

Cycle de Formation des Ingénieurs en Informatique

Rapport de Projet de Fin d'Études

Conception et réalisation d'une application web de recommandation intelligente de contenu pour le réseau social Pepolls

Elaboré Par
DJOBBI Fahmi

Encadré Par
Mme. TBARKI Khaoula
M. BEN RHOUMA Amine

En collaboration avec



Encadrant professionnel, **M. BEN RHOUMA Amine**

Signature et cachet

Encadrant académique, **Mme. TBARKI Khaoula**

Signature

DÉDICACE

À ma mère Fadhila, rayonnante lumière qui a embrasé mon existence de sa flamme, infusant espoir dans les méandres de mon âme. À chaque moment d'abattement, la clarté de ta présence, chère maman, réanime mon cœur avec ardeur.

À mon père Jalel, pilier inébranlable qui m'a émerveillé par ses sacrifices, guide qui m'a enseigné l'art de vivre en homme véritable. Tu es mon modèle inégalé, source constante de fierté.

À mes chers frères et sœurs, vous qui êtes mon soutien infaillible, colonnes solides qui ne flétrissent point. Par votre présence, ma détermination se fortifie et ma volonté s'enracine avec puissance.

À Layan et Joud, doux éclats de rire, à vous que j'aime sans réserve. Votre sourire irradie en moi une énergie renouvelée, et votre optimisme nourrit mon être.

À Hakim, Pour l'âme généreuse qui a toujours été à mes côtés, comme une boussole dans mes moments d'égarement. Plus qu'un frère, tu as été un phare dans la tempête.

À mes amis et ma famille, gardiens bienveillants en dépit de la noirceur du monde, témoins de sincérité face aux épreuves de la vie. Vous que nous avons aimés avec une profonde authenticité, et qui demeurez fidèles à nos liens indéfectibles.

À celle qui un jour me murmura : 'Je crois en toi, ne cesse jamais de rêver', tes paroles restent gravées à jamais, intemporelles et ineffaçables.

Que ce projet soit dédié à chacun de vous, unis dans une symbiose éternelle.

Fahmi DJOBBI

REMERCIEMENT

Tout d'abord, nous tenons à exprimer notre profonde gratitude envers **Allah** pour nous avoir accordé la force et la détermination nécessaires pour mener à bien ce travail.

Avant de présenter notre travail, nous souhaitons exprimer notre sincère reconnaissance envers toutes les personnes qui ont contribué à la réalisation de ce projet.

En particulier, nous tenons à remercier chaleureusement **Monsieur Amine Bn Rhouma**, notre encadrant professionnel, pour sa disponibilité, ses encouragements et sa patience tout au long de notre stage. Sa précieuse expertise et ses conseils avisés ont grandement contribué à l'aboutissement de ce projet.

Nous tenons également à exprimer notre profonde gratitude envers **Madame Khaoula Tbarki**, notre encadrante académique, pour ses conseils éclairés, son dévouement et son soutien tout au long de la réalisation de notre projet ainsi que dans la rédaction de ce rapport de fin d'études.

Nos remerciements vont également à toute l'équipe de Pepolls, notamment à **Monsieur Jihed Rahmouni** et **Madame Jihen Rahmouni**, pour leur accueil chaleureux et leurs encouragements tout au long de notre collaboration.

Nous exprimons également notre reconnaissance envers **Monsieur Ahmed Khelifi** pour son aide précieuse et son soutien indéfectible qui ont joué un rôle essentiel dans notre acquisition d'une expérience professionnelle enrichissante.

Enfin, nous souhaitons exprimer notre gratitude envers tous nos professeurs académiques pour leurs efforts inlassables tout au long de notre parcours d'études à l'Université **Tek-up**.

Nous espérons que ce projet sera à la hauteur de vos attentes.

Résumé

Dans le cadre de notre projet de fin d'études, réalisé en vue de l'obtention du diplôme national d'ingénieur, nous avons collaboré avec l'entreprise « Pepolls » pour développer une application web de recommandation de contenu intelligent pour leur réseau social. Notre objectif était de fournir aux utilisateurs des suggestions pertinentes en se basant sur leurs interactions passées avec la plateforme. Pour cela, nous avons adopté une approche centrée sur l'utilisation de modèles avancés de traitement du langage naturel (NLP). En utilisant les technologies Flask et ReactJS, nous avons créé une plateforme de médias sociaux qui nous a permis de déployer ces modèles de manière efficace et optimisée.

Mots clés : Traitement du Langage Naturel, ReactJS, MongoDB, Recommandation, Deep Learning, Intelligence Artificielle, Réseau Social.

Abstract

As part of our final year project, conducted to obtain the national engineering degree, we collaborated with « Pepolls » company to develop an intelligent content recommendation web application for their social network. Our objective was to provide users with relevant suggestions based on their past interactions with the platform. To achieve this, we adopted an approach centered around the utilization of advanced natural language processing (NLP) models. By leveraging Flask and ReactJS technologies, we built a social media platform that facilitated the efficient and optimized deployment of these models.

Keywords : Natural Language Processing, ReactJs, MongoDB, Recommendation, Deep Learning, Artificial Intelligence, Social Network.

TABLE DES MATIÈRES

Liste des Figures	xiii
Liste des tableaux	xvii
Liste des acronymes	xviii
Introduction générale	1
Chapitre 1: Contexte général du projet	3
Introduction	4
1.1 Présentation de l'organisme d'accueil	4
1.1.1 Présentation générale	4
1.1.2 Organigramme de l'entreprise	5
1.1.3 Objectifs de l'entreprise	5
1.2 Présentation du projet	6
1.2.1 Contexte générale	6
1.2.2 Motivation et Problématique	7
1.2.3 Solution proposée	8
1.3 Méthodologie et approche de développement	8
1.3.1 Méthodologies traditionnelles vs agiles	8
1.3.2 Gestion du projet	9
1.3.2.1 Aperçu de Scrum	10
1.3.2.2 Rôles dans Scrum	10
1.3.2.3 Principes de Scrum	11

1.4	Langage de modélisation	12
	Conclusion	12
Chapitre 2:	Système de recommandation NLP proposé	13
	Introduction	14
2.1	Les notions de base sur les systèmes de recommandation	14
2.1.1	Les systèmes de recommandations	14
2.1.2	Traitement du Langage Naturel NLP	14
2.1.3	Les systèmes de recommandations NLP	14
2.1.4	Historique des systèmes de recommandations NLP	15
2.1.5	Avantages des systèmes de recommandation "NLP"	16
2.1.6	Types des systèmes de recommandation NLP	16
2.1.7	Différentes applications des systèmes de recommandation NLP	17
2.2	Le fonctionnement du système de recommandation NLP proposé	19
2.3	Système de recommandation basée sur Deep Learning	20
2.3.1	Prétraitement des données	20
2.3.1.1	Tokenisation	20
2.3.1.2	Padding de séquences	20
2.3.1.3	L'approche Id2Word	21
2.3.2	Vectorisation des données (Word Embedding)	21
2.3.2.1	L'approche Word2Vec	21
2.3.2.2	L'approche TF-IDF	23
2.3.2.3	L'approche One Hot Encoding	24
2.3.3	Modèles de deep learning pour la recommandation basée sur NLP	25
2.3.3.1	Convolutional Neural Networks (CNN)	25
2.3.3.2	Long Short-Term Memory (LSTM)	27
2.3.3.3	Bidirectional Long Short-Term Memory (BLSTM)	29
2.3.3.4	Gated Recurrent Unit (GRU)	30
2.3.4	Protocoles expérimentaux d'évaluer et améliorer les performances des modèles	31
2.3.4.1	Métriques d'évaluation	31
2.3.4.2	Overfitting	33
2.3.4.3	Solutions pour prévenir l'overfitting	33
	Conclusion	34

Chapitre 3: Conception et architecture du projet	35
Introduction	36
3.1 Spécification des besoins	36
3.1.1 Identification des acteurs	36
3.1.2 Besoins fonctionnels	36
3.1.3 Besoins non fonctionnels	37
3.2 Étude analytique	38
3.2.1 Diagramme de cas d'utilisation général	38
3.2.2 Diagramme de classe global	39
3.3 Pilotage du projet avec Scrum	39
3.3.1 Le Backlog du produit	39
3.3.2 Planification des releases	41
3.4 Architecture de la solution	42
3.4.1 Développement Full stack	42
3.4.2 Couche et niveau	43
3.4.3 Architecture logique du projet	43
3.4.4 Diagramme de paquetages global	44
3.4.5 Architecture physique du projet	45
3.5 Environnement de travail	46
3.5.1 Environnement matériel	46
3.5.2 Environnement logiciel	46
3.5.2.1 Éditeur de code	46
3.5.2.2 Client API	47
3.5.2.3 Open API	48
3.5.2.4 Frameworks et Technologies de développement	48
3.5.2.5 Langages de programmation	49
3.5.2.6 Base de donnée	50
3.5.2.7 Outils DevOps	51
3.5.2.8 Outils de modélisation	51
Conclusion	52
Chapitre 4: Release 1	53
Introduction	54
4.1 Développement de sprint 1	54

4.1.1	Analyse et conception	55
4.1.1.1	Diagramme de cas d'utilisation du sprint 1	55
4.1.1.2	Diagrammes de séquence	55
4.1.2	Réalisation	58
4.1.2.1	Interface d'inscription	58
4.1.2.2	Interface de l'authentification	59
4.1.2.3	Interfaces de réinitialisation de mots de passe	59
4.2	Développement du sprint 2	60
4.2.1	Analyse et conception	61
4.2.1.1	Diagrammes de cas d'utilisation du sprint 2	61
4.2.1.2	Diagrammes de séquence	63
4.2.2	Réalisation	68
4.2.2.1	Interface d'édition de profil	68
4.2.2.2	Interface de messagerie "Talk"	69
4.2.2.3	Interfaces d'ajout de publication	69
4.2.2.4	Interface d'ajout de commentaire	70
4.2.2.5	Interface de suppression de publication	71
4.2.2.6	Interfaces du flux d'actualités	71
4.2.2.7	Interface de partage de publication	72
	Conclusion	73
Chapitre 5:	Release 2	74
	Introduction	75
5.1	Choix méthodologique	75
5.1.1	Compréhension du Business	75
5.1.2	Compréhension des Données	76
5.1.3	Préparation des Données	76
5.1.4	Modélisation	76
5.1.5	Évaluation	76
5.1.6	Déploiement	77
5.2	Élaboration du sprint 3	77
5.2.1	Business comprendre	77
5.2.2	Compréhension des Données	78
5.2.2.1	Analyse des données réelles de l'application Pepolls	78
5.2.2.2	Extraction de données (Scraping)	79

5.2.2.3	Exploration des données	80
5.2.3	Préparation des Données	83
5.2.3.1	Nettoyage de texte	83
5.2.3.2	L'élimination des valeurs manquantes et indéfinies	84
5.2.3.3	Dévision des données	84
5.2.3.4	Lemmatisation	85
5.2.3.5	ID2Word,Tokenization et Padding des séquences	85
5.2.3.6	Visualisation avec WordCloud	86
5.3	Élaboration du sprint 4	87
5.3.1	Modélisation	87
5.3.2	Évaluation	89
5.3.2.1	Score de cohérence et perplexité de modèle LDA	89
5.3.2.2	Étude comparative des résultats de classification des sujets .	90
5.3.2.3	Analyse des domaines à améliorer et identification des pro- chaines étapes	92
5.3.3	Déploiement du système de recommandation	92
5.3.3.1	Analyse et conception	92
5.3.3.2	Processus de recommandation en back-end	93
5.3.3.3	Processus de recommandation en front-end	94
	Conclusion	95
	Conclusion générale	96
	Références	97

