Projet IA Générative
Analyse Sémantique et Système de Recommandation
Agent Intelligent Sémantique et Génératif

Annexe I - Thématiques alternatives acceptées

Epreuve certifiante RNCP40875 Expert en ingénierie de données (Bloc de compétences à valider : Bloc 2)

Veuillez soumettre votre solution de ce projet sur Moodle + Git/GitHub, en respectant la date limite fixée.
Toute tricherie ne sera pas tolérée et sera sanctionnée.
Vous devez travailler en binôme.

Introduction
Attention : A lire impérativement avant de choisir une thématique alternative

Veuillez vous référer en premier lieu au document principal intitulé « **Analyse Sémantique pour la Cartographie des Compétences et la Recommandation de Métiers / AISCA : Agent Intelligent Sémantique et Génératif pour la Cartographie des Compétences** », qui présente l'ensemble du cadre théorique, la logique méthodologique, les recommandations techniques et le cahier des charges complet du projet. Ce document constitue la source de référence pour toutes les exigences pédagogiques, techniques et évaluatives du projet.

Le document présent ne remplace en aucun cas le sujet principal mais sert uniquement de support complémentaire pour les étudiants qui souhaitent adapter la thématique et appliquer la même démarche, les mêmes exigences fonctionnelles et non fonctionnelles, la même architecture IA et la même logique de réalisation, dans un domaine différent de celui initialement proposé.

Il permet d'orienter les étudiants souhaitant proposer une autre application (cinéma, musique, santé, livres…) tout en respectant strictement la structure AISCA, les exigences RNCP40875 (Bloc 2), les contraintes d'IA Générative, la logique RAG et l'analyse sémantique SBERT. Les sections qui suivent illustrent simplement comment transposer AISCA vers d'autres thématiques et quelles variantes sont acceptées dans le cadre du même projet académique.

Objectifs Pédagogiques du Projet

En réalisant ce projet complet d'IA Générative et NLP sémantique, vous apprendrez à :

- appliquer le prétraitement de texte et les embeddings sémantiques ;
- distinguer l'analyse numérique (scores bruts) de l'analyse sémantique contextualisée ;

- implémenter un moteur de similarité basé sur SBERT ;
- structurer un référentiel de compétences (format professionnel) ;
- développer une interface web (Streamlit ou autre) ;
- intégrer la GenAI de manière responsable, contrôlée et économique ;
- concevoir un pipeline complet NLP / Scoring / Recommandation / GenAI ;
- travailler en équipe pour produire une solution professionnelle ;
- présenter un prototype en conditions proches du monde réel.

Compétences RNCP visées (RNCP40875 – Bloc 2)

Ce projet IA Générative constitue l'un des projets majeurs de l'année.
Ce projet permet de valider certaines compétences du Bloc 2 : BC2 - Piloter et implémenter des solutions d'IA en s'aidant notamment de l'IA générative du référentiel RNCP40875.

Selon le référentiel, ce bloc couvre notamment :
Compétences principales évaluées

- Collecter, analyser et préparer des données structurées et non structurées
- Concevoir et implémenter des modèles Data Science / NLP / IA
- Évaluer et optimiser les modèles
- Prototyper des solutions IA (API, NLP, RAG, embeddings, GenAI)
- Développer des pipelines data de bout en bout
- Industrialiser une solution (architecture, performance, contraintes coût)
- Documenter et présenter une solution technique complète

Compétences transverses également mobilisées :

- Travail en équipe
- Documentation technique
- Présentation orale
- Respect des exigences fonctionnelles et non fonctionnelles
- Sécurisation et gestion responsable de l'IA générative

<u>Liste synthétique des compétences clés mobilisées dans ce projet</u>
Voici certaines compétences concrètes que vous allez développer durant ce projet (et tout au long de l'année) :
Compétences Data Science / NLP

- Prétraitement textuel avancé
- Construction d'embeddings (SBERT)
- Calcul de similarité cosinus
- Structuration et nettoyage de référentiels

Compétences IA Générative

- Prompt engineering
- Intégration d'une API GenAI (Gemini/OpenAI)
- Génération contextuelle (RAG)
- Mise en place d'un caching local

Compétences Data Engineering

- Capture et stockage structuré des données

- Architecture logique de pipeline IA
- Optimisation des performances et coût
- Versioning du code (Git)

Compétences Software

- Développement d'interface utilisateur (Streamlit)
- Déploiement local d'une application IA
- Visualisation (graphiques, radars, tableaux)

Compétences Professionnelles

- Conception d'un MVP
- Analyse documentaire et justification des choix
- Reporting et documentation technique
- Présentation orale du produit final

Veuillez consulter la grille de notation détaillée afin d'avoir une vision globale de l'ensemble des compétences, critères d'évaluation et modalités de notation.

