AVIGNON UNIVERSITÉ

Rapport de TP : Développement d'une application de traitement de texte avec Scrum

Hanan BOUD
Asma BOUKRA
Farouk GNANKAMBARY
Walid FARAH
M. Steve B. SANOGO

MASTER INFORMATIQUE

Master 1 en Intelligence Artificielle

Concevoir et Organiser des Solutions Logicielles Complexes

Processus de Développement Logicielle

UFR SCIENCES TECHNOLOGIES SANTÉ

AVIGNON UNIVERSITÉ
CERI

CENTRE D'ENSEIGNEMENT ET DE RECHERCHE EN INFORMATIQUE
ceri.univ-avignon.fr

Sommaire

1. Contexte

Dans le cadre de l'UE *Processus du Développement Logiciel*, nous avons réalisé une série de travaux pratiques organisés autour du développement d'un projet logiciel « dummy », conformément aux consignes pédagogiques de l'enseignement.

L'objectif principal de ces travaux n'était pas la complexité technique du produit développé, mais la mise en application concrète des concepts, méthodes et outils de conduite de projet logiciel, et en particulier du framework Scrum.

Chaque séance de TP correspondait à un sprint, comprenant une revue des livrables du sprint précédent, une démonstration, ainsi qu'une définition des objectifs du sprint suivant.

Dans ce cadre :

- Le **Product Owner (PO)**, notre enseignant **Ludovic Bonnefoy**, définit les besoins et priorités.
- L'équipe de développement (nous) est responsable de la mise en œuvre des fonctionnalités.
- Chaque sprint inclut une **revue avec le PO** pour présenter les résultats du sprint précédent et ajuster le backlog.

2. Présentation du projet

Le projet retenu consiste en le développement d'une **bibliothèque d'analyse** et de **vectorisation de texte**, accessible **via une interface web simple**. Cette application permet de réaliser différentes opérations de traitement automatique du langage naturel, telles que :

- **le calcul de similarité entre textes**,
- **le résumé automatique**,
- **l'extraction de snippets pertinents,**
- **la ponctuation un texte à minima (**avec le point ou la virgule**),**
- la **vectorisation TF-IDF** (l'IDF étant basé sur des corpus de Wikipédia)

Ce projet a été volontairement conçu avec une **complexité technique modérée** afin de se concentrer sur l'**organisation du travail**, **la gestion des sprints**, **la collaboration en équipe** et l'**appropriation des pratiques agiles**. Les fonctionnalités ont été définies et ajustées progressivement en collaboration avec le Product Owner, dans une logique itérative conforme à Scrum.

3. Synthèse du Product Backlog

ID	User Story	Définition de Done (DoD)	Description / Fonctionnalité	Sprint	Statut
US1	En tant qu'utilisateur, je veux une page d'accueil web claire avec les 4 modules	Page responsive 4 boutons fonctionnels Navigation fluide	Interface pour saisir le texte, afficher top mots, accéder aux modules	1-4	Fait
US2	En tant qu'utilisateur, je veux afficher les vecteurs de mots	Upload texte Calcul fréquence Top 10 mots affichés	Visualisation statistique des mots importants	1	Fait
US3	En tant qu'utilisateur, je veux calculer TF et IDF	Saisie texte Calcul TF/IDF Affichage tableau	Module de calcul TF-IDF	1	Fait
US4	En tant qu'utilisateur, je veux comparer deux textes pour détecter une similitude	2 zones texte Score 0-1 Visualisation	Comparaison et score de similarité	2-3	Fait
US5	En tant qu'utilisateur, je veux coller un texte et obtenir un résumé/ plagiat	Choix % réduction Résumé cohérent Mots-clés conservés	Module de résumé automatique	2-3	Fait
US6	En tant qu'utilisateur, je veux un moteur de recherche de snippets	Upload multiples fichiers Recherche texte Résultats avec contexte	Recherche d'extraits précis dans plusieurs fichiers	4	Fait
US7	En tant qu'utilisateur, je veux faire la ponctuation de mon texte	Texte saisi ou collé, ponctuation automatique appliquée (points,virgules, majuscules). Résultat affiché dans l'interface web.	Module corrige et ajoute la ponctuation pour améliorer la lisibilité du texte.	5-6	Fait
US8	En tant qu'utilisateur, je veux utiliser un LLM pour reformuler un texte	Intégration API Paramètres réglables Qualité vérifiable	Module de reformulation avec un petit modèle de langage	5	Pas commencé

