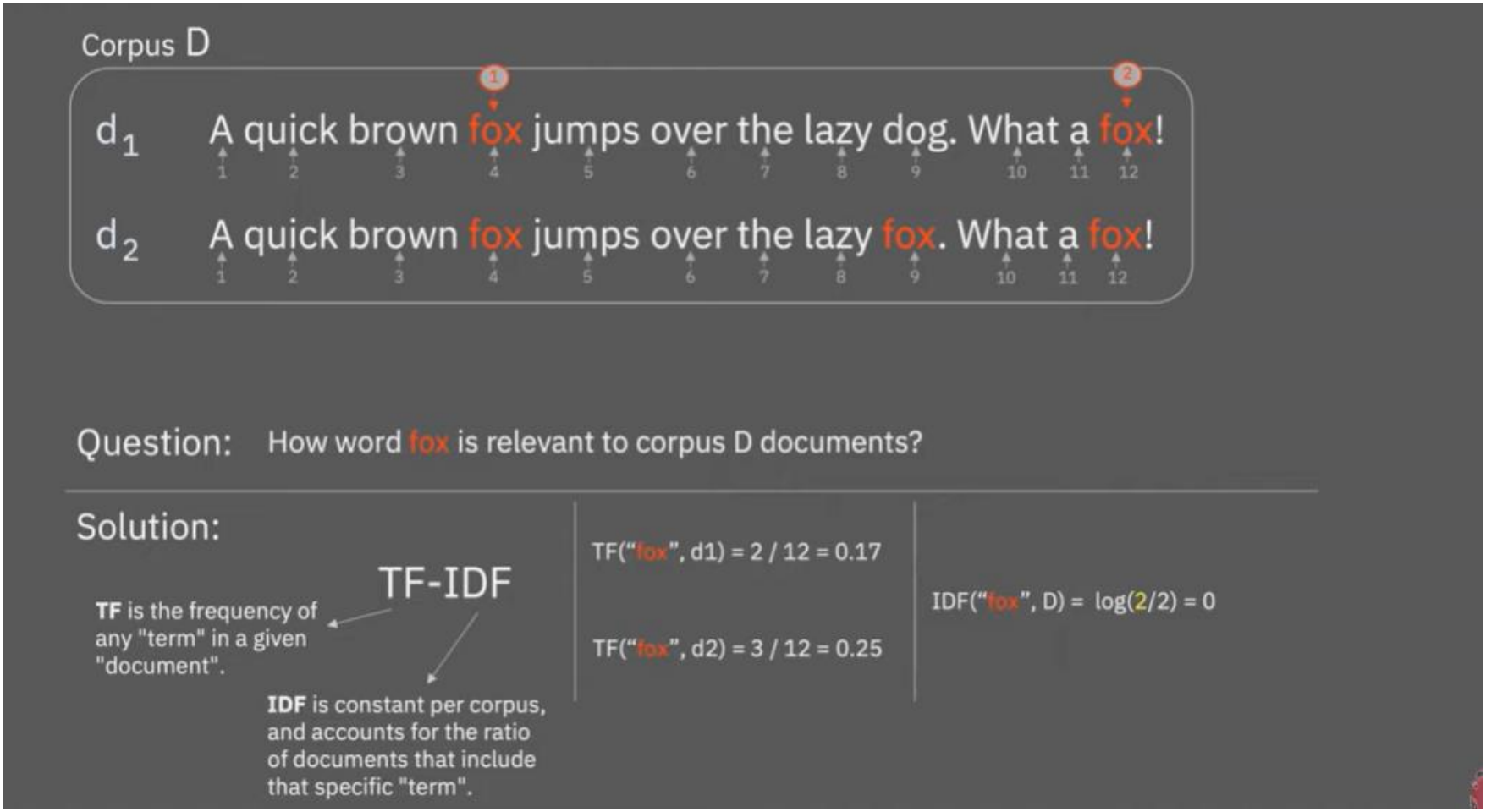



https://www.youtube.com/watch?v=vZAXpvHhQow

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

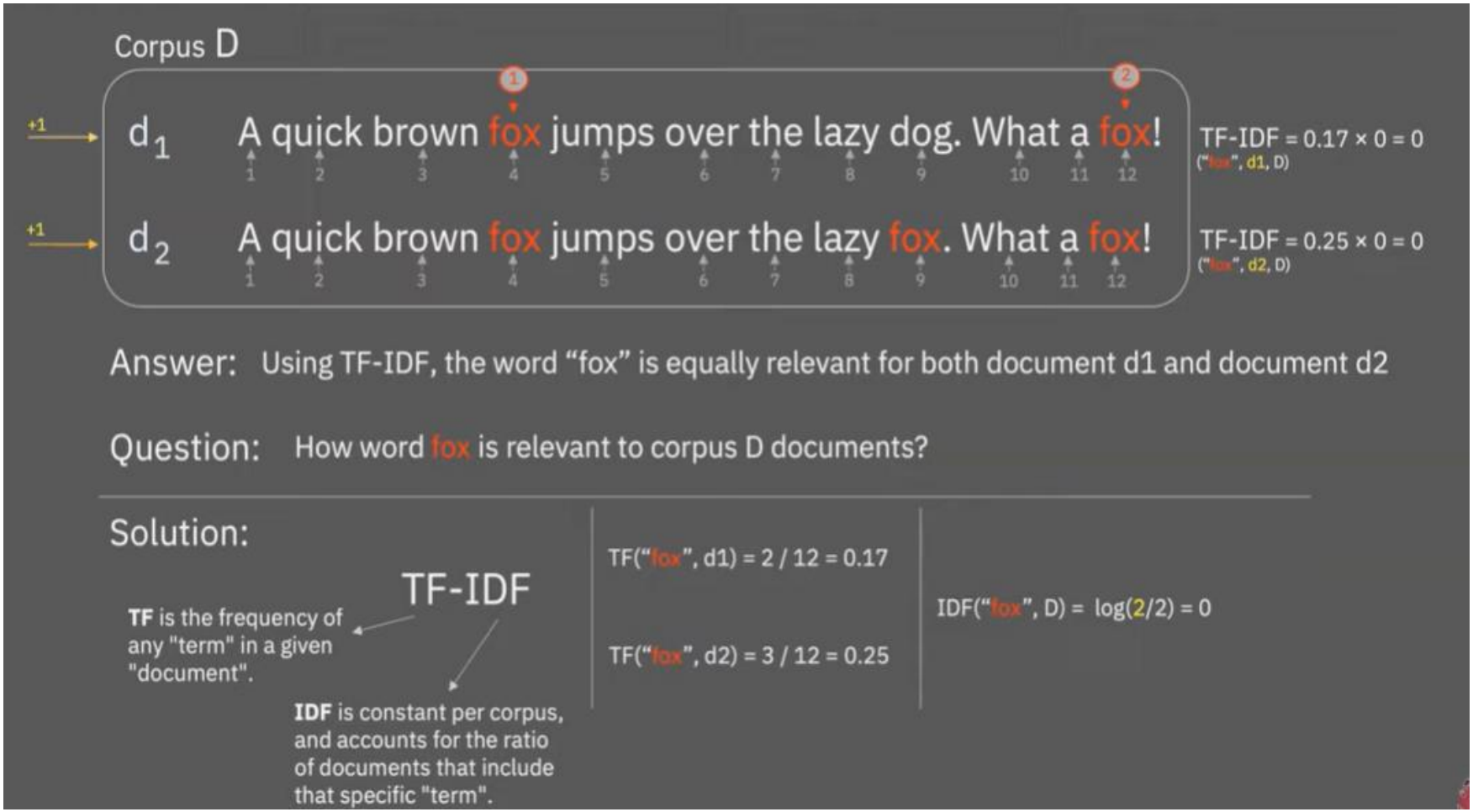
## TF-IDF



# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## TF-IDF



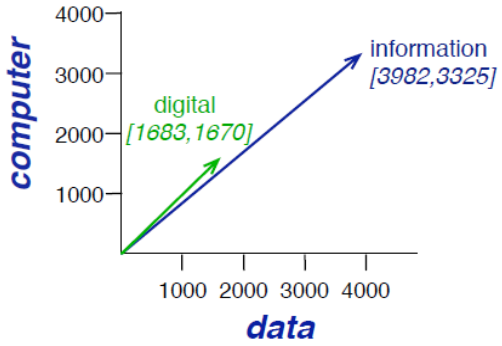
# Représentation vectorielle

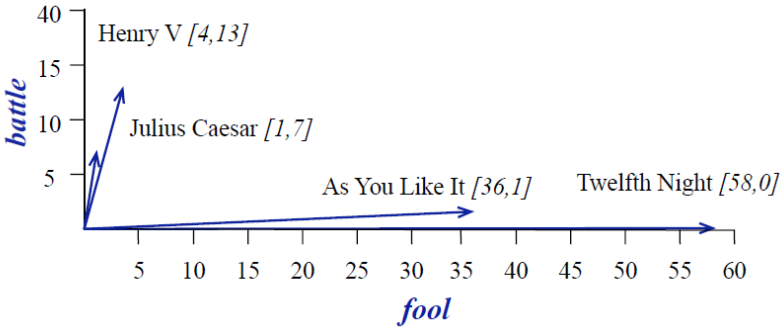
**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Mesure de Similarité : Cosinus

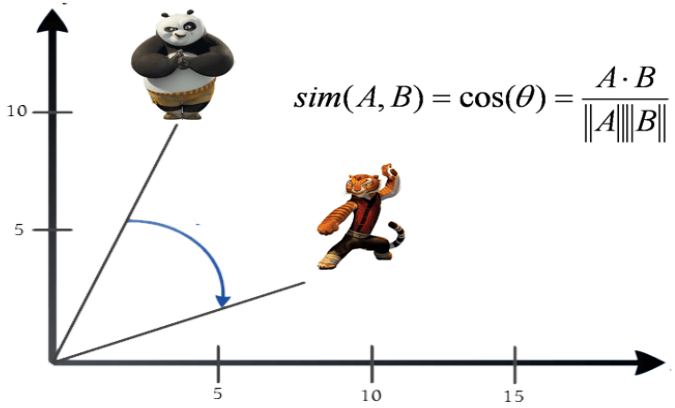
