

# Sprint backlog 1

Date de début : @10 novembre 2025

- Doit contenir

- ~~Objectif globale~~
- ~~Tâches précises~~
- ~~Definition of done~~

review avec le client de ce qui marche + retrospective ce qu'on peut améliorer pour le prochain

## Sprint Backlog

- En tant qu'utilisateur, je veux une **page d'accueil web claire** avec les 4 modules
- En tant qu'utilisateur, je veux **afficher les vecteurs de mots** (mots les plus fréquents en haut)

## Objectif ⇒ Affichage des mots les plus fréquents + vecteurs + mini page

- **Durée :** 3H
- Analyser un texte et afficher les mots les plus fréquents en haut dans un espace vectoriel qui dépend d'une manière précise choisie de vectorisation

### Planification des Tâches

Aa Tâche	≡ Détails	👥 Responsable	≡ Temps	🕒 Statut	⌚ Priorité
<u>Créatio</u> <u>n du</u> <u>dépot</u> <u>Github</u>	Depot Github	 steve sanogo	5 min	Terminé	
<u>Choix</u> <u>du</u> <u>framew</u>	Sélectionner un framework léger et rapide ( REACT) pour créer la	(B) Blami Gnankambary		Terminé	

Aa Tâche	≡ Détails	👥 Responsable	≡ Temps	⌚ Statut	⌚ Priorité
<u>ork</u> <u>web</u>	première version de l'application.				
<u>Mise</u> <u>en</u> <u>place</u> <u>du</u> <u>prétrait</u> <u>ement</u> <u>du</u> <u>texte</u>	Créer une fonction qui nettoie le texte : convertir en minuscules, retirer la ponctuation et les caractères spéciaux. Cette étape prépare les données avant le traitement linguistique.	(A) Asma Nihal Boukraa (H) Hanan Boud		Terminé	
<u>Lemma</u> <u>tisation</u> <u>et</u> <u>suppre</u> <u>ssion</u> <u>des</u> <u>stopwo</u> <u>rds</u>	Utiliser la bibliothèque spaCy pour transformer chaque mot en sa forme de base (ex. "mangé" → "manger") et supprimer les mots très fréquents et peu informatifs (ex. "le", "la", "de"). Le résultat est une liste de mots "utiles".	(A) Asma Nihal Boukraa (B) Blami Gnankambary		Terminé	
<u>Compt</u> <u>age et</u> <u>vectori</u> <u>sation</u> <u>combin</u> <u>és</u>	Calculer simultanément : • la <b>fréquence brute</b> de chaque mot (combien de fois il apparaît), • le <b>score TF-IDF</b> (son importance dans le texte). Combiner les deux résultats dans un tableau unique trié d'abord par fréquence, puis par score TF-IDF	(A) Asma Nihal Boukraa (F) FARAH Mohamed Walid		Terminé	

Aa Tâche	≡ Détails	👥 Responsable	≡ Temps	⌚ Statut	⭐ Priorité
	pour obtenir les mots à la fois fréquents et représentatifs.				
<u>Création d'une première page web</u>	Créer une page minimale avec un champ texte et un bouton "Analyser". Lorsqu'un texte est soumis, le backend applique le traitement et affiche le tableau des mots avec leurs fréquences et leurs scores.	(B) Blami Gnankambaray		Terminé	
<u>Test et vérification finale</u>	Tester l'analyse sur plusieurs textes français (courts et longs) pour s'assurer que les résultats sont cohérents : les mots les plus fréquents apparaissent en haut et les scores TF-IDF sont bien calculés.	(A) Asma Nihal Boukraa (F) FARAH Mohamed Walid (B) Blami Gnankambaray		En cours	

## Definition of done

La tache est terminé lorsque :

- Le projet démarre avec un framework fonctionnel et une page web simple,
- Le texte est correctement nettoyé, lemmatisé et filtré,
- Afficher les vecteurs de mots les plus fréquents en haut

- Le code fonctionne sans erreur avec plusieurs exemples de texte.