Charges Général applicable à toutes les thématiques

Description du projet

Dans toutes les thématiques proposées, les étudiants doivent appliquer exactement les mêmes principes techniques, pédagogiques et méthodologiques que dans le projet initial AISCA. Il s'agit de concevoir une application web intégrant un questionnaire à entrées textuelles, une analyse sémantique basée sur des embeddings contextuels, un calcul de similarité, une logique de scoring, un mécanisme de recommandation et une génération finale effectuée via une IA générative. Le sujet choisi peut varier, mais la démarche technique doit impérativement rester identique. Les décisions doivent donc être prises dans une logique de transfert, c'est-à-dire en réadaptant la taxonomie et la logique d'AISCA à un nouveau domaine.

Objectif

L'objectif général du projet, quelle que soit la thématique retenue, consiste à collecter des informations auprès d'un utilisateur, à les interpréter selon un mécanisme d'analyse sémantique contextuelle (et non simplement numérique), puis à exploiter la proximité de sens pour produire une recommandation adaptée. L'application doit donc être conçue comme un système expert utilisant le langage naturel et non comme un simple questionnaire à choix fixes.

Prérequis et exigences principales

- L'analyse doit être contextuelle et sémantique, non purement numérique.
- Les entrées utilisateur doivent être associées aux blocs de compétences à l'aide de techniques telles que :
    - Word embeddings (Word2Vec, GloVe, fastText)

  - Contextual embeddings (BERT, SBERT)
  - Mesures de similarité (cosine similarity, clustering sémantique).
- Un score de couverture sera calculé selon une formule combinant plusieurs blocs de compétences (pondérations, seuils ou règles d'agrégation).
- Le système doit générer un profil de compétences pour l'utilisateur et suggérer des métiers correspondants.

Ce projet s'appuie sur des embeddings contextualisés, tels que SBERT, pour transformer les textes en représentations vectorielles capables de préserver le sens. Ces vecteurs sont ensuite comparés selon une mesure de similarité sémantique (la similarité cosinus), ce qui permet au système de déterminer le niveau d'affinité entre les réponses de l'utilisateur et les éléments du référentiel. À partir de ce pourcentage de proximité, il doit ensuite produire une recommandation ordonnée et argumentée.

La conception du questionnaire doit reposer sur la collecte d'informations textuelles de manière hybride : l'utilisateur exprime ses préférences en texte libre, complète son profil grâce à une auto-déclaration progressive (par exemple via une échelle type Likert) et précise certains éléments via des questions guidées. Les réponses doivent être stockées dans un format structuré exploitable par le moteur sémantique.

L'application doit comporter un moteur NLP sémantique conçu autour de l'encodage des réponses utilisateur et des éléments du référentiel. Dans AISCA, ce référentiel est constitué de blocs de compétences et de profils métiers. Dans le cas de projets alternatifs, le référentiel peut concerner par exemple des films, des genres littéraires, des spécialités médicales ou des playlists musicales. L'important est que les données soient organisées sous la forme de catégories contenant des descriptions textuelles suffisamment riches pour permettre l'analyse sémantique.

Le système doit ensuite calculer un score global de proximité, qui correspond à un niveau de correspondance entre les informations fournies par l'utilisateur et les éléments du référentiel. La logique de scoring peut s'appuyer sur des pondérations, mais doit surtout permettre de classer des résultats selon une logique de pertinence et d'affinité contextuelle. Le système doit finalement proposer plusieurs recommandations, typiquement les trois plus pertinentes.

L'intégration d'une IA générative est obligatoire mais doit rester limitée à quelques usages précis. Elle peut notamment permettre d'enrichir les entrées utilisateur lorsque les phrases sont trop courtes, d'expliquer la recommandation, de synthétiser un profil, ou encore de produire une formulation plus professionnelle ou personnalisée. Dans tous les cas, cette intégration doit être réalisée avec parcimonie, en tenant compte des contraintes de coûts liées aux appels API et en respectant les mécanismes de caching. Une seule requête doit être effectuée pour chaque type de sortie générée.