4. Organisation Scrum

4.1 Rôles et cérémonies

Rôles

- **Product Owner** : Ludovic Bonnefoy, fournit les besoins et priorités.
- **Scrum Master** : Hanan Boud, facilite l'organisation et le suivi des sprints.
- **Équipe de développement** : Farouk, Asma, Walid, Steve

Cérémonies

- **Sprint Planning** : définition des objectifs et du backlog du sprint.
- **Daily Scrum** : suivi quotidien de l'avancement et identification des blocages.
- **Sprint Review** : démonstration des fonctionnalités livrées et collecte de feedback du PO.
- **Sprint Retrospective** : discussion sur les points positifs et axes d'amélioration.

4.2 Stratégies d'équipe avancées

Au-delà du Scrum classique, nous avons mis en œuvre des stratégies spécifiques pour optimiser notre vélocité :

- **Complexity Matching** : À partir du Sprint 3, nous avons adopté cette méthode consistant à associer les tâches les plus complexes (algorithmes NLP) aux profils techniques les plus adaptés, tandis que les tâches d'intégration étaient réparties autrement. Cela a permis de réduire les risques de blocage sur le cœur technique.
- **Pair Programming (Driver & Navigator)** : Sur les modules critiques (calcul de similarité, résumé), nous avons travaillé en binôme. Un développeur tenait le rôle de Driver (codage) et l'autre de Navigator (relecture, validation logique). Cela a drastiquement réduit le nombre de bugs lors des revues.
- **Le principe du "Joker"** : Lors du Sprint 4, pour pallier des absences ou des surcharges, nous avons instauré un système de "Joker" permettant à un membre d'intervenir ponctuellement sur une tâche hors de son périmètre habituel pour débloquer un coéquipier.

5. Synthèse des sprints réalisés

Pour visualiser l'avancement du projet, nous présentons un tableau récapitulatif des sprints, des fonctionnalités travaillées, des statuts et des difficultés rencontrées :

Sprint	User Stories (US) principales	Statut	Difficultés rencontrées	Livrables / Incréments
1	US1 – Page d'accueil US2 – Vecteurs de mots	Fait	Première appropriation de Scrum, communication initiale	Page d'accueil fonctionnelle, visualisation top 10 mots
2	US3 – TF/IDF US4 – Similarité texte	Fai	Estimation du temps trop optimiste pour TF-IDF, synchronisation front/back	Module TF-IDF backend fonctionnel, menu principal mis à jour
3	US4 – Similarité texte US5 – Résumé automatique	Fait	Intégration front-end lente, tests insuffisants	Algorithmes de résumé paramétrable, tests unitaires, interface partiellement intégrée
4	US5 – Méthodes de résumé & Snippets US6 – Ponctuation automatique	Fait	Limitation du module de ponctuation, absence d'un membre	Méthodes de résumé finalisées, Snippet 50 mots implémenté, interface web partielle
5	US6 – Top-5 snippets US7 – Ponctuation probabiliste	Fait	Répartition des tâches moins équilibrée, synchronisation front/back	Snippets best + Top-5 générés, module ponctuation terminé côté backend, page web fonctionnelle pour snippets

5.1 Rétrospective globale des sprints

Phase 1 : Lancement et Calibrage (Sprints 1 & 2) Les premiers sprints ont été marqués par une phase d'apprentissage.

- **Réussite :** La mise en place rapide d'une interface web et des premiers calculs de vecteurs.
- **Difficulté :** Nous avons péché par optimisme (Biais d'optimisme) lors du Sprint 2, notamment sur l'estimation du temps nécessaire pour implémenter le TF-IDF avec le corpus Wikipédia. Cela a conduit au report de certaines *User Stories* (US3).