## **Review**

- 

## **Retro**

-

# Sprint backlog 2

Date de début : @10 novembre 2025

- Doit contenir

- ~~Objectif globale~~
- ~~Tâches précises~~
- ~~Definition of done~~

review avec le client de ce qui marche + retrospective ce qu'on peut améliorer pour le prochain

## Objectif ⇒ Affichage la similarité entre deux textes + les métriques TF IDF pour les mots

- **Durée :** 3H
- Analyser un texte et afficher les mots les plus fréquents en haut + leur métriques TF et IDF et , affichage la similarité entre deux textes

## User Stories

- En tant qu'utilisateur, je veux une **page d'accueil web claire** avec le module similarité de texte
- En tant qu'utilisateur, je veux **L'affichage de TF et IDF**
- En tant qu'utilisateur, je veux **afficher la similarité entre deux textes**

## Tâches

- Calculer TF IDF avec wikipedia ⇒ hanane + steve
- Calcul de similarité entre textes ⇒ asma + steve
- Afficher TF IDF ⇒ walid + farouq
- Affichage de la page web pour le module similarité entre textes ⇒ walid et farouq
- Modification de menu d'entrée pour intégrer similarité ⇒ walid + farouq

## Definition of done

- La page principale est accessible et affiche 2 entrées claires :
  - Similarité de texte
  - TF-IDF avec Wikipédia
- Le module TF-IDF :
  - permet de saisir un texte,
  - calcule TF, IDF des mots par rapport à un corpus Wikipédia,
  - affiche le résultat.
- Le module Similarité de texte :
  - permet de saisir deux textes,
  - calcule leur similarité TF-IDF + cosinus,
  - affiche un score numérique de similarité.

## Review sur la production

ce qu'on a raté en terme de production

- Afficher les TF , IDF , TF-IDF basé sur wikipedia et sur le texte lui même sur la page
- git pull pour les autres
- dans IDF dans wikipedia , TF sur le texte lui même ⇒ cosinus entre les deux
- Affichage du 10 premiere phrase
- resume top 10% des phrases vont être celle les plus similaires avec l'ensemble de texte
- afficher 10 plus qui ont plus de similarité , classé pa score par ordre d'apparition
- choisir le pourcentage de résumé

## Retro sur management

- améliorer la gestion de temps
-

# Sprint backlog 3

Date de début : @10 novembre 2025

- Doit contenir

~~Objectif globale~~

~~Tâches précises~~

~~Definition of done~~

review avec le client de ce qui marche + retrospective ce qu'on peut améliorer pour le prochain

## timing

Ao taches	Date	Etat
<u>rectification de TF-IDF et des stopwords</u>	@4 décembre 2025 16:00 → 17:30	 Hanan Boud Terminé
<u>résumé les 3 méthodes</u>	@4 décembre 2025 16:00 → 17:30	 Asma Nihal Boukraa  steve sanogo Terminé
<u>page du résumé + affichage du résultat</u>	@4 décembre 2025 16:00 → 17:30	 FARAH Mohamed Walid Terminé

**Objectif ⇒ Affichage la similarité entre deux textes + les métriques TF IDF pour les mots**

- Durée : 3H

- Analyser un texte et afficher les mots les plus fréquents en haut + leur métriques TF et IDF et , affichage la similarité entre deux textes

## User Stories

- En tant qu'utilisateur, je veux une avoir 3 méthodes de résumé
- En tant qu'utilisateur, je veux **L'affichage de TF et IDF**
- En tant qu'utilisateur, je veux **afficher la similarité entre deux textes**

## Tâches

- ~~Calculer TF-IDF avec wikipedia → hanane + steve~~
- ~~Calcul de similarité entre textes → asma + steve~~
- ~~Afficher TF-IDF → walid + farouq~~
- ~~Affichage de la page web pour le module similarité entre textes → walid et farouq~~
- ~~Modification de menu d'entrée pour intégrer similarité → walid + farouq~~

## Definition of done

- La page principale est accessible et affiche 2 entrées claires :
  - Similarité de texte
  - TF-IDF avec Wikipédia
- Le module TF-IDF :
  - permet de saisir un texte,
  - calcule TF, IDF des mots par rapport à un corpus Wikipédia,
  - affiche le résultat.
- Le module Similarité de texte :
  - permet de saisir deux textes,
  - calcule leur similarité TF-IDF + cosinus,

