Algorithmes d’analyse de données

Sommaire

- Traitement automatique des langues (TAL – NLP)
- Analyse des sentiments (SA)

TAL : Introduction

- **Définition** : L'objectif du TAL est d'octroyer à un système informatique la capacité de comprendre et d'interagir avec le langage naturel des humains en tenant compte du contexte.

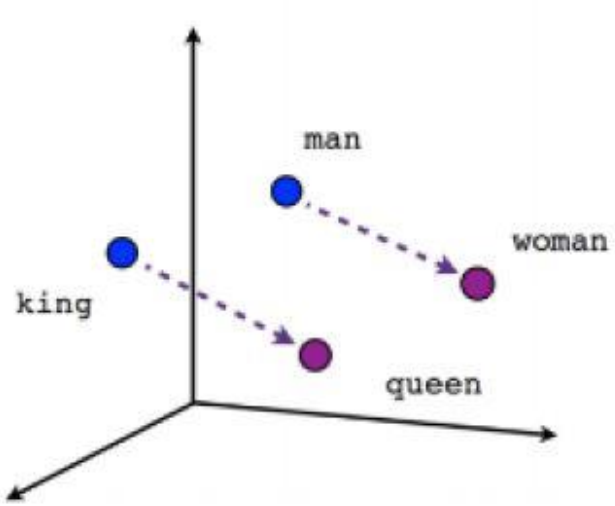
- **Multimodal** :
  - Écrit (texte)
  - Parlé (audio)

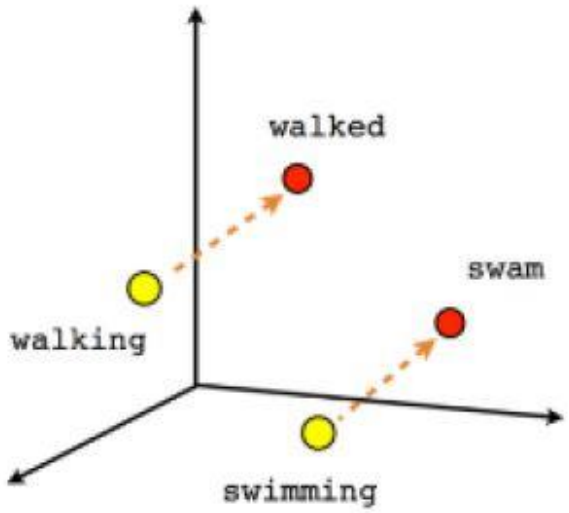
- **Interdisciplinaire** :
  - Science des données
  - Linguistique

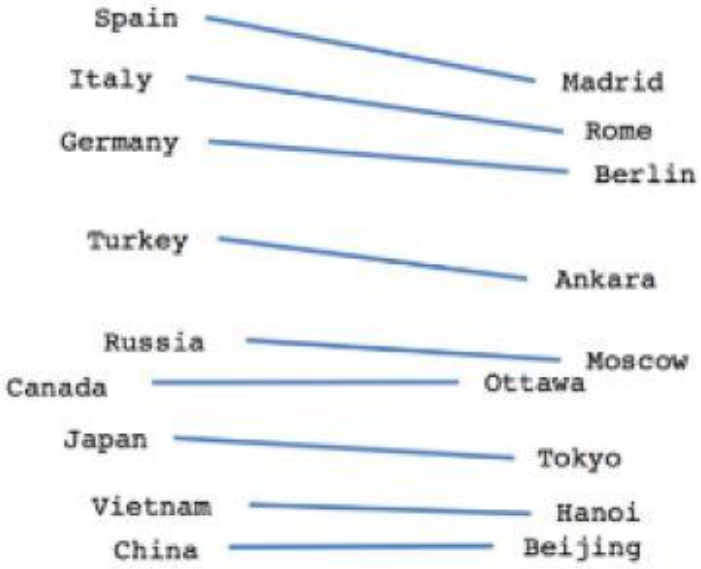
TAL : Applications

- **Multilingues** :
  - Traduction d'une langue à une autre,
  - Reconnaissance de langue.
- **Transcription :**
  - Reconnaissance vocale (audio ➔ texte),
  - Synthèse vocale (texte ➔ audio).
- **Analyse :**
  - analyse syntaxique : découverte d'éléments syntaxiques (sujet, verbe, objet, etc.),
  - reconnaissance d'entités : pays, prénom/personnage, etc.
  - étiquetage sémantique : thème sous-jacent.
- **Génération :**
  - IA générative (ChatGPT, BERT, etc.)
    - Auto-complétion
    - Résumé
    - Q&A

TAL : Applications - examples







TAL : comment passer du texte à une quantité ?