L'utilisation principale de la vectorisation du texte est de déterminer la similarité.

Comment mesurer la similarité entre deux mots / deux documents ? => Similarité entre leurs vecteurs.




**Cosine Similarity**




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

# Représentation vectorielle

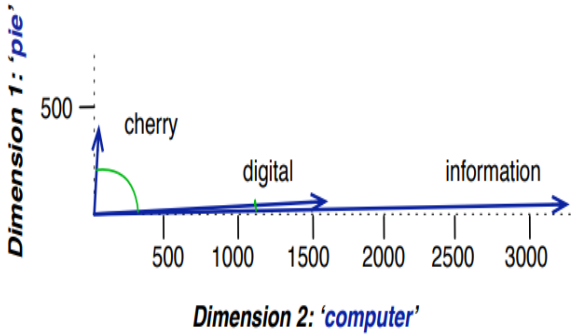
**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Mesure de Similarité : Cosinus

- Exemple: la similarité cosinus entre deux mots

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

| | pie | data | computer |
|---|---|---|---|
| cherry | 442 | 8 | 2 |
| digital | 5 | 1683 | 1670 |
| information | 5 | 3982 | 3325 |




$$\cos(\text{cherry}, \text{information}) = \frac{442*5+8*3982+2*3325}{\sqrt{442^2+8^2+2^2}\sqrt{5^2+3982^2+3325^2}} = .017$$

$$\cos(\text{digital}, \text{information}) = \frac{5*5+1683*3982+1670*3325}{\sqrt{5^2+1683^2+1670^2}\sqrt{5^2+3982^2+3325^2}} = .996$$