Phase 2 : Optimisation et Performance (Sprints 3 & 4) Suite aux retards du début, nous avons restructuré notre approche.

- **Adaptation :** L'introduction du *Complexity Matching* et du *Pair Programming* a permis de livrer les algorithmes de résumé et de similarité dans les temps.
- **Expérimentation DevOps (CI/CD) :** Au Sprint 4, nous avons tenté de mettre en place une Intégration Continue (CI Light). Cependant, nous avons conclu lors de la rétrospective que cette

approche était trop lourde ("overkill") pour des séances de 3h. Nous avons donc décidé de revenir à des tests plus simples mais exécutés systématiquement, illustrant notre capacité à adapter les outils au contexte réel du projet.

Phase 3 : Consolidation et Limites Techniques (Sprint 5) Le dernier sprint s'est concentré sur des fonctionnalités avancées (Snippets Top-5, Ponctuation).

- **Défi technique :** Le module de ponctuation a révélé les limites de nos modèles. Nous avons implémenté une logique probabiliste (seuil de probabilité sur des bigrammes/quadrigrammes) , mais nous avons réalisé qu'un modèle robuste nécessiterait un volume de données d'entraînement inatteignable dans le cadre de ce TP.

6. Méthodes de travail et organisation

- Travail organisé par **pair programming** et **Complexity Matching** : les tâches complexes sont attribuées aux membres les plus expérimentés.
- Mise en place du **principe du Joker** pour faciliter l'entraide et pallier les absences.
- Communication quotidienne via WhatsApp et points réguliers toutes les 10 minutes lors des TP.

7. Difficultés rencontrées

- Difficultés initiales de communication, améliorées au fil du temps.
- Rédaction des rapports parfois diluée avec du contenu annexe non lié aux fonctionnalités.
- Appropriation progressive de la méthodologie Scrum par l'ensemble de l'équipe.

8. Bilan général

- **Avancement technique :** Les modules fondamentaux du projet (vecteurs, TF-IDF, résumé, snippets, ponctuation) ont été développés et testés sur des textes types. Les interfaces web permettent une interaction utilisateur fluide avec les modules implémentés.
- **Appropriation de la méthode Scrum :**
    - Mise en place effective des cérémonies : Daily, Sprint Review et Retrospective.
    - Application progressive des concepts de backlog, définition de DoD et planning de sprint.
    - Amélioration continue dans la communication et la coordination des tâches.
- **Points forts :**
    - Travail collaboratif efficace via pair programming et complexity matching.
    - Gestion des absences anticipée, utilisation du "joker" pour répartir les tâches critiques.
    - Tests unitaires systématiques sur chaque module pour garantir la fiabilité des fonctionnalités.
- **Difficultés :**
    - Synchronisation front/back parfois difficile, notamment pour l'affichage TF-IDF et snippets.

- Estimation du temps initiale trop optimiste sur certaines fonctionnalités.
- Intégration continue limitée par le temps de TP et la centralisation du code.

