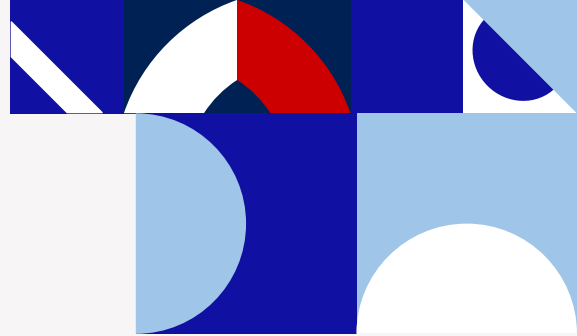


PROPOSITION D'UNE MÉTHODE ORIGINALE DE RÉÉCRITURE CRÉATIVE DE TEXTE “OULIPO”

Présenté par:
MESSILI Islem





Plan de Présentation

01

Introduction

02

Methode Proposer

03

Implementation

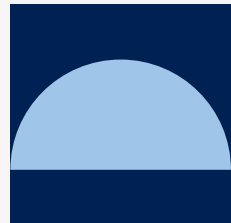
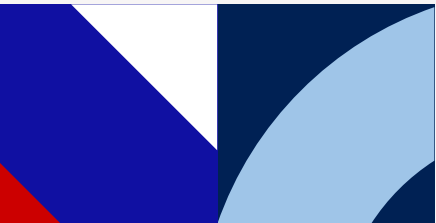
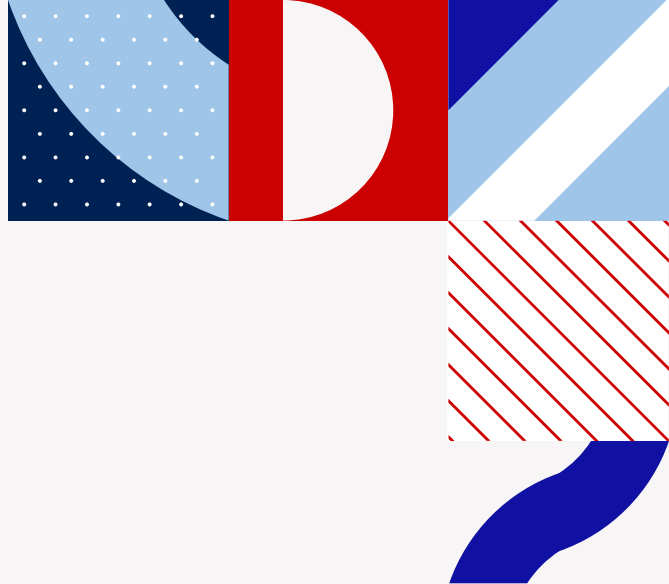
04

Conclusion



01

Introduction



C'est quoi OULIPO

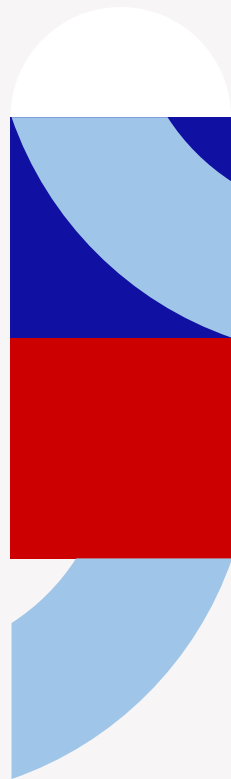
Créé en 1960 par **Raymond Queneau** et **François Le Lionnais**, l'Oulipo, ou Ouvroir de littérature potentielle est un groupe de recherche littéraire. Il a pour but de découvrir de nouvelles potentialités du langage et de moderniser l'expression à travers des **"contraintes"**.





C'est quoi les Contraintes OULIPO

Les **contraintes** Oulipo sont des règles spécifiques imposées à la création littéraire pour stimuler **la créativité**. Elles peuvent être mathématiques, linguistiques ou structurelles, poussant les écrivains à explorer des voies inattendues et à repousser **les limites traditionnelles** de l'écriture.





Quelque Contraintes

Lipogramme : Exclut délibérément une ou plusieurs lettres de l'alphabet dans un texte.

S+7 (Substitution par 7) : Remplace chaque substantif d'un texte par le septième substantif suivant dans un dictionnaire.

Centon : Compose un texte à partir de fragments d'autres textes existants, les mélangeant de manière nouvelle.

Hétérogramme : Exclut une lettre donnée tout en utilisant toutes les autres dans un texte.