LISTE DES FIGURES

1.1	Logo de Pepolls	4
1.2	Fiche d'information de Pepolls	5
1.3	Organigramme du département technologie	5
1.4	Objectifs de Pepolls	6
1.5	Cycle itératif agile vs Cycle séquentiel traditionnel	9
1.6	Cycle Scrum pour un sprint	10
1.7	Les 6 principes de Scrum	12
2.1	Types des systèmes de recommandation NLP, (a) Filtrage Collaboratif, (b) Filtrage basé sur le contenu	17
2.2	Principe de Tokenizer, (a) avant Tokenizer, (b) après Tokenizer	20
2.3	Principe de pad sequences, (a) avant pad sequence, (b) après pad sequence . .	21
2.4	Similarité des mots	22
2.5	Skip-gram vs CBOW	22
2.6	La conversion du TF-IDF , (a) avant TF-IDF (b) après TF-IDF	23
2.7	Exemple de One Hot Encoding	24
2.8	L'architecture du modèle CNN dans le domaine NLP	25
2.9	L'architecture du modèle CNN de classification de texte	26
2.10	L'architecture du modèle RNN	27
2.11	La structure interne de RNN	28
2.12	L'architecture de LSTM	28
2.13	La structure interne de BLSTM	29
2.14	L'architecture de GRU	30

2.15 la matrice de confusion	32
2.16 Overfitting dans le Deep Learning	33
3.1 Diagramme de cas d'utilisation général	38
3.2 Diagramme de classe global	39
3.3 Planification des releases	42
3.4 Développement Full stack	43
3.5 Diagramme de Package Global	44
3.6 Aperçu de l'architecture physique de la solution	46
3.7 Logo VsCode	47
3.8 Logo Google Colab	47
3.9 Logo Thunder Client	47
3.10 Logo Swagger	48
3.11 Logo ReactJs	48
3.12 Logo Flask	48
3.13 Logo Bootstrap	49
3.14 Logo Pusher	49
3.15 Logo TensorFlow	49
3.16 Logo Python	50
3.17 Logo JavaScript	50
3.18 Logo MongoDB	50
3.19 Logo Git	51
3.20 Logo GitLab	51
3.21 Logo ClickUp	51
3.22 Logo Draw.io	52
4.1 Diagramme de cas d'utilisation du sprint 1	55
4.2 Diagramme de séquence du cas d'utilisation «S'inscrire»	56
4.3 Diagramme de séquence du cas d'utilisation «S'authentifier»	57
4.4 Diagramme de séquence du cas d'utilisation «Réinitialiser de mot de passe»	57
4.5 L'interface initiale d'inscription	58
4.6 La deuxième interface d'inscription	58
4.7 Interface de l'authentification	59
4.8 Interface de réinitialisation de mot de passe	59
4.9 Interface de vérification code	60

4.10	Diagramme de cas d'utilisation du Gestion profil et messages	61
4.11	Diagramme de cas d'utilisation du Gestion publications et commentaires . . .	62
4.12	Diagramme de séquence du cas d'utilisation «Consulter Profil»	63
4.13	Diagramme de séquence du cas d'utilisation «Modifier Profil»	64
4.14	Diagramme de séquence du cas d'utilisation «Gérer messages»	64
4.15	Diagramme de séquence du cas d'utilisation «Créer publication»	65
4.16	Diagramme de séquence du cas d'utilisation «Explorer publication»	66
4.17	Diagramme de séquence du cas d'utilisation «Partager publication»	66
4.18	Diagramme de séquence du cas d'utilisation «Gérer commentaires»	67
4.19	Diagramme de séquence du cas d'utilisation «Réagir sur publication»	68
4.20	Interface d'édition de profil	69
4.21	Interface d'édition de profil	69
4.22	Interface d'ajout de publication	70
4.23	Interface d'ajout de publication	70
4.24	Interface d'ajout de commentaire	71
4.25	Interface de suppression de publication	71
4.26	Interfaces du flux d'actualités	72
4.27	Interfaces du flux d'actualités	72
4.28	Interface de partage de publication	72
5.1	Schéma de la méthodologie CRISP	75
5.2	Graphique de polls par type d'activités	78
5.3	Graphique de polls par topics	79
5.4	Scraping data à partir de Reddit	79
5.5	Les données extraites	80
5.6	Histogramme de subreddit	81
5.7	Diagramme circulaire du pourcentage de subreddit	81
5.8	Une visualisation en Treemap et Heatmap	82
5.9	Un scatterplot pour la fréquence de subreddit	82
5.10	Informations sur notre données	83
5.11	Nettoyage de texte, (a) avant nettoyage, (b) après nettoyage	84
5.12	l'élimination des valeurs manquantes et indéfinies	84
5.13	Taille de données de train et de test	85
5.14	Lemmatisation	85

5.15 ID2Word,Tokenization et Padding des séquences	85
5.16 les mots les plus fréquemment utilisés dans notre Data	86
5.17 les mots les plus fréquemment utilisés dans chaque sujet	86
5.18 Réduction de dimensions avec LDA	87
5.19 Un aperçu des paramètres utilisés dans chaque modèle	88
5.20 Score de cohérence et perplexité de modéle LDA	89
5.21 Comparaison de la précision, du rappel et du f1score entre GRU et DescionTree	91
5.22 Comparaison de la matrice de confusion entre GRU et DescionTree	91
5.23 Test de modéle	92
5.24 Diagramme de cas d'utilisation de recommandation	93
5.25 Processus de recommandation en back-end	93
5.26 API Recommandation	94
5.27 Déploiement du système de recommandation	95

LISTE DES TABLEAUX

3.1	Backlog de produit	40
3.2	Caractéristiques du Portatif	46
4.1	Backlog du premier sprint	54
4.2	Backlog du deuxième Sprint	60
5.1	Description des colonnes de nos données	80
5.2	Comparaison de l'exactitude (accuracy) des 5 modèles	90

LISTE DES ACRONYMES

BLSTM = Bidirectional Long Short-Term Memory (29)

CNN = Convolutional Neural Networks (25)

CRISP-DM = Cross-Industry Standard Process for Data Mining (75)

CSV = Comma-Separated Values (79)

DL = Deep Learning (33)

DS = Data Science (75)

GRU = Gated Recurrent Unit (30)

IA = Intelligence Artificielle (5, 6)

JSON = JavaScript Object Notation (94)

LDA = Latent Dirichlet Allocation (87)

LSTM = Long Short-Term Memory (27)

ML = Machine Learning (33)

NLP = Traitement du Langage Naturel (8, 14, 19)

RNN = Recurrent Neural Network (27)

UML = Langage de Modélisation Unifié (12)

INTRODUCTION GÉNÉRALE

Dans ce monde en constante évolution, l'intelligence artificielle (IA) apparaît comme une force indéniable qui façonne notre réalité. Au fil des décennies, l'IA s'est répandue dans tous les domaines, influençant nos interactions, nos décisions et même notre perception de la réalité de manière inédite.

Parmi les secteurs touchés, les réseaux sociaux se distinguent par leur prolifération massive et désordonnée de données, habilement exploitées par les géants de la technologie à travers le déploiement de systèmes de recommandation basés sur l'IA, dans le but de manipuler l'opinion publique à leur avantage. Face à cette réalité, il devient essentiel de bien réfléchir à l'éthique et à la responsabilité qui entourent la science des données. Ainsi, la mise en place de systèmes de recommandation régulés de manière pacifique et transparente est primordiale, tout en offrant aux utilisateurs des suggestions personnalisées basées sur leurs préférences et leur comportement passé, en respectant leurs droits et en évitant les abus.

Dans ce contexte, notre projet est réalisé au sein de l'équipe d'intelligence artificielle de Pepolls, dans le cadre de notre projet de fin d'études en ingénierie informatique, spécialisé en Data Science. Notre objectif principal est d'automatiser le processus de recommandation sur le réseau social Pepolls, afin de fournir aux utilisateurs des suggestions pertinentes, basées de manière transparente et éthique sur leurs interactions passées avec la plateforme. Notre approche est centrée sur l'utilisation de modèles avancés de traitement du langage naturel (NLP). Nous sommes convaincus que l'utilisation de ces modèles permettra aux utilisateurs de découvrir de nouvelles idées, de s'immerger dans de nouveaux domaines d'intérêt et d'interagir de manière enrichissante avec la communauté.

Dans le même temps, notre mécanisme de recommandation offre des avantages significatifs aux annonceurs, en leur fournissant des moyens plus efficaces d'atteindre leur public cible et d'accroître leur visibilité.

Ce présent rapport s'articule autour de cinq chapitres. Le premier chapitre introduit l'organisme d'accueil, présente en détail le sujet et expose les méthodologies et la modélisation que nous avons adoptées. Dans le deuxième chapitre, nous examinons en profondeur les systèmes de recommandation NLP, en expliquant le fonctionnement de notre système de recommandation et en discutant de l'approche basée sur l'apprentissage profond. Le troisième chapitre fournit une analyse complète, spécifiant les besoins fonctionnels et non fonctionnels, la gestion et la planification, l'architecture de la solution et l'environnement de travail. Les quatrième et cinquième chapitres fournissent une analyse détaillée des différentes Releases. Enfin, nous clôturons ce rapport par une conclusion générale et abordons quelques perspectives d'avenir.

CHAPITRE 1

CONTEXTE GÉNÉRAL DU PROJET

Introduction

Le travail de mon projet de fin d'études est réalisé au sein de l'entreprise Pepolls dans le département Recherche & Développement (R&D) . Notre projet vise à proposer une solution web de recommandation intelligente de publications dans un réseau social.

Comme toute initiation commence par une phase de découverte, ce chapitre vise à situer notre projet dans son contexte général. Nous commençons tout d'abord par la présentation de l'organisme d'accueil. Ensuite, nous décrivons brièvement le contexte général du projet. Puis, nous exposons la problématique et la solution proposée. Enfin, nous définissons la méthode de travail et le langage de modélisation adopté.

1.1 Présentation de l'organisme d'accueil

Cette section présente l'organisme d'accueil Pepolls, ses objectifs ainsi que leur organigramme.

1.1.1 Présentation générale

Pepolls est une startup tunisienne où leur logo est présenté dans la figure 1.1, qui utilise la technologie de la blockchain pour révolutionner le monde numérique, en particulier les réseaux sociaux. Son objectif est de créer une nouvelle définition de la présence digitale en connectant les utilisateurs dans un espace proactif, décentralisé et auto-souverain. Pour cela, Pepolls a créé un réseau social basé sur des sondages qui fusionne les styles de vie, les intérêts et les revenus des utilisateurs. Ce réseau social valorise l'opinion publique et le débat en offrant un espace de participation et de collaboration globales. Les utilisateurs peuvent publier des sondages et gagner des récompenses en crypto-monnaie pour leur participation. L'objectif de Pepolls est de lancer des solutions centrées sur l'humain qui équilibrivent le progrès social et économique et la résolution des problèmes humains dans un système qui intègre fortement l'espace cyberspatial et physique[1].