9. Perspectives et améliorations

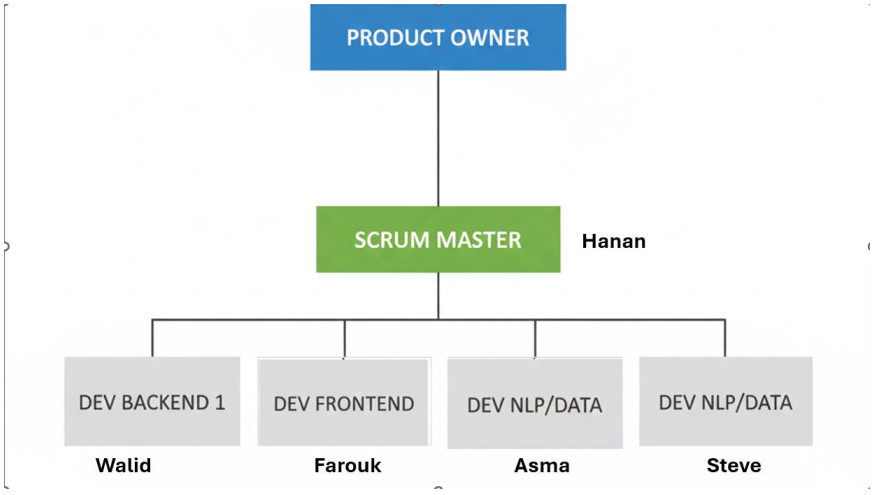
- **Techniques :**
  - Finaliser l'intégration front-end pour tous les modules et assurer une expérience utilisateur complète.
  - Développer le module de détection de plagiat et intégrer un LLM pour la reformulation de texte.
  - Optimiser la gestion des N-grammes et la ponctuation probabiliste pour une meilleure lisibilité.
- **Méthodologie / organisation :**
  - Renforcer les tests d'intégration et scénarios bout-en-bout.
  - Maintenir la pratique du pair programming pour les modules complexes.
  - Poursuivre l'amélioration continue via rétrospectives et ajustement du backlog pour prioriser les fonctionnalités critiques.

CONCLUSION

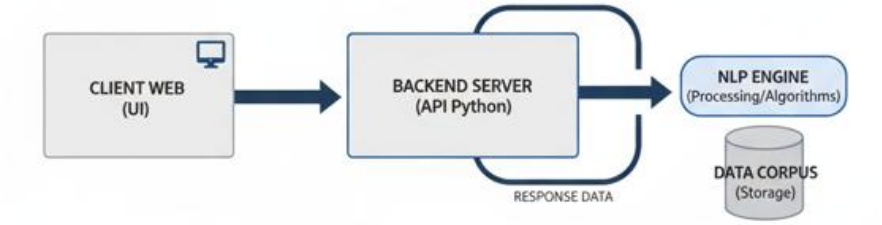
Le projet nous a permis de comprendre que Scrum n'est pas une « recette miracle », mais un cadre qui expose rapidement les problèmes et oblige à l'adaptation. **Au-delà des fonctionnalités techniques, nous retenons surtout que l'agilité véritable réside dans la capacité à remettre en question ses propres processus** quitte à simplifier, répartir autrement, ou modifier nos outils. **Cette expérience nous a préparés à aborder des projets plus ambitieux, en gardant à l'esprit que la flexibilité méthodologique est aussi importante que la rigueur technique.**