- affiche un score numérique de similarité.
- 

## Review sur la production

ce qu'on a raté en terme de production

- Afficher les TF , IDF , TF-IDF basé sur wikipedia et sur le texte lui même sur la page
- git pull pour les autres
- dans IDF dans wikipedia , TF sur le texte lui même ⇒ cosinus entre les deux
- Affichage du 10 première phrase
- resume top 10% des phrases vont être celle les plus similaires avec l'ensemble de texte
- afficher 10 plus qui ont plus de similarité , classé par score par ordre d'apparition
- choisir le pourcentage de résumé

## Retro sur management

- améliorer la gestion de temps

# Rapport de Sprint 4 – Snippet, Ponctuation & Amélioration Agile

---

## Projet : Bibliothèque de vectorisation, similarité et résumé de texte

**Sprint concerné :** Sprint 4 – Consolidation des modules Résumé & Similarité

**Durée :** séances de TP (3h)

**Équipe :** Asma, Steve, Hanane, Farouk

---

## 1. Objectifs du Sprint 4

Suite aux retours des sprints précédents, les objectifs de ce sprint étaient les suivants :

- Corriger définitivement les **méthodes 2 et 3 de résumé** afin de clarifier leur logique respective (classement par score vs ordre du texte).
  - Mettre en place une nouvelle fonctionnalité de type **Snippet**, permettant d'extraire la phrase classée n°1 et de la limiter à un extrait de **50 mots** maximum.
  - Développer une fonction de **ponctuation automatique**, capable de restituer un texte brut sans ponctuation avec des signes de ponctuation pertinents.
  - Finaliser l'intégration des modules côté **back-end et front-end**.
  - Tester un mode de fonctionnement intégrant une première approche d'**intégration continue (CI light)**.
- 

## 2. User Stories principales travaillées

Les besoins fonctionnels du sprint ont été formalisés sous forme de user stories :

---

### **US1 – Correction des méthodes de résumé 2 & 3**

*En tant qu'utilisateur, je veux disposer de deux méthodes de résumé distinctes afin de comparer un classement par pertinence et une restitution dans l'ordre d'apparition du texte.*

#### **Definition of Done :**

Les méthodes 2 et 3 doivent produire des résumés différents sur les mêmes textes tests :

- la méthode 2 classe les phrases par score de similarité décroissant ;
- la méthode 3 sélectionne les mêmes phrases mais les restitue dans l'ordre d'apparition du texte.

Des tests unitaires valident la cohérence des résultats et l'absence de doublons.

---

### **US2 – Génération d'un Snippet**

*En tant qu'utilisateur, je veux obtenir automatiquement un extrait (snippet) correspondant à la phrase la plus pertinente du texte afin de visualiser rapidement l'idée principale.*

#### **Definition of Done :**

Le système doit :

- déterminer la phrase classée n°1 selon la similarité TF-IDF ;
- limiter l'extrait à **50 mots maximum** ;
- afficher le snippet dans l'interface.

Des tests valident le bon respect de la longueur maximale.

---

### **US3 – Ponctuation automatique**

*En tant qu'utilisateur, je veux transformer un texte dépourvu de ponctuation en un texte lisible afin d'en améliorer la compréhension.*

#### **Definition of Done :**

La fonctionnalité doit :

- analyser un texte brut sans ponctuation ;

- insérer les principaux signes de ponctuation (. , ? !) ;
- produire un texte grammaticalement lisible.

La sortie est validée manuellement et par tests simples comparatifs.

---

#### **US4 – Intégration Web**

*En tant qu'utilisateur, je veux accéder à l'ensemble des fonctionnalités depuis une interface web unique pour tester facilement la similarité, le résumé et le snippet.*

##### **Definition of Done :**

L'interface doit permettre :

- la saisie de textes ;
- l'exécution des modules de similarité, résumé et snippet ;
- l'affichage clair des résultats.