The slide features four decorative geometric patterns in the corners. Top-left: A dark blue square with a light blue dotted quarter-circle. Top-right: A collage of shapes including a red vertical bar, a white semi-circle, and blue and white diagonal stripes. Bottom-left: A dark blue square with a light blue quarter-circle. Bottom-right: A dark blue square with a light blue semi-circle. A red and white diagonal striped square is also visible on the right side.

02

Methode Proposer



Methode Proposer

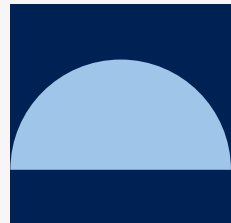
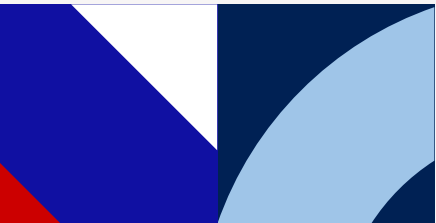
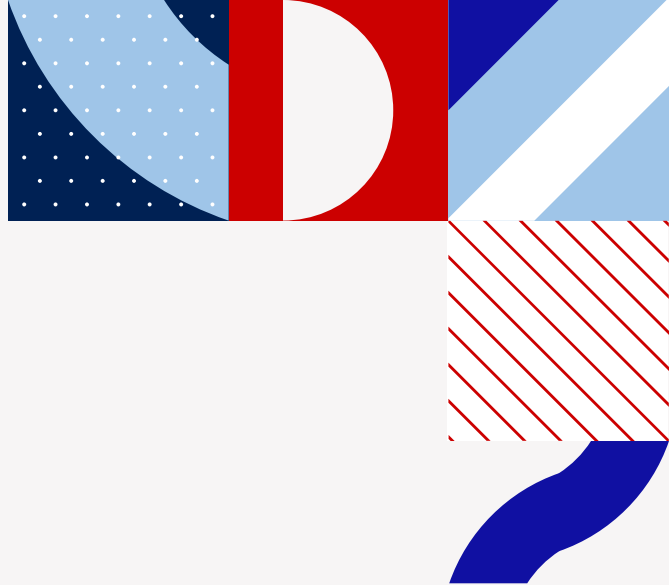
Dans le cadre d'Oulipo, nous proposons une contrainte qui substitue chaque substantif par un autre mot provenant de sa définition, en choisissant le mot le plus proche possible par plongement lexical.

Pour les Adjectifs on les remplace par leur traduction français.



03

Implementation





Outils Utiliser

- Langage de programmation utilisé : **Python**
- Bibliothèques : **NLTK, WordNet, Gensim, Scikit-learn, SpaCy**
- Environnement d'édition de texte : **Jupyter Notebook**

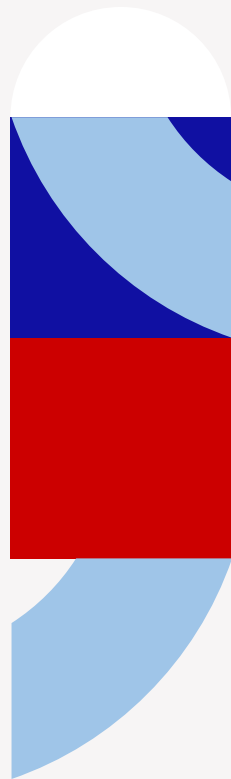


Étapes

Pour les substantifs:

- Évaluer la position du mot.
- Identifier le sens du mot en utilisant l'algorithme Lesk.
- Utiliser les embeddings lexicaux pour trouver le mot le plus proche dans sa définition.
- Substituer le mot par le terme extrait.

Pour les adjectifs:

- Évaluer la position du mot.
 - Substituer le mot par la traduction française.
- 

Code source

```
import spacy
from spacy import displacy
import numpy as np
from gensim.models import KeyedVectors
from sklearn.metrics.pairwise import cosine_similarity
from nltk.corpus import wordnet as wn
import re
from translate import Translator

def lesk(context_sentence, ambiguous_word, pos=None, synsets=None):

    context = set(context_sentence)

    if synsets is None:
        # synsets = wn.synsets(ambiguous_word, lang='fra')
        synsets = wn.synsets(ambiguous_word)

    if pos:
        synsets = [ss for ss in synsets if str(ss.pos()) == pos]

    if not synsets:
        return None

    _, sense = max(
        (len(context.intersection(ss.definition().split())), ss) for ss in synsets
    )

    return sense
```