FIGURE 1.1 – Logo de Pepolls

La figure 1.2 détaille les informations nécessaires de l'entreprise d'accueil Pepolls.

Nom de l'entreprise :	Pepolls
Directeur Général :	Mr Jihed RAHMOUNI
Forme juridique:	Startup
Adresse :	Rue du Lac Tibériade, Immeuble Lac Center, Bloc C - Etage 1 Les Berges du Lac, 1053
Téléphone :	+216 23 680 343

FIGURE 1.2 – Fiche d'information de Pepolls

1.1.2 Organigramme de l'entreprise

La figure 1.3 présente l'organigramme de toutes les équipes rattachées à la direction recherche et développement technologique.

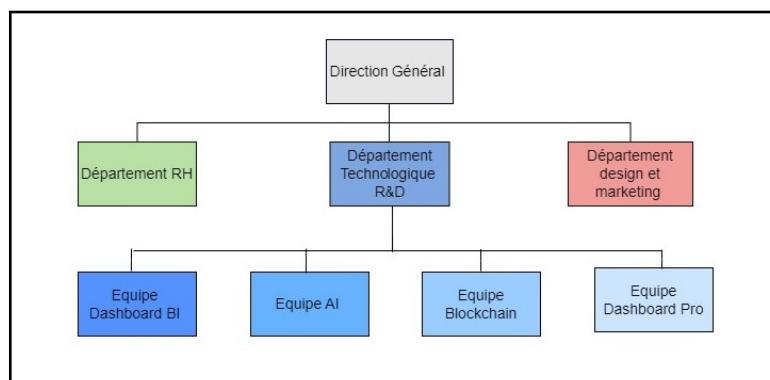


FIGURE 1.3 – Organigramme du département technologie

Notre projet est réalisé au sein du département technologique R&D et plus précisément dans l'équipe Intelligence Artificielle (IA).

1.1.3 Objectifs de l'entreprise

L'objectif de Pepolls est de réaliser le potentiel de l'humanité et de bâtir un monde où la coopération et la résolution collective des problèmes sont prioritaires. Pepolls est prêt à collaborer avec toutes les entités et personnes pour atteindre les objectifs de développement durable qui sont, comme montre la figure 1.4 .

- **Mettre fin à la pauvreté**
- **Santé et bien-être numériques**
- **Partager les connaissances**

- L'équité numérique
- Faites entendre la voix
- Travaillent tous ensemble



FIGURE 1.4 – Objectifs de Pepolls

1.2 Présentation du projet

Cette partie est composée de trois sous-sections. Dans la première sous-section, nous expliquons le contexte et l'objectif du projet. Ensuite, nous détaillons notre problématique. Dans la dernière sous-section, nous présentons notre solution proposée.

1.2.1 Contexte générale

Pepolls est une plateforme de médias sociaux qui permet aux utilisateurs de s'exprimer librement et les récompense avec des Pecoins pour monétiser les sondages et acheter des offres. Puisque leurs intérêts sont une priorité, Pepolls prévoit d'utiliser la technologie de l'IA pour améliorer encore l'expérience de l'utilisateur dans l'application.

Notre projet s'articule autour de ces deux objectifs fondamentaux :

- Réaliser un système de recommandation de contenu basé sur l'intelligence artificielle pour aider les utilisateurs à découvrir des contenus pertinents et intéressants en fonction de leurs préférences et de leur historique de navigation.
- Développer une plateforme web de média social pour incorporer les modèles d'Intelligence Artificielle (IA).

1.2.2 Motivation et Problématique

Le fil d'actualités d'une plateforme de médias sociaux fait référence au flux de contenu qu'un utilisateur voit lorsqu'il ouvre l'application ou le site web. Le fil d'actualités se compose généralement de publications, de photos, de vidéos et d'autres types de contenus partagés par les connexions de l'utilisateur ou les comptes qu'il suit sur la plateforme.

L'expérience de l'utilisateur dans le fil d'actualités peut varier en fonction de la manière dont la plateforme conçoit et présente le contenu. Lorsqu'il s'agit d'un fil d'actualités statique, le contenu est affiché dans un ordre prédéterminé, souvent en fonction de l'heure à laquelle le contenu a été publié ou de la source du contenu. Les utilisateurs peuvent ainsi voir des contenus qui ne sont pas pertinents ou intéressants pour eux, ce qui peut rendre leur expérience moins attrayante. En outre, l'absence de recommandations publicitaires personnalisées limite l'engagement des utilisateurs et l'efficacité des dépenses publicitaires, ce qui risque d'entraîner des investissements publicitaires inutiles et des pertes financières pour les annonceurs.

Si le fil d'actualités d'une plateforme de médias sociaux est statique et ne repose pas sur l'IA, cela peut entraîner plusieurs problèmes :

- **Une personnalisation limitée :** Les flux de médias sociaux sont limités dans leur capacité à personnaliser le contenu pour les utilisateurs individuels. Les utilisateurs risquent donc de voir des contenus non pertinents ou inintéressants, ce qui peut entraîner une baisse de l'engagement et de la satisfaction à l'égard de la plateforme.
- **Une utilisation inefficace des ressources :** Les flux statiques peuvent se traduire par un contenu moins pertinent pour les utilisateurs, ce qui peut entraîner un gaspillage de ressources et un retour sur investissement plus faible pour les annonceurs qui utiliseront les flux à des fins publicitaires, ainsi qu'une baisse des revenus pour la plateforme.
- **Une difficulté à suivre le volume de contenu :** En raison du grand volume de contenu publié sur les médias sociaux, les fils d'actualité peuvent rapidement devenir saturés pour les utilisateurs. Cette saturation peut entraîner une surcharge d'informations, qui peuvent alors se sentir dépassés et désengagés de la plateforme.

1.2.3 Solution proposée

La solution proposée pour résoudre les problèmes liés à l'engagement des utilisateurs, à la découverte de contenu et à l'efficacité de la publicité sur une plateforme de médias sociaux consiste à mettre en œuvre un système de recommandation basé sur l'IA avec des modèles Traitement du Langage Naturel (NLP).

En analysant le comportement des utilisateurs sur la plateforme, le système de recommandation basé sur l'IA peut identifier les contenus susceptibles de les intéresser, augmentant ainsi leur engagement et leur satisfaction globale. Cela peut aider les utilisateurs à découvrir de nouveaux contenus et profils à suivre, tout en offrant aux annonceurs des moyens plus efficaces d'atteindre leur audience cible.

Pour tester l'efficacité de cette solution, il est nécessaire de développer une plateforme d'essai capable de collecter des données d'utilisation, telles que l'historique de navigation et les interactions de l'utilisateur avec le contenu.

Globalement, la mise en œuvre d'un système de recommandation basé sur l'IA avec des modèles NLP peut constituer une solution prometteuse pour améliorer l'expérience des utilisateurs et stimuler l'engagement sur les plateformes de médias sociaux, tout en offrant aux annonceurs des moyens plus efficaces d'atteindre leur public cible.

1.3 Méthodologie et approche de développement

Il existe des nombreuses méthodologies de gestion de projets où leur choix se fait suivant la nature du projet lui-même. Toutefois, dans un environnement changeable et évolutif, les méthodologies traditionnelles comme le Waterfall ou le cycle en V ne présentent plus le résultat souhaité du point de vue du temps et de qualité, ce qui a poussé les entreprises à changer leurs méthodes de gestion. De ce fait, d'autres méthodologies plus adaptées qui se basent sur l'esprit agile les ont remplacées sur le marché.

1.3.1 Méthodologies traditionnelles vs agiles

La gestion de projet traditionnelle est linéaire. Elle mène un cycle séquentiel : initiation, planification, implémentation, vérification, acceptance. Avant le commencement du développement, elle nécessite une planification complète, y compris les tâches, le budget et le temps nécessaire. Généralement, le client n'est présent avec l'équipe de développement qu'au début lors de l'explication des besoins et à la fin après l'achèvement du développement. Par

conséquent, c'est la méthodologie à suivre quand il s'agit d'un projet prévisible et de complexité faible ayant des objectifs très clairs au préalable.

La gestion de projet agile est itérative. C'est l'approche à suivre lorsqu'il s'agit d'un projet imprévisible et versatile. Le développement du projet est divisé en itérations de petites durées telles que chaque itération commence avec une planification et finit avec un incrément livrable. Elle se base sur le travail d'équipe, la flexibilité et l'adaptation aux changements au fur et à mesure du développement. Le client est présent tout au long du développement du projet et il collabore avec l'équipe pour la création d'incréments satisfaisants grâce à ses feedback fréquents. Comme exemples d'approches agiles : Kanban, Scrumban, SAFe et Scrum. Donc la distinction entre la gestion d'un projet selon chaque méthodologie peut être illustrée dans la figure 1.5 suivante.

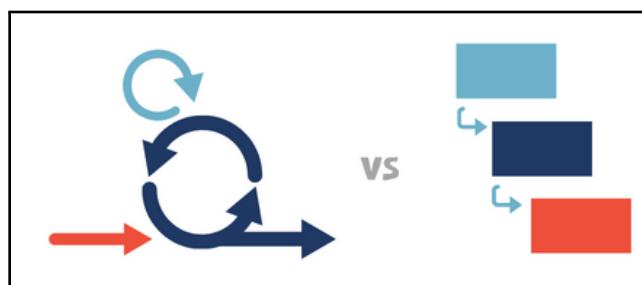


FIGURE 1.5 – Cycle itératif agile vs Cycle séquentiel traditionnel

1.3.2 Gestion du projet

Au sein de l'équipe IA à Pepolls, la méthodologie adéquate choisie depuis le début de la startup est Scrum et nous avons décidé de continuer avec cette méthodologie. Cette décision est due à plusieurs raisons :

1. Le besoin n'est pas complètement fixé dès le début. Effectivement, la régie prépare ses demandes au fur et à mesure du développement de l'application. De ce fait, Scrum est adéquat car il est un framework agile dont la phase de planification se fait graduellement et parallèlement au développement du produit.
2. Le changement des besoins est fréquent. En effet, Scrum encourage l'adaptation au changement via un ensemble de processus et de principes riches en retour d'informations de la part des parties prenantes.
3. Les contraintes du temps, coût et qualité sont majeures. Comme solution, Scrum fournit des processus et des outils détaillés qui servent à gérer ces contraintes.

1.3.2.1 Aperçu de Scrum

Un cycle Scrum commence par une réunion des parties prenantes pour définir la vision du projet. Ensuite, le Product Owner crée le Product Backlog contenant une liste priorisée non finalisée des besoins du client, appelées User Stories, qui peut être affinée tout au long du projet [2].

Le développement du produit se fait en itérations appelées Sprints qui, chacun, dure en général entre 1 et 6 semaines. Un sprint commence par une réunion de planification appelée Sprint Planning, où l'équipe crée ensemble un Sprint Backlog à partir du Product Backlog contenant les User Stories à faire au sein du sprint. Il finit par une réunion de Revue, appelée Sprint Review, et une autre de rétrospection, appelée Sprint Retrospective, pour avoir le feedback des parties prenantes sur l'incrément livrable issu du Sprint et pour améliorer le travail lors des prochains sprints.

Durant le sprint, l'équipe core Scrum se réunit pendant 15 minutes quotidiennement dans une réunion nommée Daily Standup pour parler de leurs objectifs pour ce jour-là et pour trouver des solutions à leurs obstacles s'il y en a.