- **Pourquoi ?** Quantifier le texte pour appliquer des algorithmes

- **Comment ?** Plongement sémantique (embeddings)
  - Transformation (projection) des données textuelles (non quantitatives) en un vecteur de nombres continus (espace numérique) ➔ représentation vectorielle de mots
  - Exemples : word2vec, bag-of-words, etc.
  - Deux méthodes :
    - Réseau de neurones :
      - Les descriptions du texte sont extraites automatiquement par un réseau de neurones
    - Fréquence :
      - Les descriptions du texte sont calculés en comptant la fréquence (normalisée ou pas) d'apparition des mots

TAL : bag-of-words VS. word2vec

bag-of-words

- Plongement éparse
- mot=[0 0 0 0 0 0 0 0 1 0 0 0 00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
- Très grande dimensionnalité
- Ils ne capturent pas les similitudes sémantiques

word2vec

- Plongement denses
- mot=[0.06 -0.01 0.13 0.07 -0.06-0.04 -0.5]
- Faible dimensionnalité
- Capture des similarités sémantiques et syntaxiques

TAL : fréquence des termes (TF)

- **Définition** : TF quantifie la fréquence d'apparition (l'importance) d'un mot dans un document. TF calcule le ratio du nombre d'occurrences du mot $m \in M$ dans un corpus C :

$$TF(m, C) = \frac{\text{Nombre d'occurrences du mot } m \text{ dans le corpus C}}{\text{Nombre total de mots dans le corpus C}}$$

Exemple d'un corpus :

1. « Chouette le M2 IDEAL ! En tant qu'étudiant on fait que apprendre des algorithmes et on fait apprendre à des algorithmes des données d'étudiants (cf. dataset1). On peut dire que c'est le plus idéal des M2. »

Exercice : calculer $TF(apprendre, 1)$

TAL : fréquence inverse du document (IDF)

- **Définition** : IDF quantifie la fréquence d'apparition (propagation/présence) du mot *m* ϵ *M* dans l'ensemble des corpus. IDF calcule le ratio du nombre total de corpus sur le nombre de corpus contenant le mot *m* :

$$IDF(m) = \log\left(\frac{\text{Nombre total de corpus}}{\text{Nombre de corpus contenant le mot } m}\right)$$

Exemple d'un ensemble de corpus :

1. « Chouette le M2 IDEAL ! En tant qu'étudiant on fait que apprendre des algorithmes. On peut dire que c'est le plus idéal des M2. »
2. « Chouette le M2 IDEAL ! En tant qu'étudiant on fait apprendre à des algorithmes des données d'étudiants (cf. dataset1). On peut dire que c'est le plus idéal des M2. »

Exercice : calculer *IDF*(IDEAL) et *IDF*(données)

TAL : Fréquence des termes - fréquence inverse du document (TF-IDF)

- **Définition** : TF-IDF est l'association de TF et IDF dans une seule mesure. TF-IDF identifie les mots qui sont plus fréquents dans un corpus spécifique et moins fréquents dans l'ensemble des corpus :

$$\mathit{TF\text{-}IDF}(m, C) = \mathrm{TF}(m, C) \times \mathrm{IDF}(m)$$

Plus TF-IDF est élevé plus un mot est important dans un corpus spécifique mais relativement rare dans l'ensemble des corpus.

Exemple d'un ensemble de corpus :

1. « Chouette le M2 IDEAL ! En tant qu'étudiant on fait que apprendre des algorithmes. On peut dire que c'est le plus idéal des M2. »
2. « Chouette le M2 IDEAL ! En tant qu'étudiant on fait apprendre à des algorithmes des données d'étudiants (cf. dataset1). On peut dire que c'est le plus idéal des M2. »

Exercice : calculer *TF-IDF*(apprendre, 1), *TF-IDF*(apprendre, 2), *TF-IDF*(données, 1) et *TF*-IDF(données, 2)

TAL & SA : DELTA TF-IDF

- **Objectif** : Extraire le sentiment/polarité d'un texte (négatif ou positif)

DELTA TF-IDF(*m, C*) = *TF-IDF*(*m, C*) [POS] — *TF-IDF*(*m, C*) [NEG]

- **Augmente l'importance des mots qui sont répartis de manière inégale entre les classes positives et négatives et réduit l'importance des mots répartis de manière égale.**
- **Pour savoir quels mots ont un sentiment positif ou négatif, on calcule pour chaque mot du corpus son score DELTA TF-IDF. Les mots dont le score est > 0 sont positifs, tandis que les mots dont le score est < 0 sont négatifs.**