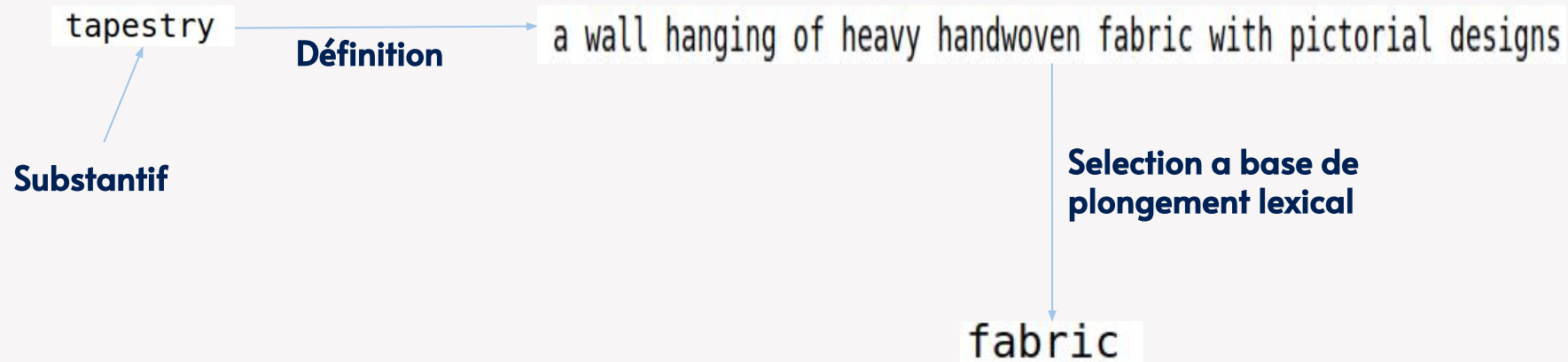
```
for context_sent in doc.sents:
    for token in context_sent:
        if token.pos_ in pos_sub:
            context_sentence = context_sent.text
            ambiguous_word = token.text
            sense = lesk(context_sentence, ambiguous_word, pos = 'n')
            if sense:

                word_def = sense.definition()
                split_def = re.split(r'\s|[\^a-zA-Z0-9]', word_def)
                split_def = list(filter(None, split_def))

                best_similar = 0
                similar_word = ""

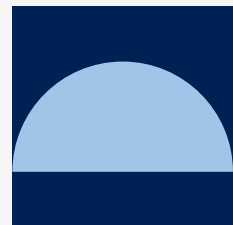
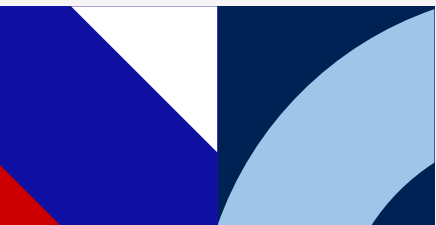
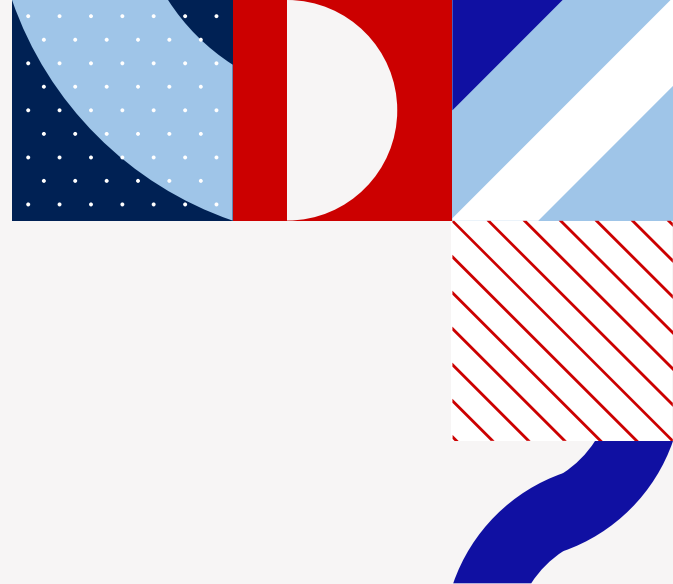
                for word in split_def:
                    if word in ww.key_to_index:
                        if ww.similarity(token.text.lower(), word.lower()) > best_similar:
                            best_similar = ww.similarity(token.text.lower(), word.lower())
                            similar_word = word

                print(f"\033[1;31m{similar_word}\033[0m - \033[1;32m{token.text}\033[0m")
            else:
                print(token.text)
        else:
            if token.pos_ in pos_adj:
                translation = translator.translate(token.text)
                print(f"\033[1;34m{translation}\033[0m - \033[1;32m{token.text}\033[0m")
            else:
                print(token.text)
```



04

Conclusion





Conclusion

Ce projet nous a ouvert les portes pour explorer le monde de l'Oulipo, nous permettant d'apprendre sur les contraintes et leur fonctionnement.





**Merci Pour Votre
Attention!**