Les échanges front/back doivent être fonctionnels et validés par des tests d'intégration légers.

---

### **3. Organisation du travail et répartition des tâches**

La répartition a suivi une stratégie combinant **Complexity Matching**.et l'utilisation du **Joker d'affectation des tâches**.

#### **Principe du Joker**

Le *joker* a permis à chaque membre de l'équipe d'intervenir ponctuellement sur une tâche différente de sa responsabilité principale afin de :

- soulager un coéquipier en difficulté,
- renforcer la dynamique d'entraide,
- éviter les blocages en cas d'imprévu.

#### **Suivi des tâches**

Aa Nom de la tâche	Etat	Personne assignée	Date d'échéance	Priorité
<u>Page web snippets + ponctuation de texte</u>	En cours	(B) Blami Gnankambary	@02/03/2025 14:30 → 16:30	Élevée
 <u>Snippets</u>	Terminé	(H) Hanan Boud	@02/20/2025 14:30 → 16:30	Élevée
 <u>Correctio n des problèmes de résumé</u>	Terminé	(A) Asma Nihal Boukraa	@02/28/2025 14:30 → 16:30	Moyenne
 <u>Module de ponctuation de texte</u>	Terminé	(s) steve sanogo	@12/08/2025 14:30 → 16:30	Élevée

## 5. Sprint Review

### Fonctionnalités terminées

- Correction complète et validation des **méthodes 2 et 3 de résumé** :
- Implémentation du module de **Snippet** :
  - sélection de la phrase la plus pertinente,
  - extraction d'un **extrait brut de 50 mots maximum**,
- Développement d'un **premier module de ponctuation automatique** capable de transformer un texte brut (sans ponctuation) en texte plus lisible.

### Limites et points constatés

- Pour le **module de ponctuation**, l'équipe a identifié que :
  - un modèle vraiment robuste nécessiterait **beaucoup de données d'entraînement** et un temps important de calibration,
  - ce niveau de qualité n'est **pas atteignable dans une seule séance de 3h.**

Le module actuel est donc une **première version** démonstrative, à améliorer ultérieurement
- Un membre de l'équipe était **absent** lors de la séance, ce qui a limité :
  - la **visualisation complète** des résultats côté front-end de snippets ,
  - la possibilité de corriger rapidement certains problèmes d'affichage liés à son code.

## Reste à finaliser

- Stabiliser l'**affichage front-end** (mise en page, remontée des résultats, présentation du snippet et du résumé).
- **Renforcer les tests d'intégration :**
  - scénarios de bout en bout (saisie → back-end → résultats),
  - validation manuelle collective en fin de sprint.
- Résultat de snippets est inexploitable côté interface.
- Centraliser progressivement l'exécution autour d'un **main unique** pour faciliter les futurs pipelines automatisés.
- L'approche **CI light / pipeline** testée dans ce sprint s'est révélée **trop lourde** pour le format de TP :
  - trop de temps passé à la config par rapport au temps disponible de 3h,
  - synchronisation compliquée tant que le code n'est pas suffisamment centralisé.
- Le nombre de **tests** reste insuffisant :
  - peu de cas "utilisateur complet" (saisie → traitement → affichage),
  - pas encore de vraie habitude de lancer systématiquement les tests avant la fin de la séance.

## 6. Rétrospective

### Points positifs

- L'équipe a réalisé une **grande avancée fonctionnelle** sur les modules cœur (résumé, snippet, ponctuation), ce qui rapproche le projet d'une version exploitable de bout en bout.
- La répartition des tâches est restée **globalement équitable** : chaque membre a alterné entre des tâches plus simples et des tâches plus techniques, ce qui a permis à tout le monde de progresser.
- Le principe du **joker** a facilité l'entraide : certains blocages ont pu être levés rapidement en laissant un autre membre reprendre temporairement une tâche.
- L'équipe a su **identifier clairement les limites** du module de ponctuation (besoin de données, de temps, de modèle plus poussé), ce qui prépare bien les axes d'amélioration pour la suite.