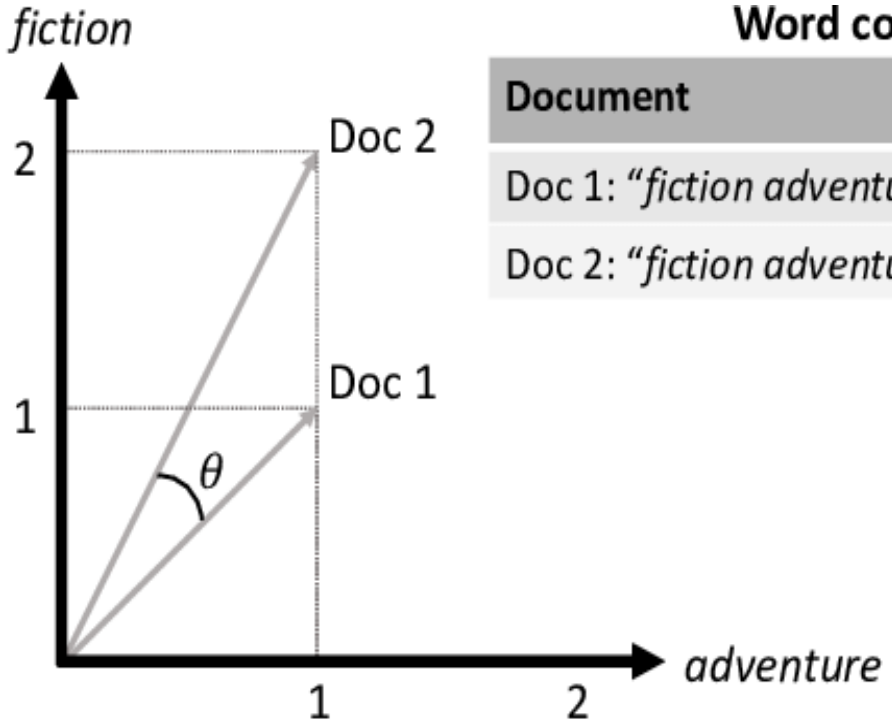
# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Mesure de Similarité : Cosinus

- Exemple: la similarité cosinus entre deux documents

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




**Word count per document**

| Document | *fiction* | *adventure* |
|---|---|---|
| Doc 1: "*fiction adventure*" | 1 | 1 |
| Doc 2: "*fiction adventure fiction*" | 2 | 1 |

$$\cos\theta = \frac{\overrightarrow{Doc\ 1} \cdot \overrightarrow{Doc\ 2}}{\|\overrightarrow{Doc\ 1}\|\|\overrightarrow{Doc\ 2}\|}$$

$$= \frac{(1 * 2) + (1 * 1)}{\sqrt{1^2 + 1^2}\sqrt{2^2 + 1^2}}$$

$$= 0.95$$

# Représentation vectorielle

La **vectorisation du texte** est le processus de conversion de texte en vecteurs numériques. Il peut y avoir différentes représentations numériques vectorielles du même texte.

- Types:

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

Ex : One-Hot, N-grams, BoW, TF-IDF, PMI, Count Vectorizer, co-occurrence matrix, etc.

**New Age Techniques**
**Prediction / Neural Network based vectorization approach**

Ex : Word2Vec, CBoW, Skip Gram, Glove, FastText, ELMo, BERT, XLNet, etc.


Fig. Evolution of NLP Models

# Représentation vectorielle

La **vectorisation du texte** est le processus de conversion de texte en vecteurs numériques. Il peut y avoir différentes représentations numériques vectorielles du même texte.

- Types:

**Traditional Techniques**
Frequency-based or Statistical based vectorization approach

→ **Inconvénients et problématiques**

Ex : One-Hot, N-grams, BoW, TF-IDF, PMI, Count Vectorizer, co-occurrence matrix, etc.