Méthodologie :

1. Diviser le corpus annoté en sous-ensembles POS et NEG contenant des corpus annotés positifs et négatifs,
2. Pour chaque mot, calculer sa valeur TF-IDF dans le corpus positif et négatif,
    1. Le TF sera le même pour les deux (répétition du mot dans le document) tandis que l'IDF changera en fonction des valeurs de l'IDF en utilisant les corpus positifs et négatifs,
3. La valeur DELTA TF-IDF pour chaque terme est égale à la différence entre son score TF-IDF dans les corpus positif et négatif

TAL & SA : DELTA TF-IDF

Exemple d'un ensemble de corpus :

1. « Chouette le M2 IDEAL ! En tant qu'étudiant on fait que apprendre des algorithmes. On peut dire que c'est le plus idéal des M2. », polarité -1
2. « Chouette le M2 IDEAL ! En tant qu'étudiant on fait apprendre à des algorithmes des données d'étudiants (cf. dataset1). On peut dire que c'est le plus idéal des M2. », polarité +1

Exercice : calculer DELTA *TF-IDF*(apprendre, 1) et
DELTA *TF-IDF*(données, 2)

TAL & SA : DELTA TF-IDF

Autre écriture/formule :

$$V_{w,t} = C_{w,t} \times \log_2\left(\frac{|N|\, P_w}{N_w\, |P|}\right)$$

Où :

- $C_{w,t}$: Nombre de fois où le mot w apparaît dans le texte t
- $|N|$ et $|P|$: Nombre de textes étiquetés négativement et positivement, respectivement.
- N_w , P_w : Nombre de textes contenant le mot w dans l'ensemble des textes étiquetés négativement et positivement respectivement.

TAL & SA : DELTA TF-IDF améliorée

Inconvénient

Version standard			
Inst.	P_w	N_w	$V_{w,t}$
t_1	100	0	18.85
t_2	2	0	13.21
t_3	100	1	9.22

Ce résultat va à l'encontre de notre intuition selon laquelle les poids de t_1 et t_3 sont relativement proches et plus élevés que ceux de t_2

Amélioration

Version adaptée			
Inst.	P_w	N_w	$V_{w,t}$
t_1	100	0	6,90
t_2	2	0	1,57
t_3	100	1	5,32

Ces résultats semblent plus raisonnables lorsque les poids de t_1 et t_3 sont proches et relativement importants et que le poids de t_2 est faible par rapport à t_1 et t_3

Nous utilisons un paramètre de lissage = 0,5 pour éviter les erreurs causées par le cas d'une polarité à 0

$$V_{w,t} = C_{w,t} \times \log_2\left(\frac{|N|\, P_w + \alpha}{|P|\, N_w + \alpha}\right)$$

Rania Othman, **Youcef Abdelsadek**, Kamel Chelghoum, Imed Kacem, and Rim Faiz. **Improving Sentiment Analysis in Twitter Using Sentiment Specific Word Embeddings**, In: *International Conference on Intelligent Data Acquisition and Advanced Computing Systems*. IEEEE, 2019.

TAL & SA : Le vecteur de caractéristique texte

Formule :