La visualisation du résultat occupe enfin une place importante : l'étudiant doit proposer une représentation claire des résultats, permettant au public de comprendre la recommandation produite et la logique sur laquelle elle repose. Une interface Streamlit est fortement recommandée pour assurer une présentation claire, interactive et adaptée à un usage pédagogique.

Ce cahier des charges général s'applique intégralement à toutes les thématiques. Les sections suivantes décrivent uniquement les adaptations qui s'appliquent à chaque domaine choisi.

Résultats attendus

- Une interface web fonctionnelle du questionnaire.
- Un moteur d'analyse sémantique comparant les réponses avec votre référentiel.
- Un score de couverture des critères/compétences agrégé.
- Un module de recommandation pertinents.
- Une visualisation des résultats (ex. graphique radar, barres, cartes de similarité).

Exigences Fonctionnelles Détaillées (EF)

EF1 : Acquisition de la Donnée

- **EF1.1 : Questionnaire Hybride** : Le questionnaire doit combiner des questions numériques (échelle de Likert pour l'auto-déclaration de niveau, d'intérêt, de préférence, d'intensité…) et des questions ouvertes pour la collecte de preuves contextuelles, de besoins ou de descriptions selon la thématique (symptômes, goûts, expériences, styles…).
- **EF1.2 : Structuration** : Les réponses collectées doivent être stockées dans un format structuré (CSV, JSON, ou base de données locale) pour le traitement NLP.

EF2 : Moteur NLP Sémantique (Cœur du Projet - Coût Zéro)

- **EF2.1 : Référentiel de connaissances de base** : Vous devez créer un référentiel de connaissances en fonction de la thématique choisie (compétences, métiers, films, médecins, livres, playlists, etc.).
- **EF2.2 : Modélisation Sémantique** : Vous devez utiliser un modèle d'embeddings Open-Source et local (ex. SBERT via la librairie SentenceTransformer) pour transformer les entrées utilisateur et les phrases du référentiel en vecteurs.
- **EF2.3 : Mesure de Similarité** : Le moteur doit calculer la Similarité Cosinus entre les vecteurs utilisateur et les vecteurs de compétences ou critères du référentiel.

EF3 : Système de Scoring et Recommandation

- **EF3.1 : Formule de Score** : Vous devez implémenter une formule de score pondérée pour calculer le Score de Couverture Sémantique ou d'affinité sémantique selon la thématique.
- **EF3.2 : Recommandation** : Le système doit proposer les 3 Tops Propositions (profils métiers, spécialités médicales, films, livres, playlists, etc.) pour lesquels l'utilisateur obtient le score le plus élevé.

EF4 : Augmentation par GenAI (Stratégique et Limitée)

- **EF4.1 : Augmentation de l'Entrée (Pre-Processing)** : Vous pouvez développer une fonction qui utilise la GenAI pour enrichir les phrases d'entrée utilisateur jugées trop

courtes (moins de 5 mots), en ajoutant du contexte technique ou descriptif pour améliorer la précision des embeddings NLP locaux.

  - Exigence d'usage : Appel limité à un usage conditionnel (si la phrase est trop courte).

- **EF4.2 : Génération du Plan de Progression** : Le système doit générer un texte personnalisé (par GenAI) qui identifie les éléments prioritaires à développer ou les recommandations pertinentes (celles ayant les plus faibles scores de similarité), et propose un chemin d'évolution ou une recommandation détaillée selon la thématique.
  - Exigence d'usage : Un seul appel API pour la sortie finale.
- **EF4.3 : Synthèse de Proposition** : Générer une courte synthèse explicative utilisant un style approprié (ex. Executive Summary, justification de la recommandation, orientation ou mise en contexte), adaptée au domaine choisi.

Livrables :

• Vous devez soumettre les livrables sur MOODLE + Git/GitHub :
- Code source de la solution fonctionnelle + documentation (Note de groupe)
- Support de Présentation du projet
• Présentation et démonstration en classe lors de la dernière séance du module : Tous les membres du groupe doivent participer ; Evaluation/Note individuelle)
• Grille d'évaluation est partagée dans un document séparé.

Thématiques alternatives acceptées

Thématique : Recommandation cinématographique

Ce projet consiste à recommander des films correspondant au profil cinématographique exprimé par l'utilisateur. Celui-ci formule son souhait via une description libre, complétée par l'indication de préférences personnelles et de styles recherchés. L'application repose sur SBERT pour rapprocher la requête utilisateur de descriptions narratives présentes dans le référentiel, et proposer ainsi des films pertinents accompagnés d'une justification générée par IA.