Fig. Evolution of NLP Models

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Inconvénients et problématiques

- One-Hot ne capture pas les relations entre les différents mots. Par conséquent, il ne transmet pas d'informations sur le contexte.
- La **taille** du vecteur est égale au nombre de mots uniques dans le vocabulaire.
- BOW ne préserve pas l'ordre des mots. Il ne permet pas de tirer des conclusions utiles pour les tâches NLP en aval.
- N-Gram a trop de caractéristiques. Ce qui cause le problème de **sparsity** et ça coûte cher en calculs. Aussi, choisir la valeur optimale de N n'est pas facile.
- Pour stocker la matrice de cooccurrence, nous avons besoin d'une énorme quantité de mémoire. Problème de sparsity (composé de 0).

Sparse Matrix

$$\begin{bmatrix} 1.1 & 0 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 1.9 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 2.6 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 7.8 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 2.7 & 0 & 0 \\ 1.6 & 0 & 0 & 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9 & 1.7 \end{bmatrix}$$

# Représentation vectorielle

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

## Inconvénients et problématiques

Pour stocker la matrice de cooccurrence, nous avons besoin d'une énorme quantité de mémoire. Problème de sparsity. Very high dimensional.

### Dense Matrix

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 31 | 2 | 9 | 7 | 34 | 22 | 11 | 5 |
| 11 | 92 | 4 | 3 | 2 | 2 | 3 | 3 | 2 | 1 |
| 3 | 9 | 13 | 8 | 21 | 17 | 4 | 2 | 1 | 4 |
| 8 | 32 | 1 | 2 | 34 | 18 | 7 | 78 | 10 | 7 |
| 9 | 22 | 3 | 9 | 8 | 71 | 12 | 22 | 17 | 3 |
| 13 | 21 | 21 | 9 | 2 | 47 | 1 | 81 | 21 | 9 |
| 21 | 12 | 53 | 12 | 91 | 24 | 81 | 8 | 91 | 2 |
| 61 | 8 | 33 | 82 | 19 | 87 | 16 | 3 | 1 | 55 |
| 54 | 4 | 78 | 24 | 18 | 11 | 4 | 2 | 99 | 5 |
| 13 | 22 | 32 | 42 | 9 | 15 | 9 | 22 | 1 | 21 |

### Sparse Matrix

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | . | 3 | . | 9 | . | 3 | . | . | . |
| 11 | . | 4 | . | . | . | . | . | 2 | 1 |
| . | . | 1 | . | . | . | 4 | . | 1 | . |
| 8 | . | . | . | 3 | 1 | . | . | . | . |
| . | . | . | 9 | . | . | 1 | . | 17 | . |
| 13 | 21 | . | 9 | 2 | 47 | 1 | 81 | 21 | 9 |
| . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | 19 | 8 | 16 | . | . | 55 |
| 54 | 4 | . | . | . | 11 | . | . | . | . |
| . | . | 2 | . | . | . | . | 22 | . | 21 |

# Représentation vectorielle

La **vectorisation du texte** est le processus de conversion de texte en vecteurs numériques. Il peut y avoir différentes représentations numériques vectorielles du même texte.

- Types:

**Traditional Techniques**
**Frequency-based or Statistical based vectorization approach**

Ex : One-Hot, N-grams, BoW, TF-IDF, PMI, Count Vectorizer, co-occurrence matrix, etc.

**New Age Techniques**
**Prediction / Neural Network based vectorization approach**

Ex : Word2Vec, CBoW, Skip Gram, Glove, FastText, ELMo, BERT, XLNet, etc.


Fig. Evolution of NLP Models

# Références

Speech and Language Processing - Livre de Dan Jurafsk - https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

Cours - *François Yvon* – Une petite introduction au Traitement Automatique des Langues Naturelles, https://perso.limsi.fr/anne/coursM2R/intro.pdf

Article - Pascale Sébillot - Le traitement automatique des langues face aux données textuelles volumineuses et potentiellement dégradées : qu'est-ce que cela change ?
- https://hal.archives-ouvertes.fr/hal-01056396/document

Cours - ARIES Abdelkrime - Le traitement automatique du langage naturel. https://github.com/projeduc/ESI_2CS_TALN

Articles - Step by Step Guide to Master NLP, by CHIRAG GOYAL - https://www.analyticsvidhya.com/blog/