

# Tale Box

Projet Traitement automatique du texte  
en Intelligence Artificielle

---

Ben Fredj Yasmine



2020 - 2021

# Plan

---

- I. Introduction
- II. Les Taches principales
  - Collecte des données
  - Prétraitement des données
  - Modélisation et prédiction du sujet
  - Génération du texte
- III. Analyse des erreurs
- IV. Amélioration Possibles
- V. Démonstration
- VI. Conclusion

# I. Introduction

## Tale Box



Jeu Textuel développer en Python et relevant des concepts NLP (Natural Language Processing). Il peut :

- Discuter avec nous (NLU / NLG)
- Nous aider à raconter une histoire (NLG)
- Deviner le thème de l'histoire (NLU)

Importé les  
données

Pré-  
traitement

Entraîner  
les modèles

## II. Les Taches principales :

---

### Collecte des données

Pour reconnaître le thème d'une histoire comme pour pouvoir en rédiger, nos modèles ont besoin d'un grand nombre de données.

Ces données consistent à 300 contes, histoires et légendes que j'ai collecter personnellement.

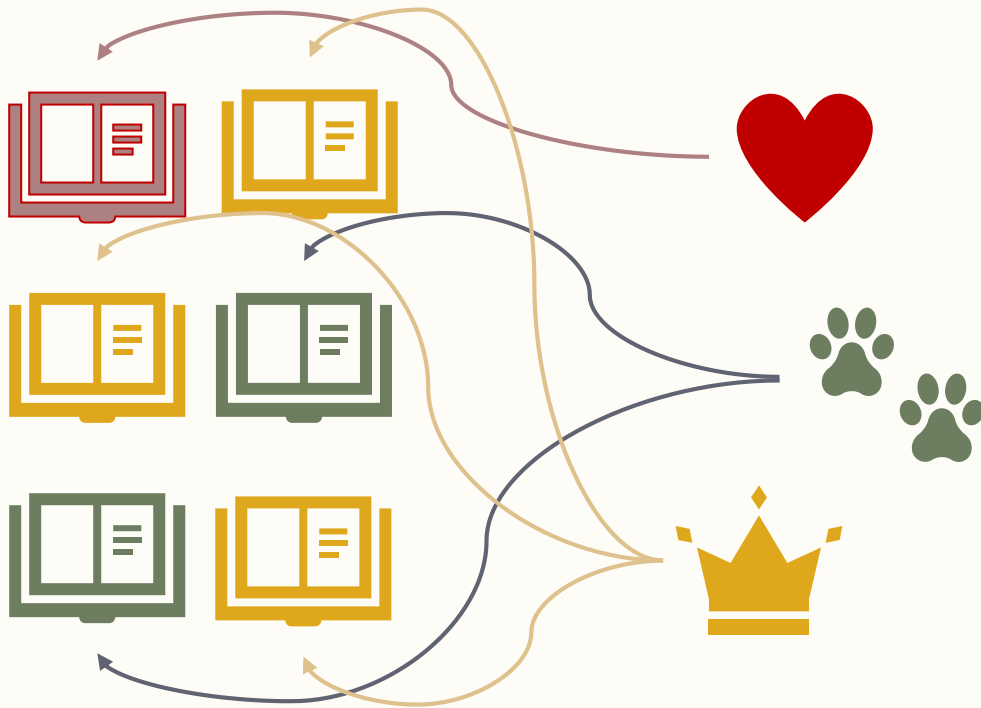
## II. Les Taches principales :

### Prétraitement des données



## II. Les Taches principales :

### Modélisation



#### Allocation de Dirichlet latente (LDA) :

- Un modèle génératif probabiliste qui permet de regrouper des données autour d'un nombre définie de thèmes par le liens de ressemblance.
- Ce modèle nous permet d'avoir K listes de N mots et chaque liste est un thème.
- Ensuite à nous d'essayer d'analyser les mots pour savoir à quelle thème elles correspondent.

## II. Les Taches principales :

---

### Modélisation : Prédiction des thèmes

**Création de sacs des mots (indice, fréquence)**

Avec « CountVectorizer » de « sklearn.feature\_extraction.text »



**Chercher les meilleurs hyperparamètres pour le model LDA**

Avec la méthode « GridSearch » de « sklearn.model\_selection »



**Créer et entrainer le model LDA avec les données prétraités**

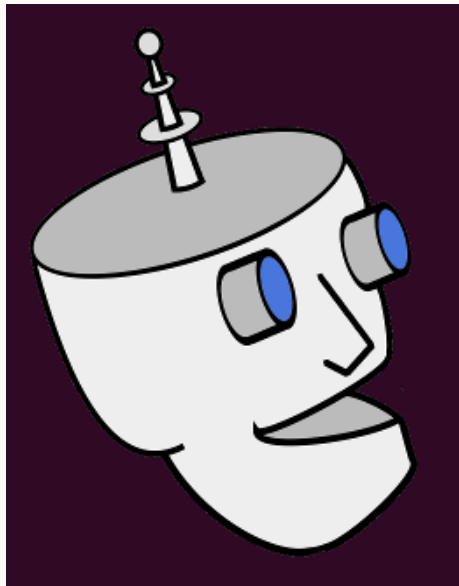
Avec « sklearn.decomposition »