Annexes

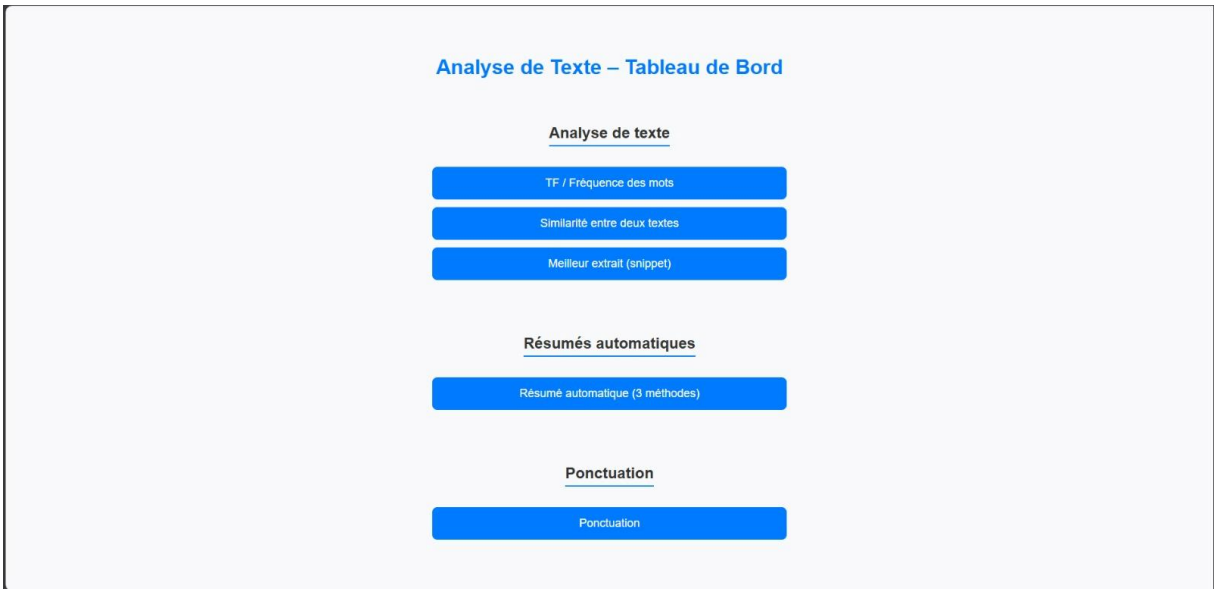
Annexe 1 : Organigramme de l'équipe



Annexe 2 : Workflow architecture générale



Annexe 3 : Captures d'écran de l'application Web



Moteur de Recherche de Snippets

1. Document (Texte source)

Le sens figuré d'éléments de langage organisés et enchaînés apparaît avant l'Empire romain : il désigne un agencement particulier du discours. Exemple : « epistolas texere = composer des épîtres » - Cicéron (Ier siècle av. J.-C.)[4] ou plus nettement chez Quintilien (Ier siècle apr. J.-C.) : « verba in textu jungantur = l'agencement des mots dans la phrase »[5].

Les formes anciennes du Moyen Âge désignent au XIIe siècle le volume qui contient le texte sacré des Évangiles, puis au XIIIe siècle, le texte original d'un livre saint ou des propos de quelqu'un. Au XVIIe siècle le mot s'applique au passage d'un ouvrage pris comme référence et au début du XIXe siècle le mot texte a son sens général d'« écrit »[6].

2. Requête (Mots clés)

texte

Taille Fenêtre (mots)

3

Nombre de phrases (Top K)

3++

Activer la pondération TF-IDF (si décoché : comptage simple)

Meilleur Passage (Fenêtre) | Top Phrases Clés

Meilleur Passage (Contexte continu)

Score: **0.7631** TF-IDF | Position mot: 0

"« Texte »"

← Retour

Résumé automatique

Texte à résumer :

»). Le verbe a aussi le sens large de construire comme dans « basilicam texere = construire une basilique » chez Cicéron[3].

Le sens figuré d'éléments de langage organisés et enchaînés apparaît avant l'Empire romain : il désigne un agencement particulier du discours. Exemple : « epistolas texere = composer des épîtres » - Cicéron (Ier siècle av. J.-C.)[4] ou plus nettement chez Quintilien (Ier siècle apr. J.-C.) : « verba in textu jungantur = l'agencement des mots dans la phrase »[5].

Les formes anciennes du Moyen Âge désignent au XIIe siècle le volume qui contient le texte sacré des Évangiles, puis au XIIIe siècle, le texte original d'un livre saint ou des propos de quelqu'un. Au XVIIe siècle le mot s'applique au passage d'un ouvrage pris comme référence et au début du XIXe siècle le mot texte a son sens général d'« écrit »[6].

Pourcentage à garder : 20 %

Méthode de résumé : Premières phrases

Générer le résumé

« Texte » est issu du mot latin « textum », dérivé du verbe « texere » qui signifie « tisser ». Le mot s'applique à l'entrelacement des fibres utilisées dans le tissage, voir par exemple Ovide : « Quo super iniecit textum rude sedula Baucis = (un siège) sur lequel Baucis empressée avait jeté un tissu grossier »[2] ou au tressage (exemple chez Martial « Vimineum textum = panier d'osier tressé »).