### Points à améliorer

- Il y a eu une **marge plus importante que dans les séances précédentes** entre les tâches : certaines parties ont beaucoup avancé, alors que d'autres (notamment l'interface) ont pris du retard.  
Résultat : nous n'avons pas pu tester **toute la chaîne complète** (back + front) en fin de séance.
- L'absence d'un membre a accentué ce décalage, surtout pour la **visualisation des résultats** côté front-end et la correction en direct des problèmes d'affichage.

### Décisions pour le prochain sprint

- **Renforcer les tests :**
  - ajouter davantage de tests unitaires sur résumé, similarité, snippets et ponctuation,
  - définir 2–3 scénarios d'utilisation complets à tester systématiquement en fin de séance.

- **Mieux synchroniser les tâches :**
  - éviter qu'une seule personne soit bloquante sur l'interface,
  - s'assurer que les tâches "front" et "back" avancent de manière plus parallèle pour pouvoir tester l'ensemble.
- **Simplifier l'approche d'intégration :**
  - garder un pipeline **léger**, centré sur un main unique et quelques tests automatiques,
  - éviter de mettre en place une CI trop complexe par rapport au volume horaire des TP

### **Planification :**

plus de tests sur les snippets , et les 5 premiers snippets

bigramme , prediction du mot apres un mot , probab que apres cest un point  
faut choisir valeur de N (gramme )

n=4

pon/pas ponc ⇒ donc si ca depasse le seuil alors , ponctuation ....

complexity matching

# Rapport de Sprint 5 – Snippet, Ponctuation & Amélioration Agile

---

## Projet : Bibliothèque de vectorisation, similarité et résumé de texte

**Sprint concerné :** Sprint 4 – Consolidation des modules Résumé & Similarité

**Durée :** séances de TP (3h)

**Équipe :** Walid, Hanane, Farouk

---

## 1. Objectifs du Sprint 5

Suite aux retours des sprints précédents, les objectifs de ce sprint étaient les suivants :

- **Améliorer le module de ponctuation** en introduisant une logique de prédiction basée sur les fréquences / probabilités entre mots (ngrams) et en tranchant sur un seuil : au-delà d'une certaine probabilité, on insère une ponctuation, sinon on s'abstient.
  - **Étendre le module de snippets** pour afficher, en plus du meilleur extrait, les 5 premiers snippets les plus pertinents (Top-5).
  - **Tester snippets et ponctuation sur des textes de grande taille** : snippets et ponctuation.
  - **Intégrer le module snippets et ponctuation dans une page web**
- 

## 2. User Stories principales travaillées

Les besoins fonctionnels du sprint ont été formalisés sous forme de user stories :

### US1 – Ponctuation avec seuil de probabilité

*En tant qu'utilisateur, je veux que la ponctuation automatique se base sur une probabilité calculée entre les mots (par exemple via des fréquences de paires de mots) afin d'éviter de sur-ponctuer ou de sous-ponctuer le texte.*

**Definition of Done :**

- un module qui calcule une probabilité de coupure entre mots (à partir de fréquences / TF-IDF) ;
- un seuil défini : si  $\text{proba} \geq \text{seuil}$  → on ajoute une ponctuation, sinon non ;
- des tests sur des phrases sans ponctuation, avec comparaison avant / après (et idéalement un texte de référence) ;
- quelques exemples documentés montrant que le résultat est lisible.

## US2 – Top-5 Snippets

*En tant qu'utilisateur, je veux voir, en plus du meilleur snippet, les 5 extraits les plus pertinents d'un texte afin d'avoir une vue plus riche des passages importants.*

**Definition of Done :**

- garder le best snippet comme avant ;
- ajouter le calcul des 5 meilleurs snippets à partir des scores TF-IDF / similarité ;
- garantir que chaque snippet fait au plus 50 mots, est trié par score et sans doublons évidents ;
- une fonction Python (ex. `find_top_k_snippets`) testée sur des textes longs ;
- affichage du best + Top-5 dans les tests (console) et prêt à être utilisé côté web.