La fin du projet marque la fin du cycle Scrum avec une réunion rétrospective pour le projet au complet. Ce cycle Scrum pour un sprint est illustré dans la figure 1.6 ci-dessous.

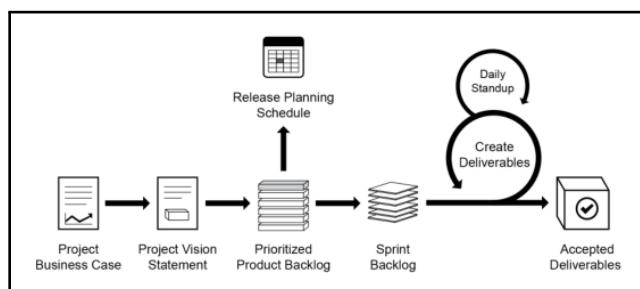


FIGURE 1.6 – Cycle Scrum pour un sprint

1.3.2.2 Rôles dans Scrum

⇒ Il y a deux types de rôles dans Scrum : les rôles core et les rôles non core.

⇒ Les rôles core sont les rôles indispensables à l'application du framework Scrum. Ils forment la Scrum Core Team et ils sont :

- **Scrum master** : Il guide l'équipe sur l'application des pratiques scrum, facilite la communication entre eux et protège l'environnement de travail de l'équipe de développement.
- **Product owner** : Il représente la voix du client. C'est son vis à vis avec l'équipe Scrum . Il comprend le besoin du client et l'explique à l'équipe de développement.

- **Scrum team :** C'est l'ensemble des développeurs qui travaillent sur les incrémentés d'un produit. Ils ont du savoir sur plusieurs domaines et chacun en maîtrise au moins un.

1.3.2.3 Principes de Scrum

Scrum est un framework riche en processus et en pratiques qui, en les appliquant correctement, promettent de la valeur commerciale pour l'organisation. En effet, Scrum fournit 19 processus, 5 aspects et 6 principes. Ces derniers sont le cœur de Scrum, ils sont invariants et inchangables contrairement aux processus et aux aspects qui peuvent être ajustés aux besoins du projet.

Voici les principes de Scrum :

- **Empirical process control ou processus Empirique :** Au lieu de se baser sur une planification à l'avant et un développement prédictif, Scrum se base sur l'observation et l'expérimentation puis l'adaptation.
- **Self-organization ou Auto-organisation :** Scrum encourage les membres de l'équipe de développement à être indépendant et à s'auto organiser au lieu de travailler sous les ordres et les commandes.
- **Collaboration :** L'équipe travaille ensemble avec une responsabilité partagée entre eux pour créer le maximum de valeur ajoutée.
- **Value based prioritization ou priorisation basée sur la valeur :** Scrum se concentre sur la valeur lors de la priorisation des tâches. Les parties qui ont le plus de valeur Business sont traitées avant.
- **Time-boxing ou délimité dans le temps :** Le temps est une contrainte dans le développement. De ce fait, Scrum délimite plusieurs pratiques par le temps. Par exemple, les Sprints ou les réunions quotidiennes Daily Standup.
- **Iterative development ou développement itératif :** Scrum part du principe que le besoin est toujours changeant. Par conséquent, Scrum encourage le développement en itérations de livrables pour pouvoir s'adapter rapidement aux changements.

Ces principes sont illustrés à la figure 1.7 .

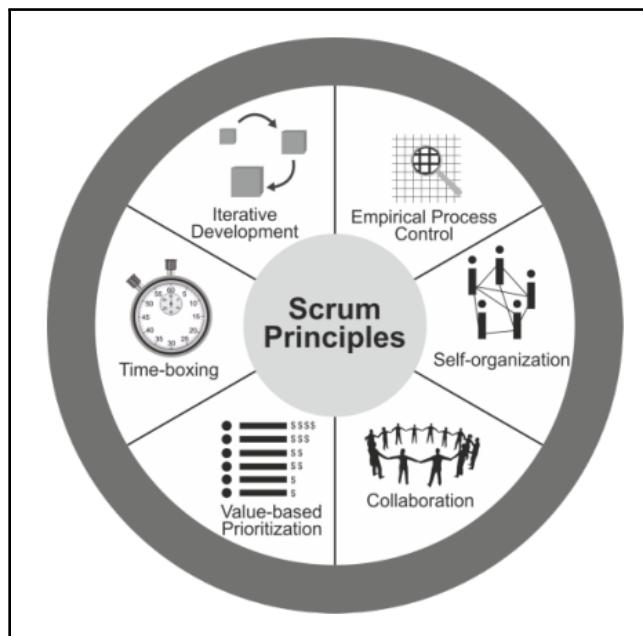


FIGURE 1.7 – Les 6 principes de Scrum

1.4 Langage de modélisation

Nous avons opté pour l'utilisation de l'Langage de Modélisation Unifié (UML) comme langage de modélisation pour faciliter notre travail. L'UML est un langage graphique de modélisation qui est principalement utilisé pour la conception, l'architecture et la mise en place de systèmes logiciels complexes. Il permet de représenter la structure et le comportement d'un système de manière visuelle, ce qui facilite la compréhension et la communication entre les membres de l'équipe [3] .

Conclusion

Durant ce chapitre, nous avons présenté l'organisme d'accueil où nous avons réalisé notre stage de fin d'études. En outre, nous avons défini brièvement la problématique et la solution adéquate tout en indiquant la méthodologie adoptée durant ce projet et dans le chapitre suivant nous présenterons les définitions et les notions théoriques nécessaires pour la compréhension de notre projet.

CHAPITRE 2

SYSTÈME DE RECOMMANDATION NLP PROPOSÉ

Introduction

Dans le précédent chapitre, nous avons abordé les aspects généraux du sujet avant de nous concentrer sur le contexte spécifique de notre projet, la problématique que nous cherchons à résoudre, ainsi que la solution que nous proposons et la méthodologie que nous avons adoptée. Dans ce chapitre, nous allons approfondir notre compréhension des systèmes de recommandation en expliquant plus en détail leur fonctionnement.

2.1 Les notions de base sur les systèmes de recommandation

Cette section se consacre à l'exploration des notions fondamentales des systèmes de recommandation NLP.

2.1.1 Les systèmes de recommandations

Les systèmes de recommandation sont des systèmes d'information qui analysent les préférences des utilisateurs pour leur recommander des produits ou des services susceptibles de les intéresser. Ces systèmes utilisent des techniques de traitement automatique de l'information et de l'apprentissage automatique pour extraire des informations à partir des données des utilisateurs et des données des produits ou services proposés [4].

2.1.2 Traitement du Langage Naturel NLP

Le Traitement du Langage Naturel (NLP) est une branche de l'intelligence artificielle qui se concentre sur l'interaction entre les ordinateurs et les langages naturels humains, tels que l'anglais ou le français. Le NLP utilise des algorithmes pour analyser, comprendre et générer des langages naturels, ce qui permet aux ordinateurs de communiquer avec les humains de manière plus naturelle et intuitive [5] .

2.1.3 Les systèmes de recommandations NLP

Les systèmes de recommandation Traitement du Langage Naturel (NLP) sont des systèmes de recommandation qui utilisent des techniques de traitement du langage naturel pour extraire des informations à partir de données textuelles telles que les commentaires, les critiques, les publications, les évaluations et les descriptions de produits ou services. Les systèmes de recommandation NLP utilisent ces informations pour comprendre les préférences

et les intérêts des utilisateurs et leur recommander des produits, services ou contenus pertinents. Les techniques de NLP utilisées dans les systèmes de recommandation comprennent la classification de texte, la modélisation de thèmes, l'analyse sémantique et la compréhension du langage naturel.

2.1.4 Historique des systèmes de recommandations NLP

L'histoire des systèmes de recommandation NLP est relativement récente, car elle est née de la convergence de deux domaines distincts : le traitement du langage naturel et les systèmes de recommandation. Les systèmes de recommandation ont commencé à se développer dans les années 1990, avec l'essor du commerce électronique et l'introduction des premières approches de filtrage collaboratif pour recommander des produits en se basant sur les préférences des utilisateurs similaires. Le filtrage basé sur le contenu a également été introduit pour améliorer la pertinence des recommandations [6].

De l'autre côté, le traitement du langage naturel a connu des avancées importantes dans les années 1990 et 2000 avec l'introduction de techniques de modélisation de langage comme les modèles de Markov cachés et les réseaux de neurones. L'utilisation de techniques de traitement du langage naturel dans les systèmes de recommandation a été initiée dans les années 2000, avec l'introduction de techniques telles que la modélisation de thèmes et l'analyse sémantique pour améliorer la qualité des recommandations.

Depuis lors, les systèmes de recommandation NLP ont connu des avancées significatives avec l'introduction de techniques d'apprentissage en profondeur telles que les réseaux de neurones récurrents et les réseaux de neurones convolutifs. De nos jours, ces systèmes sont largement utilisés dans de nombreux domaines.

Cependant, les systèmes de recommandation NLP sont confrontés à des défis importants tels que la compréhension du langage naturel complexe, l'analyse de grands volumes de données et la personnalisation des recommandations pour des utilisateurs individuels. Les développements futurs dans le traitement du langage naturel et l'apprentissage en profondeur devraient conduire à des améliorations continues dans la précision et la pertinence des recommandations, ainsi qu'à de nouvelles opportunités d'application de ces systèmes de recommandation dans divers secteurs.

2.1.5 Avantages des systèmes de recommandation "NLP"

Les systèmes de recommandation NLP peuvent offrir plusieurs avantages dans le contexte des médias sociaux [7] :

- **Une analyse plus approfondie des interactions sociales** : Les systèmes de recommandation NLP peuvent analyser les commentaires, les messages et les interactions sociales des utilisateurs pour comprendre leurs préférences, leurs centres d'intérêt et leur comportement. Cela permet de personnaliser les recommandations en fonction de ces facteurs et d'augmenter la pertinence des recommandations pour chaque utilisateur.
- **L'identification des tendances émergentes** : Les systèmes de recommandation NLP peuvent détecter les tendances émergentes en analysant les commentaires et les messages des utilisateurs sur les réseaux sociaux. Cela permet aux entreprises de réagir rapidement aux tendances du marché et d'adapter leur stratégie en conséquence.
- **Une augmentation de l'engagement des utilisateurs** : En recommandant des contenus pertinents aux utilisateurs, les systèmes de recommandation NLP peuvent augmenter leur engagement avec le contenu et les médias sociaux, ce qui peut conduire à une augmentation de la fidélité et de la satisfaction des clients.

Ces avantages peuvent aider les entreprises à mieux comprendre les préférences et les comportements de leurs clients sur les médias sociaux, à accroître l'engagement et la fidélité de ces derniers et à améliorer leur stratégie de marketing et de communication.

2.1.6 Types des systèmes de recommandation NLP

Il existe différents types de systèmes de recommandation NLP, chacun ayant des caractéristiques spécifiques [8]. Parmi ces systèmes nous citons :

- **Filtrage collaboratif basé sur le texte** : Ce type de système de recommandation utilise des techniques de NLP pour analyser le contenu textuel (par exemple, les critiques ou les commentaires) et identifie des groupes d'utilisateurs ayant des préférences similaires. Les recommandations sont ensuite basées sur les préférences des utilisateurs similaires.
- **Filtrage basé sur le contenu** : Ce type utilise des techniques de NLP pour analyser le contenu textuel et extraire les caractéristiques pertinentes du contenu (par exemple,

les mots-clés, les thèmes ou les sentiments). Les recommandations sont ensuite basées sur les caractéristiques du contenu qui correspondent aux préférences de l'utilisateur.