Moteur de Recherche de Snippets

1. Document (Texte source)

Le sens figuré d'éléments de langage organisés et enchainés apparait avant l'Empire romain : il désigne un agencement particulier du discours. Exemple : « epistolas texere = composer des épitres » - Cicéron (Ier siècle av. J.-C.)[4] ou plus nettement chez Quintilien (Ier siècle apr. J.-C.) : « verba in textu jungantur = l'agencement des mots dans la phrase »[5].

Les formes anciennes du Moyen Âge désignent au XIIe siècle le volume qui contient le texte sacré des Évangiles, puis au XIIIe siècle, le texte original d'un livre saint ou des propos de quelqu'un. Au XVIIe siècle le mot s'applique au passage d'un ouvrage pris comme référence et au début du XIXe siècle le mot

2. Requête (Mots clés)

texte

Taille Fenêtre (mots)

3

Nombre de phrases (Top K)

3

Activer la pondération TF-IDF (si décoché : comptage simple)

Meilleur Passage (Fenêtre) | Top Phrases Clés

Top NaN Phrases Pertinentes

#1 | Score: 1.5263 | Index phrase: 7

"Les formes anciennes du Moyen Âge désignent au XIIe siècle le volume qui contient le texte sacré des Évangiles, puis au XIIIe siècle, le texte original d'un livre saint ou des propos de quelqu'un."

#2 | Score: 0.7631 | Index phrase: 0

"« Texte » est issu du mot latin « textum », dérivé du verbe « texere » qui signifie « tisser »."

#3 | Score: 0.7631 | Index phrase: 8

"Au XVIIe siècle le mot s'applique au passage d'un ouvrage pris comme référence et au début du XIXe siècle le mot texte a son sens général d'« écrit »[6]."

#4 | Score: 0 | Index phrase: 1

"Le mot s'applique à l'entrelacement des fibres utilisées dans le tissage, voir par exemple Ovide : « Quo super iniecit textum rude sedula Baucis = (un siège) sur lequel Baucis empressée avait jeté un tissu grossier »[2] ou au tressage (exemple chez Martial « Vimineum textum = panier d'osier tressé"

#5 | Score: 0 | Index phrase: 2

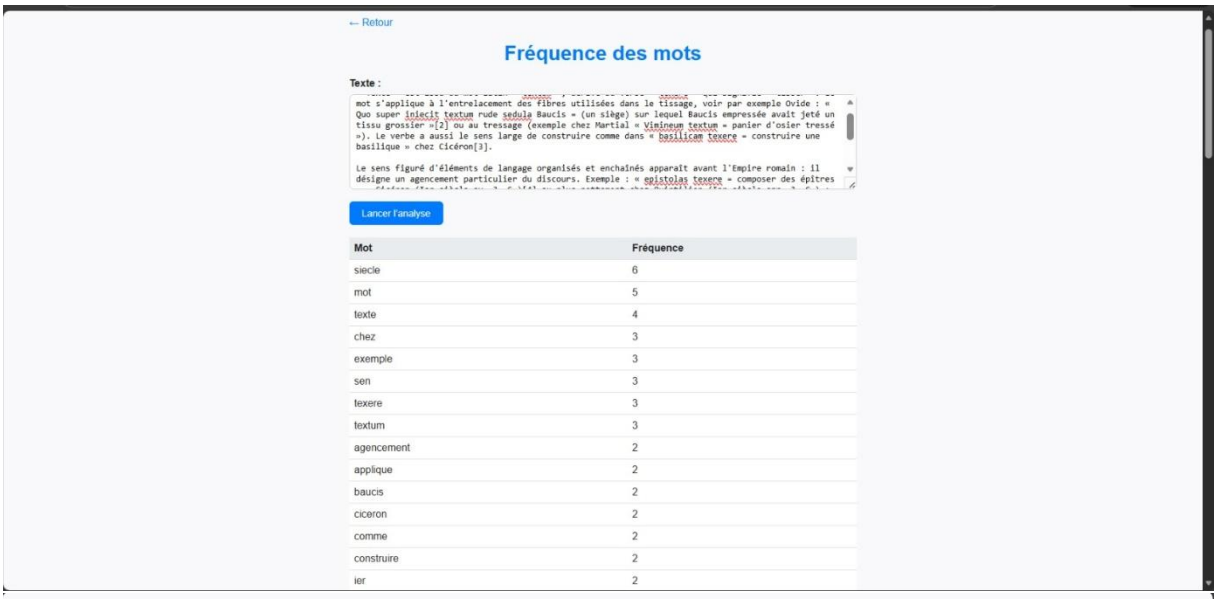
"Le verbe a aussi le sens large de construire comme dans « basilicam texere = construire une basilique » chez Cicéron[3]."