## US3 – Tests sur textes de grande taille

*En tant que développeur / enseignant, je veux que la bibliothèque soit testée sur des textes longs proches de cas réels afin de vérifier la robustesse des modules (résumé, similarité, snippets, ponctuation).*

**Definition of Done :**

- un ensemble de textes longs (type articles) pour les tests ;
- un ou plusieurs scripts qui enchaînent automatiquement :
  - TF-IDF → similarité → résumés → snippets (best + Top-5) → ponctuation ;
- pour chaque texte : affichage des principaux scores, d'un résumé, des snippets et d'un exemple de texte ponctué ;
- tests faisables en local en une séance de TP et utilisés comme validation globale ;
- les problèmes majeurs sont repérés et au moins documentés.

## US4 – Page web pour Snippets et Ponctuation

*En tant qu'utilisateur, je veux disposer d'une page web me permettant de tester la génération de snippets (best + Top-5), avec ou sans TF-IDF, ainsi que la ponctuation automatique, afin de visualiser le résultat des traitements sans passer par la console.*

### Definition of Done (version simplifiée) :

- une page web où on peut saisir un texte brut ;
- pouvoir :
  - générer des snippets en mode sans TF-IDF et avec TF-IDF, avec affichage du best snippet + Top-5 (et leurs scores) ;
  - lancer la ponctuation automatique sur un texte sans ponctuation ;
- résultats affichés dans deux zones : Snippets et Ponctuation ;
- les appels back-end fonctionnent (snippets + ponctuation) et gèrent au moins le cas "texte vide" ;
- un test complet en équipe a été fait pour vérifier que :
  - les snippets web = snippets Python,
  - le texte ponctué sur la page = sortie du module de ponctuation.

## 3. Organisation du travail et répartition des tâches

### Suivi des tâches

Aa Nom de la tâche	Etat	Personne assignée	Date d'échéance	Priorité
<u>Page</u> <u>web</u> <u>snipp</u> <u>ets +</u> <u>ponc</u> <u>tua</u> <u>on</u>	Ter miné	(B) Blami Gnankambary (F) FARAH Mohamed Walid	@02/03/2025 16:30 → 12/15/2025 14:30	Élevée
 <u>Snip</u> <u>pets</u> <u>corre</u> <u>ction</u> <u>des</u> <u>k-top</u> <u>snipp</u> <u>ets</u>	Ter miné	(H) Hanan Boud	@02/20/2025 14:30 → 16:30	Élevée

Aa Nom de la tâche	Etat	Personne assignée	Date d'échéance	Priorité
<span style="color: green;">✓</span> <u>Mod</u> <u>ule</u> <u>de</u> <u>ponc</u> <u>tuati</u> <u>on</u> <u>de</u> <u>texte</u>	Terminé	s steve sanogo	@12/08/2025 14:30 → 16:30	Élevée

## 5. Sprint Review

### Fonctionnalités terminées :

- Snippets : génération du best snippet et des 5 top snippets, avec option pour le mode avec TF-IDF et sans TF-IDF.
- Page Web : interface web fonctionnelle pour tester les snippets et la ponctuation automatique (sans ponctuation).
- Ponctuation : module de ponctuation automatique terminé, mais pas encore la page web dédiée pour l'afficher sur l'interface.
- Page Web pour Ponctuation

### Fonctionnalités restantes :

- Tests globaux : il faudra effectuer des tests d'intégration sur tout le pipeline :
  - TF-IDF → Résumé → Similarité → Ponctuation.
- Rendre le code clean sur github et merger les branches
- Voir avec le collègue a qui est attribué le module de ponctuation la méthodologie de son travail (pourquoi il fait juste des points)

## 6. Rétrospective

### Points positifs :

- Temps bien géré par rapport au sprint derniers

### Points d'amélioration :

- Répartition non équitable des tâches : chaque membre a contribué de manière moins équilibrée par rapport au sprints derniers
- Absences injustifiées : certains membres étaient absents sans explication, ce qui a pu freiner certaines avancées dans la collaboration et les tests.
- Synchronisation des tâches : bien que la répartition des tâches ait été équilibrée, il faut mieux synchroniser les efforts