- **Recommandation contextuelle :** Ce type de recommandation comprend le contexte dans lequel elle est faite, comme la localisation de l'utilisateur, l'heure de la journée, l'âge ou le sexe. Les recommandations sont alors personnalisées en fonction de ces facteurs contextuels.
- **Recommandation de séquence :** Les systèmes de cette catégorie analysent les séquences d'actions de l'utilisateur (par exemple, les clics ou les achats précédents) et génèrent des recommandations basées sur les séquences d'actions précédentes.

La figure 2.1 explique, avec un exemple, les deux types de systèmes de recommandation filtrage collaboratif et filtrage basé sur le contenu.

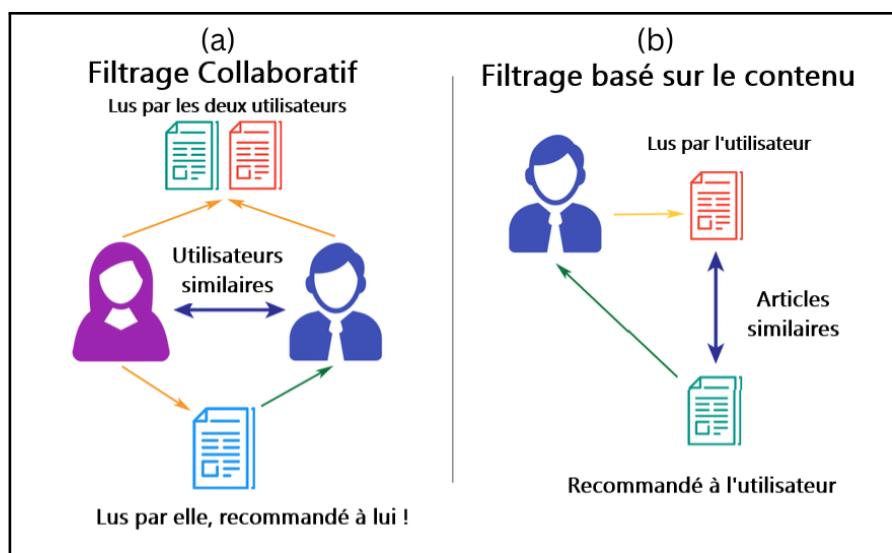


FIGURE 2.1 – Types des systèmes de recommandation NLP, (a) Filtrage Collaboratif, (b) Filtrage basé sur le contenu

Notre projet consiste à développer un système de recommandation basé sur le contenu qui utilise des informations sur les propriétés des éléments pour recommander des éléments similaires.

2.1.7 Différentes applications des systèmes de recommandation NLP

Les systèmes de recommandation NLP ont de nombreuses applications dans différents domaines, tels que :

- **E-commerce :** Afin de recommander des produits en fonction des préférences et des achats passés de l'utilisateur. En analysant les commentaires, les avis et les descriptions de produits, le système de recommandation peut fournir des recommandations plus pertinentes et personnalisées pour chaque utilisateur.
- **Médias sociaux :** Les systèmes de recommandation NLP peuvent être utilisés pour recommander des publications, des vidéos ou des pages à suivre en fonction des préférences de l'utilisateur et des interactions passées. En analysant les commentaires, les likes et les partages, le système de recommandation peut offrir des suggestions plus ciblées et individualisées à chaque utilisateur.
- **Services de streaming :** On utilise également des systèmes de recommandation NLP dans les services de diffusion en continu pour recommander des films, des séries, de la musique ou des podcasts en fonction des préférences et des habitudes d'écoute de chaque utilisateur. En analysant les descriptions, les critiques et les titres de chaque contenu, le système de recommandation peut proposer des conseils mieux adaptés , améliorant ainsi l'expérience de l'utilisateur et maximisant sa satisfaction.
- **Services de recrutement :** Il peut également être utilisé pour recommander des candidats en fonction des préférences et des compétences requises pour chaque poste. En analysant les descriptions de postes et les CV des candidats, le système de recommandation peut fournir des recommandations plus précises et plus spécifiques pour chaque employeur.
- **Services de voyage :** Pour recommander des hôtels, des vols ou des restaurants, les systèmes de recommandation NLP peuvent être utilisés en fonction des préférences de l'utilisateur et de ses réservations antérieures.
- **Assistance à la santé :** Il est possible d'utiliser ces systèmes pour aider les professionnels de la santé à prendre des décisions médicales en recommandant des traitements, des médicaments ou des protocoles de soins en fonction des symptômes, des antécédents médicaux et d'autres données cliniques.

Les systèmes de recommandation NLP peuvent être utilisés dans une variété de domaines pour offrir des recommandations personnalisées, pertinentes et engageantes, ce qui permet d'améliorer l'expérience d'utilisateur et d'optimiser les résultats pour les entreprises et les organisations.

2.2 Le fonctionnement du système de recommandation NLP proposé

Notre système de recommandation NLP proposé pour les médias sociaux repose sur plusieurs étapes clés :

- **Collecte les interactions de l'utilisateur :** Nous collectons les interactions des utilisateurs, telles que les «likes» et les «dislikes», ainsi que les tendances actuelles sur le fil d'actualité. Ces interactions fournissent des informations précieuses sur les préférences et les intérêts des utilisateurs.
- **Modèle NLP pour l'analyse textuelle de topic :** Nous utilisons un modèle de Traitement du Langage Naturel (NLP) pour analyser le contenu textuel des statuts. ce modèle NLP identifie les «topics» ou les sujets principaux abordés dans les statuts en utilisant des techniques telles que la classification de texte ou la modélisation de thèmes.
- **Rechercher des publications similaires dans la base de données :** Une fois que le «topic» est identifié, nous utilisons cette information pour rechercher dans notre base de données MongoDB des statuts similaires. Nous utilisons des algorithmes de recherche et de correspondance pour trouver les statuts qui se rapprochent le plus de ce «topic».
- **Recommandation de contenus similaires :** En utilisant les résultats de la recherche, nous générerons des recommandations personnalisées en proposant à l'utilisateur des statuts similaires à ceux qu'il a aimés ou trendés. Cela permet d'offrir des recommandations pertinentes et adaptées aux intérêts de chaque utilisateur. En outre, le flux de contenu devient dynamique avec des publicités sponsorisées correspondant aux centres d'intérêt de l'utilisateur, ainsi que des actualités de magazines reflétant ses préférences. Cette approche garantit une expérience aux utilisateurs optimale .

Grâce à ces différentes étapes, notre système de recommandation NLP basé sur le filtrage du contenu est capable de fournir des recommandations personnalisées et ciblées aux utilisateurs des médias sociaux. En analysant leurs interactions et en utilisant l'analyse textuelle de «topic», notre système identifie des publications similaires dans notre base de données, permettant ainsi de proposer des recommandations adaptées aux intérêts de chaque utilisateur.

2.3 Système de recommandation basée sur Deep Learning

Dans cette section, nous allons explorer en détails les différentes techniques de Deep Learning utilisées pour créer notre modèle de recommandation. Nous aborderons les méthodes clés qui nous permettront d'identifier les sujets d'intérêt des utilisateurs .

2.3.1 Prétraitement des données

Nous allons explorer les techniques essentielles pour le prétraitement des données dans le contexte des systèmes de recommandation basés sur le NLP .

2.3.1.1 Tokenisation

Selon Manning et Schütze (1999) [9] : La tokenisation est une étape fondamentale dans le traitement du langage naturel qui consiste à diviser un texte en unités individuelles appelées «tokens». Ceux-ci peuvent être des mots, des expressions ou même des caractères, selon la tâche ou l'application spécifique. L'objectif de la tokenisation est de segmenter le texte en unités significatives qui peuvent être ensuite traitées et analysées. La figure ci-dessous 2.2 explique le principe de la Tokenisation.

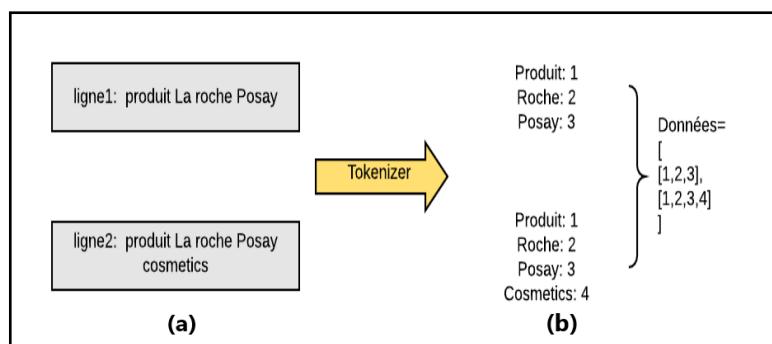


FIGURE 2.2 – Principe de Tokenizer, (a) avant Tokenizer, (b) après Tokenizer

2.3.1.2 Padding de séquences

Selon François Chollet (2018) [10] : Le padding de séquence permet de s'assurer que toutes les séquences d'entrée ont la même longueur. Dans de nombreux cas, les textes peuvent avoir des longueurs variables, ce qui peut poser des problèmes lors de l'utilisation de modèles d'apprentissage profond qui nécessitent des entrées de taille fixe. Il s'agit d'ajouter des éléments spécifiques (généralement des zéros) à la fin des séquences plus courtes pour qu'elles aient la même longueur que les séquences plus longues. La figure ci-dessous 2.3 explique le principe de Padding de séquences.

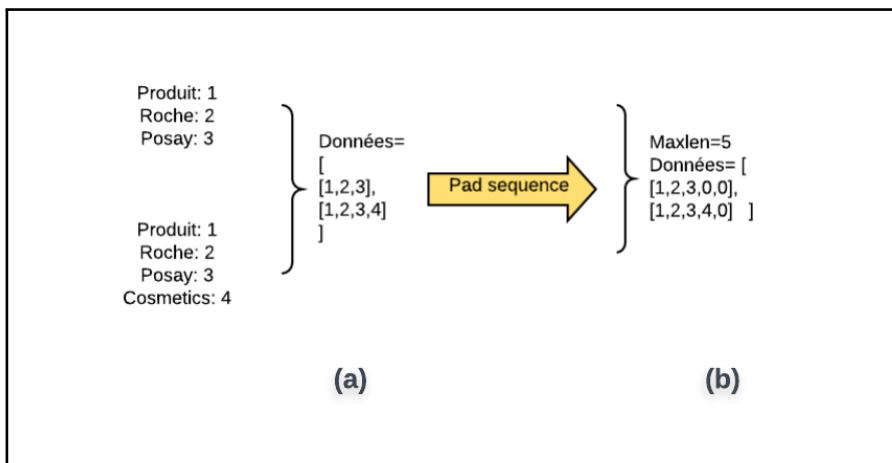


FIGURE 2.3 – Principe de pad sequences, (a) avant pad sequence, (b) après pad sequence

2.3.1.3 L'approche Id2Word

L'approche Id2Word, également connue sous le nom d'indexation des mots, est une technique utilisée dans le traitement du langage naturel pour attribuer un identifiant unique à chaque mot d'un corpus de texte. Au lieu de représenter les mots par des vecteurs de nombres réels, comme dans les approches d'embedding, l'Id2Word associe un identifiant numérique (ou token) à chaque mot. L'utilisation de l'approche Id2Word présente plusieurs avantages, notamment une réduction de la dimensionnalité des données, une meilleure efficacité de stockage et une facilité d'utilisation dans les modèles de traitement du langage naturel.