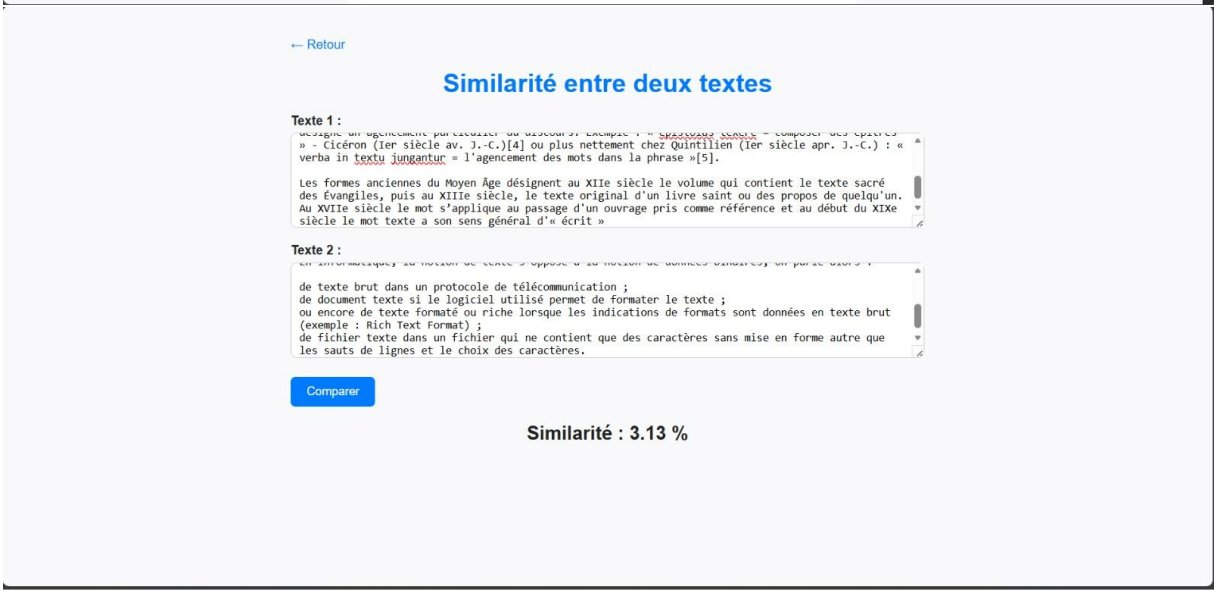
#6 | Score: 0 | Index phrase: 3

"Le sens figuré d'éléments de langage organisés et enchaînés apparaît avant l'Empire romain : il désigne un agencement particulier du discours."

#7 | Score: 0 | Index phrase: 4

"Exemple : « epistolas texere = composer des épîtres » - Cicéron (Ier siècle av."





← Retour au menu

Ponctuation automatique

Texte sans ponctuation :

il était une fois dans un pays lointain un roi sage et juste
il gouvernait avec bonté son peuple était heureux et prospère
un jour un dragon terrifiant apparut dans les montagnes il semait la terreur

Ajouter la ponctuation

Résultat :

Il était une fois. Dans un pays loin. Tain. Un roi sa ge et juste, il gou vern ait avec bon, té, son peuple était heureux et prosp, ère, un jour. Un drago n ter ri fiant appar ut dans les montagne, s, il sem ait la terre, ur,.