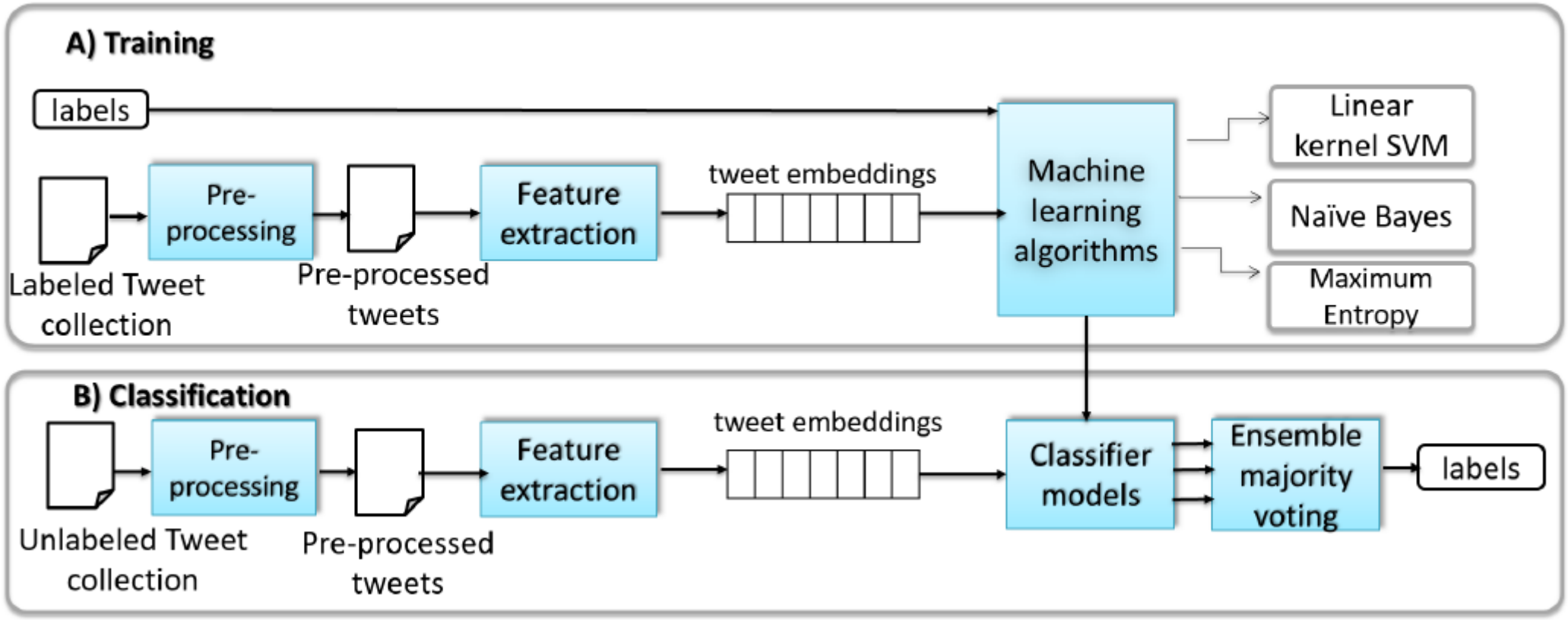
$$V_t = \frac{\sum_{i=1}^{|V|} (v_{w_i} \times wgt(w_i, t))}{\sum_{i=1}^{|V|} wgt(w_i, t)}$$

Où :

- v_{wi} : Le vecteur de plongement du mot w_i généré par word2vec
- $wgt\ (w_i, t)$: Le poids du mot w_i dans un texte donné t généré par le DELTA TF-IDF améliorée
- $|V|$: Le nombre de vecteurs de mots dans t.

Rania Othman, **Youcef Abdelsadek**, Kamel Chelghoum, Imed Kacem, and Rim Faiz. **Improving Sentiment Analysis in Twitter Using Sentiment Specific Word Embeddings**, In: *International Conference on Intelligent Data Acquisition and Advanced Computing Systems*. IEEEE, 2019.

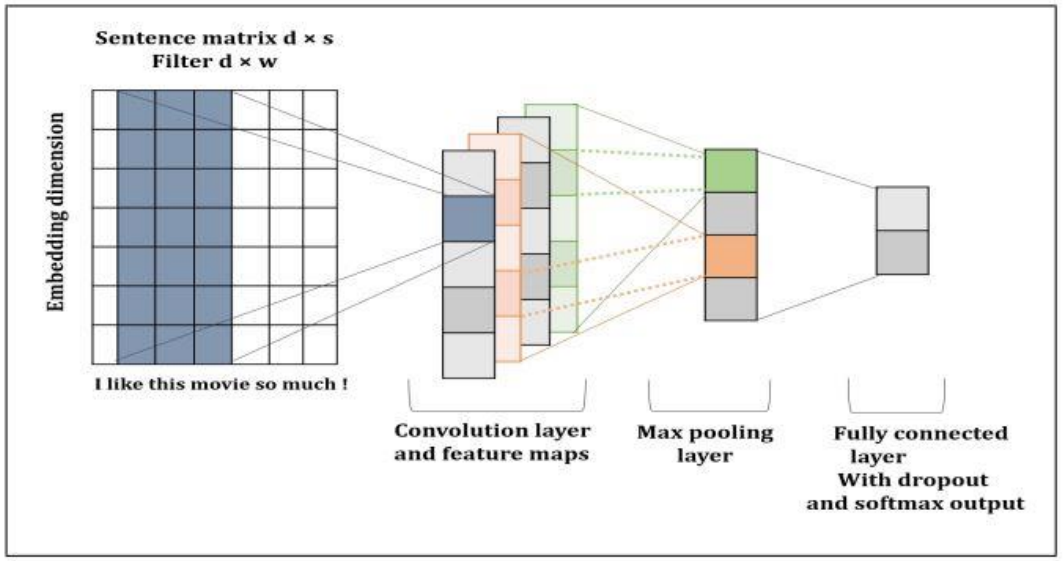
TAL & SA : Apprentissage automatique
Architecture globale – SVM + NB + ME



Rania Othman, **Youcef Abdelsadek**, Kamel Chelghoum, Imed Kacem, and Rim Faiz. **Improving Sentiment Analysis in Twitter Using Sentiment Specific Word Embeddings**, In: *International Conference on Intelligent Data Acquisition and Advanced Computing Systems*. IEEEE, 2019.