2.3.2 Vectorisation des données (Word Embedding)

Le Word Embedding a révolutionné la représentation des mots dans le traitement du langage naturel. Plutôt que de les considérer comme des symboles isolés, le Word Embedding attribue à chaque mot un vecteur numérique qui capture les relations sémantiques et contextuelles. Cette approche permet aux modèles de mieux comprendre la signification des mots dans un contexte spécifique, améliorant ainsi la précision des tâches telles que la classification de texte, la traduction automatique et les systèmes de recommandation. En utilisant le Word Embedding, les modèles peuvent exploiter les similarités entre les mots et saisir les nuances du langage humain [11].

2.3.2.1 L'approche Word2Vec

L'approche Word2Vec est une méthode populaire de Word Embedding qui vise à apprendre des représentations vectorielles denses pour les mots. Elle permet de capturer les similarités

sémantiques et les relations entre les mots dans un corpus de texte. Word2Vec se base sur l'idée selon laquelle les mots ayant des contextes similaires ont tendance à avoir des significations similaires.

La Figure 2.4 montre l'une des images les plus fréquemment utilisées dans Word2Vec. La proximité sémantique entre ces mots est équivalente à la proximité mathématique des valeurs vectorielles entre eux. L'un des exemples souvent donnés est l'équation “king-man + woman = queen”. Ce qui se passe ici, c'est que la valeur vectorielle obtenue en soustrayant et en ajoutant les vecteurs les uns aux autres est égale au vecteur correspondant à l'expression “queen”. On peut comprendre que les mots “king” et “queen” sont très similaires, mais des différences vectorielles apparaissent uniquement en raison de leur genre.

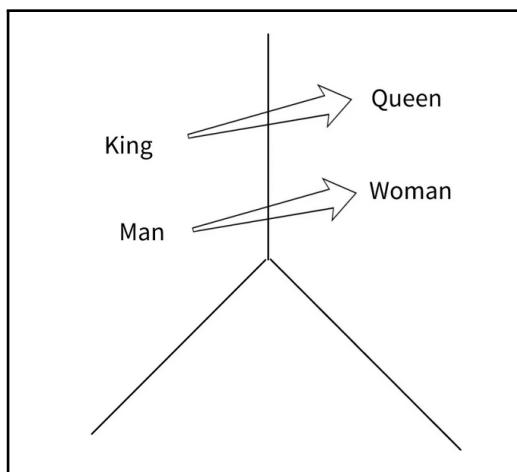


FIGURE 2.4 – Similarité des mots

Il existe deux modèles Word2Vec couramment utilisés : le modèle Skip-gram et le modèle Continuous Bag-of-Words (CBOW). Le modèle Skip-gram vise à prédire les mots environnants à partir d'un mot cible, tandis que le modèle CBOW prédit un mot cible à partir de ses mots environnants. Ces modèles permettent de générer des embeddings de mots riches en information. Cette différence est montrée dans la figure 2.5 suivante .

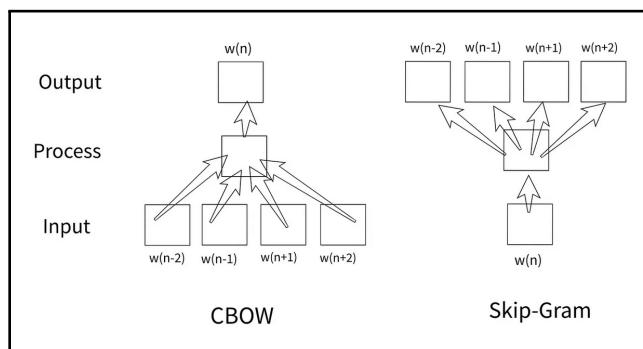


FIGURE 2.5 – Skip-gram vs CBOW

2.3.2.2 L'approche TF-IDF

L'approche TF-IDF (Term Frequency-Inverse Document Frequency) est une méthode couramment utilisée pour représenter les mots dans le NLP. Contrairement à l'approche Word2Vec, TF-IDF est basée sur une pondération statistique des mots dans un corpus de documents.

TF (Fréquence des Termes) mesure l'importance d'un mot dans un document spécifique en calculant le nombre d'occurrences du mot divisé par le nombre total de mots dans le document. IDF (Fréquence Inverse des Documents) mesure l'importance d'un mot dans le corpus global en calculant le logarithme du nombre total de documents divisé par le nombre de documents contenant le mot.

La valeur TF-IDF est obtenue en multipliant la valeur TF par la valeur IDF. Ainsi, les mots qui apparaissent fréquemment dans un document mais rarement dans le reste du corpus auront une valeur TF-IDF élevée, ce qui indique leur importance relative.

L'équation 2.1 représente l'expression de Tf-Idf.

$$\text{TF-IDF} = \text{TF} \times \text{IDF} = \text{TF} \times \log\left(\frac{N}{\text{DF}}\right) \quad (\text{Eq. 2.1})$$

où

N = Nombre total de mots.

DF = Nombre de documents qui contiennent le terme considéré.

La Figure 2.6 représente le résultat de la fonction TF-IDF sur une phrase convertie par Count Vectorizer dans l'étape précédente.

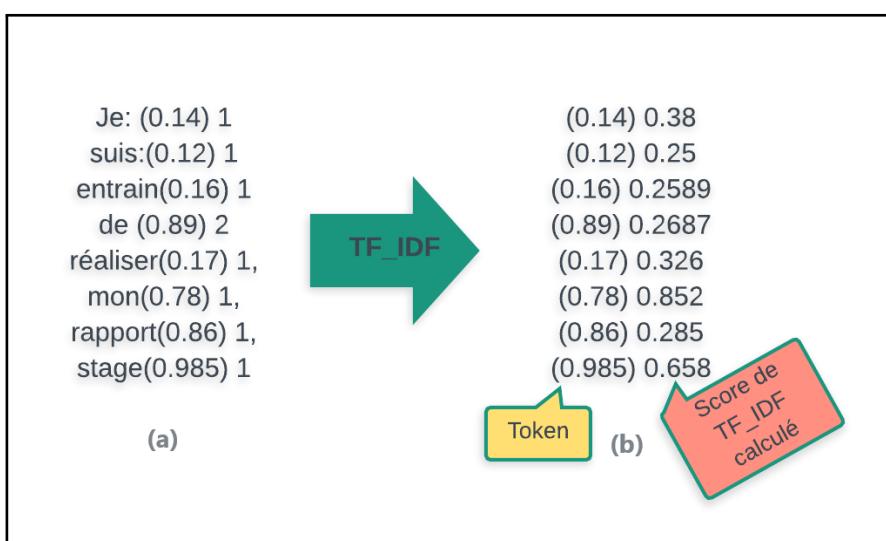


FIGURE 2.6 – La conversion du TF-IDF , (a) avant TF-IDF, (b) après TF-IDF

2.3.2.3 L'approche One Hot Encoding

L'approche One Hot Encoding est une méthode de représentation des données utilisée dans le NLP. Elle consiste à représenter chaque mot ou token d'un texte par un vecteur binaire de dimension égale au nombre total de mots distincts dans le corpus. Dans cette représentation, tous les éléments du vecteur sont nuls, à l'exception d'un unique élément qui est mis à 1 pour indiquer la présence du mot correspondant. Par exemple, supposons que notre corpus contienne les mots "MacOS", "Windows" et "Linux". Chaque mot sera représenté par un vecteur de taille 3 : [1, 0, 0] pour "MacOs", [0, 1, 0] pour "Linux" et [0, 0, 1] pour "Windows". Ainsi, chaque mot est représenté par un vecteur distinct, et l'ensemble des vecteurs forme une matrice appelée la matrice One Hot Encoding. L'avantage de l'approche One Hot Encoding est sa simplicité et son interprétabilité. Cependant, cette méthode peut conduire à des vecteurs de grande dimension, en particulier pour les corpus de grande taille avec un vocabulaire étendu. De plus, elle ne capture pas les informations sémantiques ou relationnelles entre les mots, car chaque mot est considéré comme indépendant des autres. La figure ci-dessous 2.7 explique le principe One Hot Encoding .

X	MacOS	Linux	Windows
MacOS	1	0	0
Windows	0	0	1
MacOS	1	0	0
Linux	0	1	0

FIGURE 2.7 – Exemple de One Hot Encoding

⇒ Dans les CNN et les LSTM, la couche d'embedding est généralement basée sur word2vec ou des embeddings de mots pré-entraînés, tels que GloVe ou FastText, plutôt que sur l'encodage one-hot ou TF-IDF. Ces techniques d'embedding captent les relations sémantiques entre les mots et fournissent des représentations plus significatives pour les données textuelles.

Dans la section suivante, nous examinerons de plus près le fonctionnement des modèles de deep learning dans le contexte des systèmes de recommandation basés sur le traitement du langage naturel.

2.3.3 Modèles de deep learning pour la recommandation basée sur NLP

Dans cette partie, nous explorerons différentes architectures de modèles de deep learning pour la recommandation basée sur le NLP. Nous aborderons spécifiquement les modèles CNN (Convolutional Neural Networks), LSTM (Long Short-Term Memory), BLSTM(Bidirectional Long Short-Term Memory) et GRU (Gated Recurrent Unit) qui sont largement utilisés dans le domaine de la recommandation. Nous détaillerons le fonctionnement de chaque modèle, en mettant l'accent sur leur capacité à traiter et à extraire des informations à partir de données textuelles.

2.3.3.1 Convolutional Neural Networks (CNN)

Un Convolutional Neural Networks (CNN), également appelé réseau de neurones convolutifs, est une architecture de réseau neuronal utilisée dans le domaine du traitement du langage naturel (NLP) pour la compréhension et l'analyse de texte. Les CNN sont initialement conçus pour la vision par ordinateur, mais ils ont été adaptés avec succès pour le traitement de texte. Ils sont capables d'apprendre des représentations hautement discriminantes des données textuelles [12]. L'architecture du modèle CNN est illustrée dans la figure 2.8.

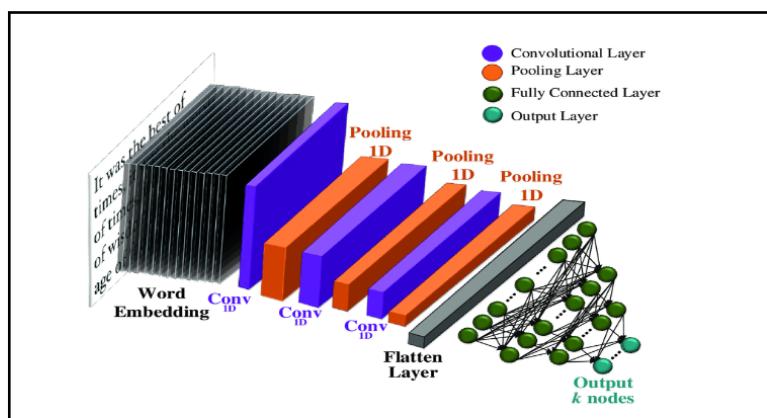


FIGURE 2.8 – L'architecture du modèle CNN dans le domaine NLP

L'architecture des CNN comprend plusieurs couches clés qui permettent d'extraire des caractéristiques pertinentes à partir des données textuelles. Voici une description de ces couches :

- Couche d'entrée :** La première couche du CNN reçoit les données textuelles en tant que séquences de mots ou de caractères. Chaque mot est généralement représenté par un vecteur d'embedding pré-entraîné, qui capture les informations sémantiques et contextuelles.
- Couches de convolution :** Les couches de convolution sont responsables de la détection de motifs et de caractéristiques dans les données textuelles. Elles utilisent des

filtres convolutifs qui se déplacent sur les séquences de mots ou de caractères pour extraire des informations locales. Différents filtres peuvent être utilisés pour capturer différentes caractéristiques, tels que les n-grammes ou les motifs syntaxiques.