## II. Les Taches principales :

### Générations de texte

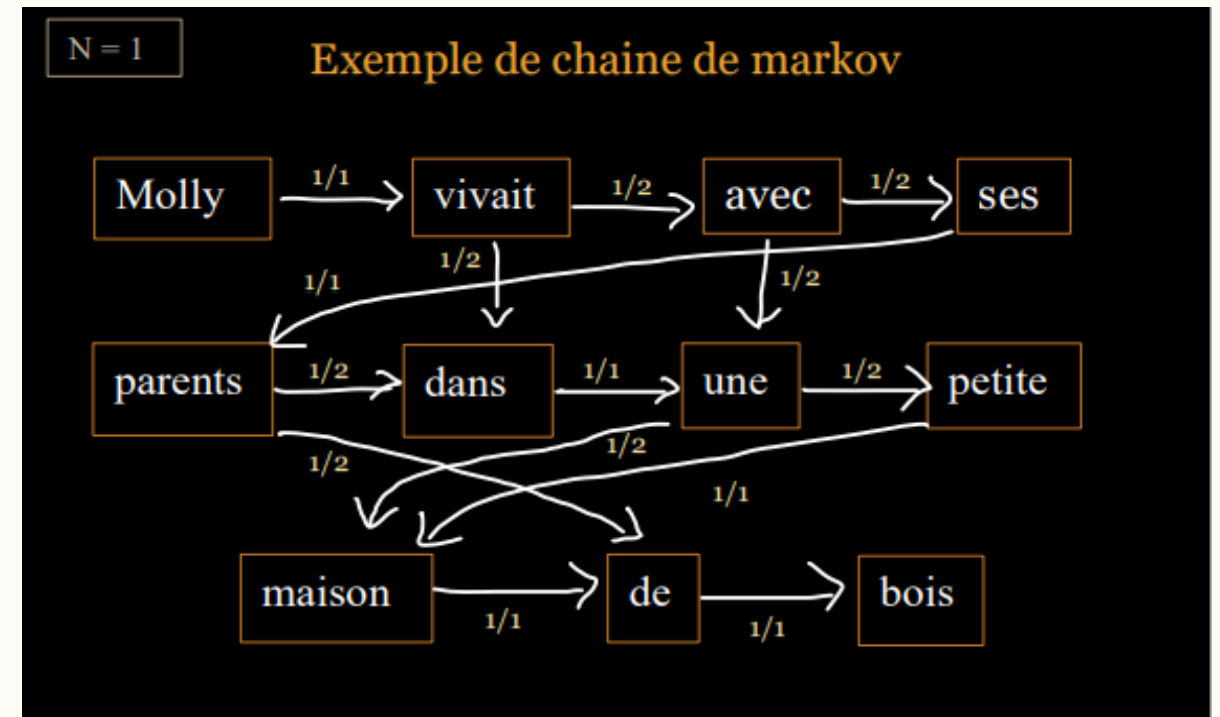
Répondre à des question basique

ChatterBot



Continuer la rédaction d'une histoire

Modèle de markov





# III. Analyse des erreurs

---

- Le model de Markiv peut ne pas reconnaitre un état s'il ne l'a jamais rencontrer dans la phase de l'entraînement .
- Erreurs de génération de texte:
  - Phrases non finis
  - Phrases avec un sens ambiguë
  - Phrases sui ont tendance à s'éloigné du contexte
- Prédiction du sujets peut ne pas être précise
- Le bot peut ne pas répondre correctement dans la discussions s'il n'a pas était entrainer sur plusieurs discussions.

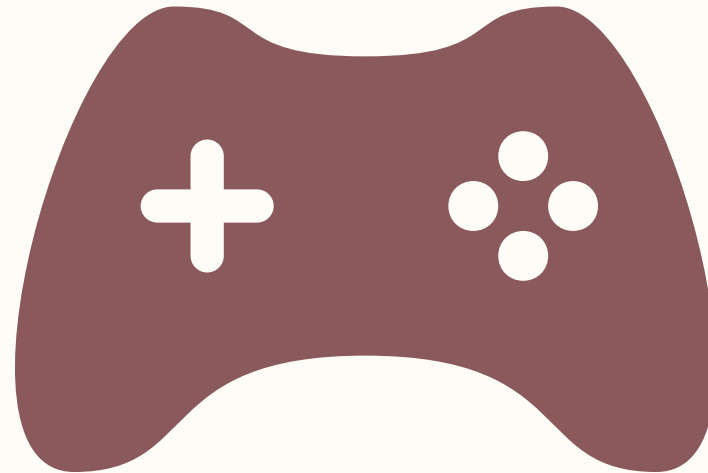
## IV. Améliorations possibles

---

- Avoir un jeu de données plus grand et plus organiser.
- Avoir une génération de texte plus adapter pour avoir des phrases fini et pouvoir garder le contexte de l'histoire tout au long.
- Entraîner notre bot sur plus de discussions.
- Améliorer l'interface graphique.
- Ajouter plus d'option au jeu:
  - Permettre au joueur de faire un choix entre plusieurs suites de texte.
  - Permettre au jet d'analyser l'état d'esprits du joueur avec l'histoire générer.

# V. Démonstration

---



# VI. Conclusion

---

- Objectifs plus ou moins Atteints
- Découvert de nombreuses librairies et concept de traitement automatique de texte
- Découverte de la difficulté de permettre à l'ordinateur d'agir comme un humain



Merci pour votre attention