Entrées utilisateur attendues

- texte libre décrivant le film souhaité, l'ambiance, le style, les émotions recherchées

- auto-déclaration de préférences par genre (Likert)
- questions guidées portant sur réalisateurs, période, mood

Référentiel attendu
Le référentiel doit contenir :
- genres cinématographiques
- styles
- moods
- thématiques
- synopsis
- mots-clés
- minimum 50 films

Structure possible :
| FilmID | BlockID | Catégorie | Film | Description narrative | Keywords |

Analyse sémantique et scoring
- embeddings SBERT
- similarité Cosinus
- calcul d'un score d'affinité
- classement des recommandations

Intégration GenAI
La GenAI sert ici à :
- enrichir les descriptions courtes,
- rédiger la justification de la recommandation,
- produire un court profil cinéphile.

Thématique : Orientation vers un médecin selon les symptômes décrits

Dans cette variante, l'utilisateur décrit librement ses symptômes. Le système doit orienter vers la spécialité médicale la plus adaptée, en s'appuyant sur la similarité sémantique avec des descriptions typiques. Il ne s'agit pas d'un diagnostic médical mais d'une orientation indicatrice.

Entrées utilisateur attendues
- description libre du symptôme
- intensité (Likert)
- durée
- localisation
- questions guidées

Référentiel attendu
Un référentiel médical doit inclure :
- spécialités
- symptômes associés
- organes concernés
- indications générales

- signaux d'alerte (red flags)
- au moins 40 symptômes + 15 spécialités

Exemple :
| MedID | BlockID | Spécialité | Symptômes associés | Indications | RedFlags |

Analyse sémantique et scoring
Le système repère les segments lexicaux associés aux pathologies courantes et calcule une proximité relative avec les descriptions du référentiel.

Intégration GenAI
L'IA générative permet de reformuler la demande, d'expliquer l'orientation et de produire une synthèse pédagogique, tout en rappelant explicitement qu'il ne s'agit pas d'un avis médical.

Thématique : Recommandation littéraire

Ce projet propose des ouvrages adaptés au style littéraire exprimé par l'utilisateur. L'analyse sémantique rapproche les descriptions libres et les résumés narratifs du référentiel afin de proposer des titres cohérents avec les thématiques, genres et styles recherchés.

Entrées utilisateur attendues
- texte libre
- auto-déclaration de préférences littéraires (ex. genre, ambiance, complexité)
- questions guidées selon époque, style, auteur favori

Référentiel attendu
Il doit contenir au minimum 60 ouvrages, comprenant :
- genre littéraire
- style narratif
- résumé
- thématique
- période
- mots-clés

Exemple structure :
| BookID | BlockID | Catégorie | Livre | Description | Keywords |

Analyse sémantique et scoring
L'application rapproche le texte libre des résumés d'ouvrages et calcule une similarité cosinus entre embeddings SBERT pour proposer un classement.

Intégration GenAI
La GenAI peut produire une synthèse mettant en contexte la recommandation, argumenter la correspondance ou proposer un commentaire personnalisé adapté au style littéraire.

Thématique : Recommandation Musicale

Cette thématique consiste à proposer des playlists ou titres musicaux correspondant aux goûts exprimés en langage naturel. L'utilisateur décrit un mood ou une ambiance, que le système analyse et rapproche des catégories musicales du référentiel afin d'extraire des playlists proches.

Entrées utilisateur attendues

- texte libre décrivant ambiance, mood
- préférences musicales (Likert)
- artistes ou genres connus

Référentiel attendu

Il doit comporter :

- ambiances
- genres
- playlists
- artistes
- moods
- au minimum 70 entrées

Structure possible :

| TrackID | BlockID | Catégorie | Titre | Keywords | Ambiance |

Analyse sémantique et scoring

SBERT associe les expressions émotionnelles de l'utilisateur aux caractéristiques du référentiel musical pour produire une recommandation hiérarchisée.

Intégration GenAI

La GenAI permet de rédiger une justification, de proposer un texte synthétique et éventuellement d'expliquer l'univers musical.

Conclusion

Toutes les thématiques alternatives doivent être conçues comme une adaptation directe du cahier des charges général AISCA. Le changement de domaine n'implique pas un changement du pipeline technique, mais uniquement une modification du référentiel et de la logique de recommandation.

GOOD LUCK!