3. **Couches de pooling :** Les couches de pooling réduisent la dimensionnalité des caractéristiques extraites par les couches de convolution. Le pooling peut être réalisé en utilisant des opérations telles que le max pooling ou le average pooling, qui agrègent les valeurs des caractéristiques sur des fenêtres de taille fixe.
4. **La couche de flatten :** est responsable de la transformation des caractéristiques spatiales multidimensionnelles en une représentation unidimensionnelle. Elle permet de convertir les informations spatiales en une séquence linéaire, qui peut ensuite être utilisée comme entrée pour les couches fully-connected.
5. **Couches fully-connected :** Après les couches de convolution et de pooling, les caractéristiques extraites sont aplatis et connectées à des couches fully-connected (entièremenr connectées). Ces couches sont responsables de l'apprentissage des relations entre les caractéristiques et la génération de représentations denses.
6. **Couche de sortie :** La couche de sortie du CNN effectue la classification ou la prédiction finale en utilisant des fonctions d'activation appropriées. Dans le contexte de la recommandation de texte, elle peut être utilisée pour prédire les préférences ou les intérêts des utilisateurs en fonction des données textuelles.

La Figure 2.9 représente cette architecture de CNN :

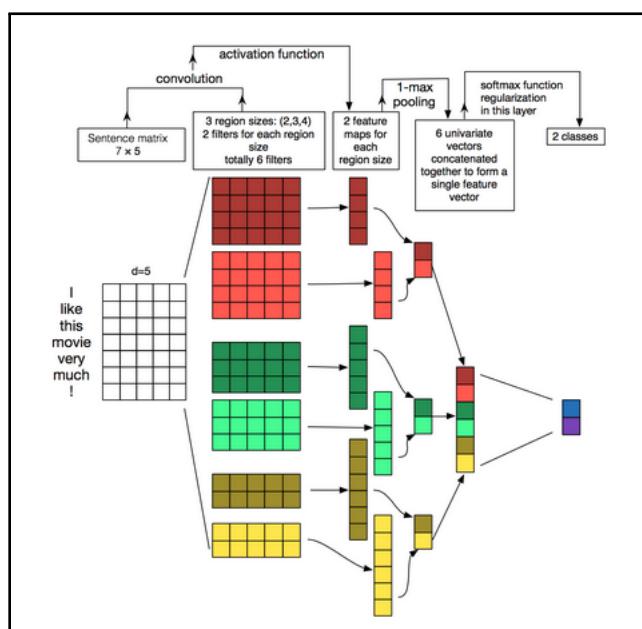


FIGURE 2.9 – L'architecture du modèle CNN de classification de texte

En utilisant cette architecture de CNN, les systèmes de recommandation de texte peuvent exploiter les informations contextuelles et sémantiques pour générer des recommandations plus précises et pertinentes. Cependant, il est important de prendre en compte les défis spécifiques liés à la taille des séquences de texte, au déséquilibre des données et au choix des hyperparamètres pour optimiser les performances du modèle.

2.3.3.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM), qui signifie Mémoire Longue à Court Terme en français, est une architecture de Recurrent Neural Network (RNN) conçue pour résoudre les limitations des RNN traditionnels dans la capture et la préservation des dépendances à long terme dans les données séquentielles. Il a été introduit par Hochreiter et Schmidhuber en (1997) [13]. Les RNN traditionnels sont confrontés au problème de disparition du gradient (vanishing gradients), où les gradients diminuent exponentiellement à mesure qu'ils se propagent en arrière dans le temps, ce qui rend difficile l'apprentissage des dépendances à long terme. La Figure 2.10 explique cette caractéristique de RNN.

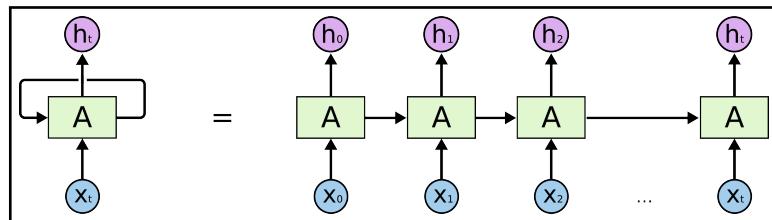


FIGURE 2.10 – L'architecture du modèle RNN

Dans l'algorithme de RNN, une seule fonction d'activation (activation function, AF) est utilisée. Cette fonction est la tangente hyperbolique (tanh), elle est exprimée par l'équation 2.2 et dans la Figure 2.11.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (\text{Eq. 2.2})$$

où :

x_t : L'entrée.

h_{t-1} : La couche cachée précédente.

h_t : La couche cachée courante.

La Figure 2.11 montre la structure interne de RNN :

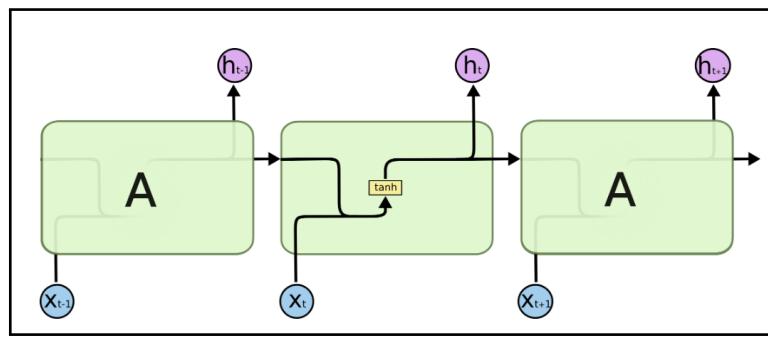


FIGURE 2.11 – La structure interne de RNN

Les LSTM résolvent ce problème en introduisant une cellule de mémoire qui permet aux informations de circuler à travers le réseau sur de longues périodes de temps sans être diluées ou perdues. Le composant clé d'une unité LSTM est la cellule de mémoire, qui est responsable du stockage et de la mise à jour des informations dans le temps. Chaque cellule de mémoire est composée de trois portes principales :

- Porte d'entrée (Input Gate) : Elle détermine la quantité d'informations nouvelles à ajouter à l'état de la mémoire à long terme en fonction de l'entrée actuelle et de l'état caché précédent.
- Porte d'oubli (Forget Gate) : Elle contrôle la quantité d'informations qui doit être supprimée de l'état de la mémoire à long terme, permettant ainsi de conserver uniquement les informations pertinentes.
- Porte de sortie (Output Gate) : Elle détermine la quantité d'informations à extraire de l'état de la mémoire à long terme pour être utilisée en tant que sortie de la cellule LSTM.

Donc, la mémoire de LSTM est plus importante que le RNN. Le Figure 2.12 exprime l'architecture de l'algorithme de LSTM.

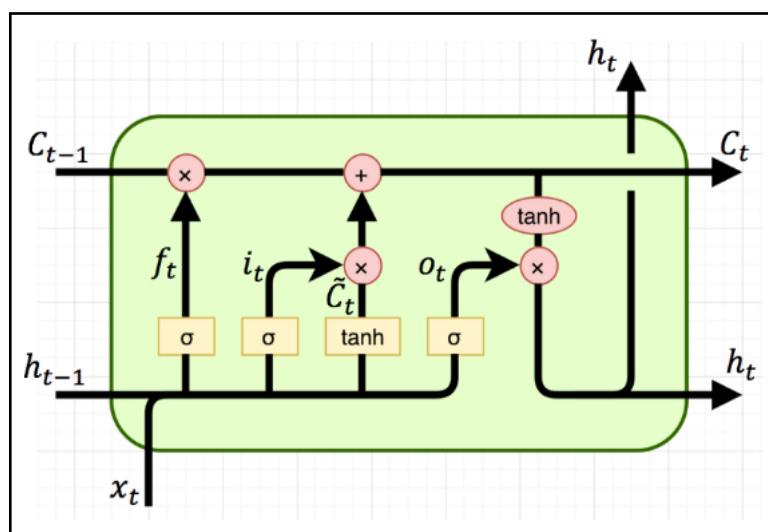


FIGURE 2.12 – L'architecture de LSTM

Nous remarquons que LSTM se diffère par rapport à RNN, par le nombre des fonctions d'activations. RNN utilise une seule fonction, alors que LSTM est construit par cinq fonctions de sigmoid et tanh afin de reformuler les quatre states qui sont :

- Forgate gate (f).
- Input gate (i).
- Output gate (o).
- Cell state (c).

Les réseaux LSTM sont couramment utilisés dans une variété de tâches impliquant des données séquentielles, grâce à leur capacité à capturer les dépendances à long terme.

2.3.3.3 Bidirectional Long Short-Term Memory (BLSTM)

Bidirectional Long Short-Term Memory (BLSTM) est une variante de l'architecture LSTM (Long Short-Term Memory) qui intègre un traitement bidirectionnel des données séquentielles. Il combine deux couches LSTM, l'une traitant la séquence dans la direction avant et l'autre dans la direction arrière. Cela permet au BLSTM de capturer à la fois le contexte passé et futur.

Dans un BLSTM, la séquence d'entrée est alimentée dans deux couches LSTM distinctes, l'une traitant la séquence dans la direction avant, de début à fin, et l'autre la traitant dans la direction arrière, de fin à début. Chaque couche LSTM maintient ses propres états cachés et états de cellule. Les sorties des deux couches LSTM sont généralement concaténées ou combinées d'une manière quelconque pour produire la sortie finale du BLSTM. La Figure 2.13 montre la structure interne de BLSTM.

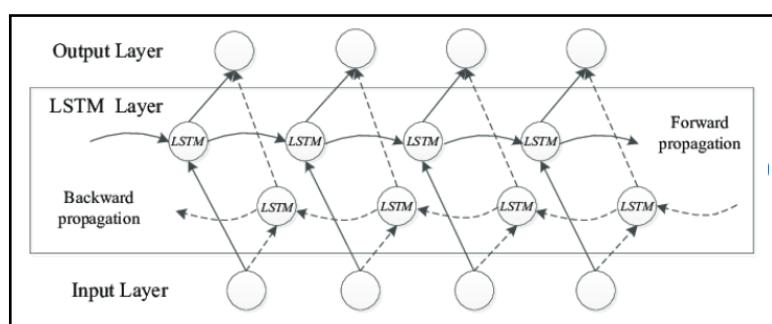


FIGURE 2.13 – La structure interne de BLSTM

Le fonctionnement du BLSTM se divise en deux parties :

1. Propagation avant (forward pass) : L'entrée séquentielle est alimentée dans le LSTM de l'ordre direct et génère une séquence d'états cachés (\vec{h}) correspondant à chaque instant de

temps. Chaque état caché est calculé en fonction de l'état caché précédent et d'entrée actuelle.