#### **Décisions pour le prochain sprint :**

- Meilleure synchronisation des tâches entre les membres
- Réaliser un test global en équipe : tous les membres se réunissent, lancent les différents modules et identifient les derniers problèmes à corriger

#### **Planification :**

plus de tests sur les snippets , et les 5 premiers snippets

bigramme , prediction du mot apres un mot , probab que apres cest un point  
faut choisir valeur de N (gramme )

n=4

pon/pas ponc ⇒ donc si ca depasse le seuil alors , ponctuation ....

complexity matching

# Rapport de Sprint 6 – Finalisation, Documentation & Préparation Démo

---

## Projet : Bibliothèque de vectorisation, similarité et résumé de texte

**Sprint concerné :** Sprint 6 – Clôture et Livrables finaux

**Durée :** Dernière séance de TP (3h)

**Équipe :** Walid, Hanane, Asma, Steve, Farouk

---

### 1. Objectifs du Sprint Final

Suite à la consolidation des modules de **résumé**, **similarité**, **snippets** et **ponctuation** lors des sprints précédents, les objectifs de ce sprint final étaient les suivants :

- Préparation de la **présentation finale** et de la **démonstration en direct**.
  - Rédaction du **rapport global**, compilant l'ensemble des travaux techniques et organisationnels
  - Réalisation des **captures d'écran** des interfaces pour servir de support à la présentation
  - Finalisation du **feedback de l'équipe** sur le déroulement du projet et l'application de la méthode Agile.
- 

### 2. Organisation du travail et répartition des tâches

Pour ce dernier sprint, l'équipe a été divisée en **trois pôles de production** afin de couvrir l'ensemble des livrables :

Tâche	Responsables	État
Rédaction du rapport du sprint 6	Hanane Boud & Asma Nihal Boukraa	Terminé
Préparation de la démonstration et finalisation des interfaces	Farah Mohamed Walid & Farouk	Terminé
Rédaction du rapport global et consolidation du feedback	Steve Sanogo	Terminé
Captures d'écran pour la présentation	farouk	Terminé

### 3. Sprint Review

#### Fonctionnalités et livrables terminés

- **Pipeline complet :**

Le module global intégrant **TF-IDF**, **Résumé**, **Similarité**, **Snippets** et **Ponctuation** est pleinement opérationnel.

- **Interfaces Web :**

Les modules de résumé, similarité et snippets sont correctement connectés au backend et prêts pour la démonstration.

- **Documentation technique :**

Le rapport global détaille les choix algorithmiques, notamment l'utilisation du **TF-IDF sur un corpus Wikipédia** et des **n-grammes (n = 4)** pour la ponctuation.

- **Support de présentation :**

La présentation inclut une **rétrospective**, les **difficultés rencontrées** et les **enseignements tirés** du projet.

#### Reste à faire

- **Soutenance orale :**

Présentation du projet et démonstration en direct devant l'enseignant.

### 4. Rétrospective Finale

#### Points positifs

- **Maîtrise technique :**

Développement réussi d'algorithmes complexes grâce au **Pair Programming** et au **Complexity Matching**.

- **Communication :**

Coordination efficace de l'équipe lors de ce sprint final, malgré les difficultés de synchronisation rencontrées auparavant.

- **Gestion du temps :**

L'ensemble des fonctionnalités prévues (**US1 à US6**) a été livré dans les délais, avant la séance finale.

## Points à améliorer / Perspectives

- **Tests d'intégration :**

Bien que réalisés manuellement, la mise en place de **tests automatiques (CI/CD)** aurait renforcé la robustesse du projet.

- **Modèle de ponctuation :**

Le module actuel reste perfectible, notamment pour ajuster les probabilités des **n-grammes** et éviter la sur-ponctuation.

- **Évolutivité :**

L'intégration future d'un **LLM (US7)** pour la reformulation représente une perspective majeure d'amélioration.

---

## Conclusion du projet

Ce sprint de clôture a permis de transformer une série de modules techniques en une **bibliothèque cohérente, fonctionnelle et documentée**, prête pour une **exploitation réelle** et des évolutions futures.