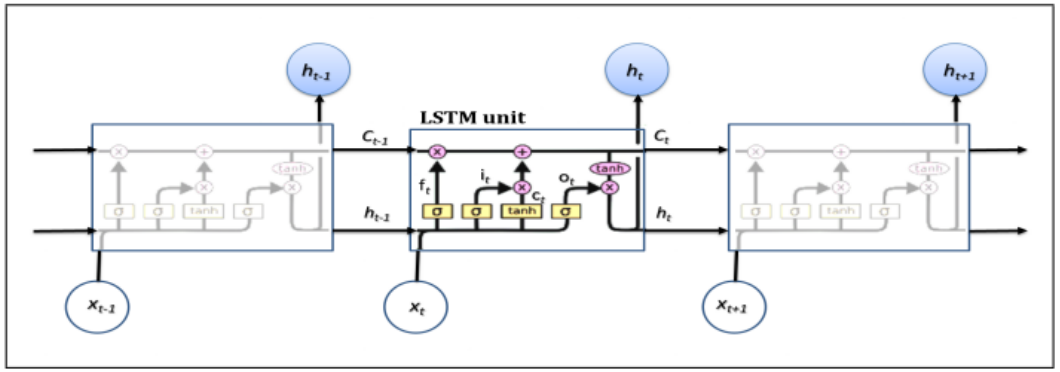
TAL & SA : Apprentissage automatique
Architecture globale du model hybride CNN+LSTM

Convolutional Neural Network (CNN)



- Extrait les caractéristiques locales grâce aux couches convolutives

Long Short-Term Memory (LSTM)



- Identifier les dépendances longue distance entre les séquences de mots

Rania Othman, Rim Faiz, **Youcef Abdelsadek**, Kamel Chelghoum, and Imed Kacem. **Deep Hybrid Neural Networks with Improved Weighted Word Embeddings for Sentiment Analysis**, In: *Advances in Intelligent Data Analysis XIX*, p. 50-62. Springer International Publishing, 2021.

TAL : Pré-traitement

- Étant donné les plongements précédents, y’a-t-il une limitation de l’utilisation des mots ”bruts” ?

- Une phase de nettoyage des données est nécessaire. Cette phase est composée de 3 étapes :

    1. Suppression des mots vides (stop words)
    2. La lemmatisation
    3. La racinisation (Stemming)

- La lemmatisation et racinisation sont des techniques utilisées pour réduire les mots à leur forme de base.
- Ce sont des étapes de prétraitement essentielles pour normaliser les données textuelles avant toute autre analyse.

TAL : Pré-traitement – Suppression des mots vides

- **Remarque** : Par définition TF-IDF les mots fréquents qui se trouvent dans tous les corpus mais la suppression des mots vides permet de converger plus rapidement et accélère le processusde nettoyage de TF-IDF.

- **Définition** : Les mots fonctionnels sont des mots courants qui sont utilisés fréquemment dans une langue mais qui non-significatifs

- **Exemples** :
  - la,
  - le,
  - une,
  - un,
  - et,
  - est,
  - etc.

TAL : Pré-traitement - Lemmatisation

- **Définition** : La lemmatisation consiste à réduire les mots à leur forme canonique, appelée lemme. Elle nécessite des connaissances lexicales pour identifier correctement le lemme.

- **Exemples** :
  - Pour un verbe: sa forme à l'infinitif
  - Pour un nom, adjectif, article, ... : sa forme au masculin singulier

  - (*cheval* ≡ *chevaux*) ≠ *chevalerie* ≠ *chevauche*

TAL : Pré-traitement - Racinisation

- **Définition** : La racinisation est le processus de suppression des suffixes et/ou des préfixes des mots pour **obtenir la forme de base** d'un mot, appelée le **stem** via des heuristiques.

- **Exemples** :
  - Cheval ➔ « cheva »
  - Chevaux ➔ « cheva »
  - Chevalier ➔ « cheva »
  - Chevalerie ➔ « cheva »
  - Chevaucher ➔ « cheva »

- **Inconvénients** :
  - Peut ne pas créer de mot significatif
  - Perte en précision
  - Peut agréger des mots différents : tracteur et trachée ➔ trac