2. Propagation arrière (backward pass) : L'entrée séquentielle est alimentée dans le LSTM de l'ordre inverse et génère une séquence d'états cachés (\overleftarrow{h}) correspondant à chaque instant de temps. Chaque état caché est calculé en fonction de l'état caché suivant et d'entrée actuelle.

2.3.3.4 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU), est une autre architecture de réseau neuronal récurrent (RNN) qui est similaire au LSTM (Long Short-Term Memory). Il a été proposé par Cho et al. en 2014 comme une alternative plus simple et légère au LSTM, tout en conservant une bonne capacité de modélisation des dépendances à long terme [14]. Le GRU se compose également d'une cellule de mémoire, mais il utilise un mécanisme de portes légèrement différent de celui du LSTM. Contrairement au LSTM, le GRU n'a pas de porte d'oubli explicite. Au lieu de cela, comprend deux portes principales qui régulent le flux d'informations :

- Porte de réinitialisation (Reset Gate) : Contrôle la quantité d'informations provenant de l'étape précédente qui doit être oubliée ou ignorée. Elle prend en compte l'état caché précédent et l'entrée courante pour décider de l'information à réinitialiser dans la cellule de mémoire. Cette porte permet de capturer les dépendances à court terme.
- Porte d'activation (Update Gate) : Détermine la quantité d'informations provenant de l'étape précédente qui doit être transmise à l'étape actuelle. Elle contrôle la mise à jour de l'état de la cellule de mémoire en fonction de l'état caché précédent et de l'entrée courante. Cette porte permet de capturer les dépendances à long terme.

La Figure 2.14 exprime l'architecture de l'algorithme de GRU :

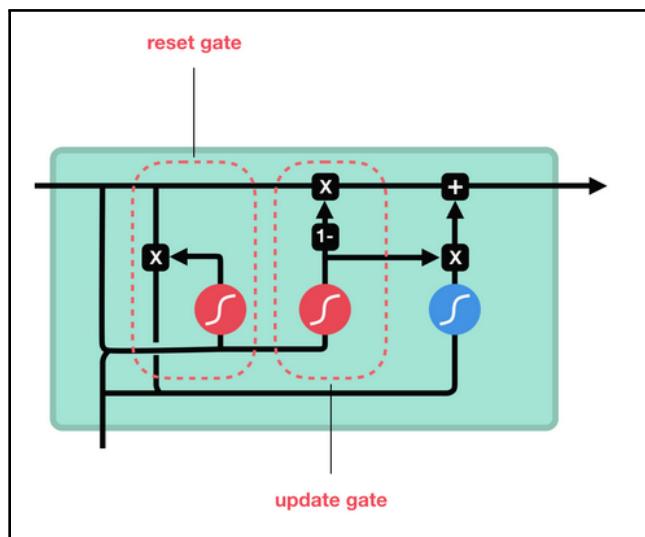


FIGURE 2.14 – L'architecture de GRU

Les équations 2.3 mathématiques décrivant les calculs du GRU :

$$\begin{aligned}
 r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] b_r) \\
 z_t &= \sigma(W_z \cdot [h_{t-1}, x_t] b_z) \\
 \hat{h}_t &= \tanh(W \cdot [r_t \odot h_{t-1}, x_t] b) \\
 h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t
 \end{aligned} \tag{Eq. 2.3}$$

Ces équations représentent les mises à jour de la porte de réinitialisation (reset gate), de la porte d'activation (update gate), de l'état candidat et de l'état caché (hidden state) dans le GRU. Les symboles σ représentent la fonction sigmoïde, \tanh représente la fonction tangente hyperbolique, \cdot représente la multiplication matricielle, \odot représente la multiplication élément par élément, W_r , W_z , W sont les matrices de poids, b_r , b_z , b sont les biais, h_{t-1} et x_t sont respectivement l'état caché précédent et l'entrée à l'instant t , r_t est la porte de réinitialisation, z_t est la porte d'activation, \hat{h}_t est l'état candidat, et h_t est l'état caché mis à jour.

2.3.4 Protocoles expérimentaux d'évaluer et améliorer les performances des modèles

Dans cette partie, nous explorons différentes métriques d'évaluation. De plus, nous examinons en détail le phénomène de l'overfitting et proposons des stratégies efficaces pour le combattre.

2.3.4.1 Métriques d'évaluation

Pour mesurer la performance de notre système de recommandation proposé, nous avons utilisé les protocoles Accuracy, Confusion Matrix, Recall et Precision. Pour faciliter l'explication de ces protocoles, nous supposons deux types d'instances : positives et négatives, de sorte qu'une comparaison entre les étiquettes réelles et prédites donne quatre cas :

- **TP(True Positive)** : Le TP se produit lorsque notre modèle identifie correctement une instance comme positive alors qu'elle l'est en réalité positive.
- **TN(True Negative)** : Le TN se produit lorsque notre modèle identifie correctement une instance comme négative alors qu'elle l'est en réalité négative.
- **FP(False Positive)** : Le FP se produit lorsque notre modèle identifie à tort une instance comme positive, alors qu'elle est en fait négative.

- **FN(False Negative)** : Le FP se produit lorsque notre modèle identifie à tort une instance comme négative, alors qu'elle est en fait positive.

Ces mesures d'évaluation sont exprimées respectivement par les équations 2.4-2.5-2.6 et la figure 2.15 :

- **Accuracy** : L'accuracy, mesure la proportion de prédictions correctes par rapport au nombre total d'échantillons. C'est la métrique globale utilisée pour évaluer la performance globale du modèle. L'accuracy est calculée comme suit :

$$\text{Accuracy} = \frac{\sum TP + TN}{\sum TP + FP + FN + TN} \quad (\text{Eq. 2.4})$$

- **Recall** : Le recall, mesure la capacité du modèle à détecter les vrais positifs d'une classe donnée. Il représente le ratio entre le nombre de vrais positifs prédicts et le nombre total de vrais positifs réels. Le recall est calculé comme suit :

$$\text{Recall} = \frac{\sum TP}{\sum TP + FN} \quad (\text{Eq. 2.5})$$

- **Précision** : La précision mesure la proportion de prédictions positives correctes parmi toutes les prédictions positives effectuées par le modèle. Elle permet d'évaluer la qualité des prédictions positives du modèle. La précision est calculée comme suit :

$$\text{Précision} = \frac{\sum TP}{\sum TP + FP} \quad (\text{Eq. 2.6})$$

- **Matrice de confusion** : La matrice de confusion dans la figure 2.15 fournit une évaluation détaillée des performances du modèle par classe. Elle répertorie les prédictions effectuées par le modèle pour chaque classe, en comparaison avec les véritables étiquettes de classe. La matrice de confusion est souvent utilisée pour calculer d'autres métriques telles que le recall et la précision.

		Real Label	
		Positive	Negative
Predicted Label	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

FIGURE 2.15 – la matrice de confusion

2.3.4.2 Overfitting

L'Overfitting est un problème courant dans le Machine Learning (ML) et Deep Learning (DL) où un modèle performe extrêmement bien sur les données d'entraînement mais échoue à généraliser correctement sur de nouvelles données inconnues. Cela se produit lorsque le modèle devient trop complexe et commence à mémoriser ou à "surapprendre" le bruit et les détails des données d'entraînement, au lieu de capturer les schémas ou relations sous-jacentes. Dans les modèles de NLP, certains mots peuvent être rares ou spécifiques au corpus d'entraînement. Si le modèle devient trop spécialisé dans ces mots rares, il peut avoir du mal à généraliser correctement lorsqu'il est confronté à de nouveaux mots ou à des mots peu fréquents dans les données de test. La figure 2.16 le montre clairement.

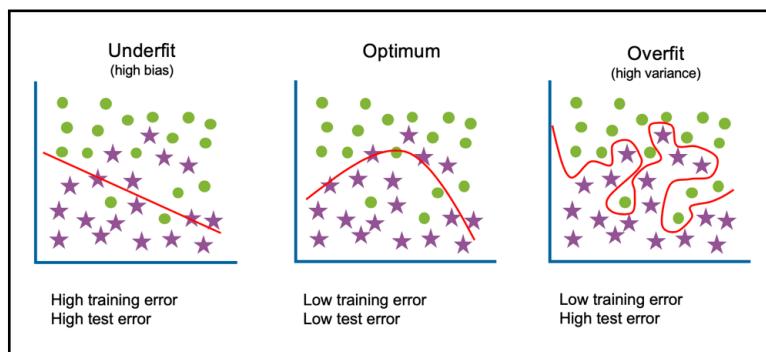


FIGURE 2.16 – Overfitting dans le Deep Learning

2.3.4.3 Solutions pour prévenir l'overfitting

L'overfitting est un problème courant dans le deep learning, mais il existe plusieurs techniques pour le prévenir et améliorer les performances des modèles. Deux des solutions les plus couramment utilisées sont le dropout et l'early stopping.

- **Dropout :** Le dropout est une technique de régularisation très efficace qui consiste à désactiver de manière aléatoire un certain pourcentage de neurones pendant l'entraînement. Cela force le réseau à ne pas trop dépendre de certains neurones spécifiques et favorise une représentation plus robuste des données. Le dropout permet de réduire l'overfitting en évitant que le réseau ne mémorise trop les exemples d'entraînement spécifiques, tout en favorisant une généralisation plus solide aux nouvelles données.
- **Early stopping :** L'early stopping est une autre technique populaire pour prévenir l'overfitting. Elle consiste à arrêter l'entraînement du modèle dès que les performances sur un ensemble de validation commencent à se dégrader. L'idée sous-jacente est que le

modèle atteint son point optimal lorsque les performances sur les données de validation cessent de s'améliorer et commencent à se détériorer. L'early stopping permet d'équilibrer la capacité de généralisation et overfitting aux données d'entraînement.

⇒ En utilisant ces techniques, ainsi que d'autres méthodes de régularisation telles que la réduction de la taille du modèle, la normalisation des données et l'augmentation des données, il est possible de prévenir l'overfitting et d'améliorer les performances des modèles de DL.

Conclusion

Au cours de ce chapitre, nous avons d'abord fait une présentation globale du système de recommandations. Ensuite, nous avons expliqué le fonctionnement de notre système de recommandation. Enfin, nous avons parlé de l'approche de recommandation basée sur l'apprentissage profond. Le chapitre suivant est réservé à la conception et à l'analyse des besoins.

CHAPITRE 3

CONCEPTION ET ARCHITECTURE DU PROJET

Introduction

Dans le chapitre précédent, nous avons discuté des notions clés utilisées dans la solution que nous proposons. Dans ce chapitre, nous allons entamer la phase d'analyse, de spécification et de conception. Nous commencerons par identifier les acteurs et effectuerons une analyse globale pour spécifier les besoins fonctionnels et non fonctionnels du projet. Ensuite, nous présenterons les diagrammes de cas d'utilisation, les diagrammes de classes, les diagrammes de séquences . Nous introduirons ensuite le backlog du produit que nous suivrons tout au long de notre projet. Enfin, nous présenterons l'architecture de notre application et l'environnement de travail. À la fin de cette phase, nous aurons un modèle complet, cohérent et fiable qui reflète les besoins de l'entreprise tout en tenant compte des exigences et des contraintes de la réalisation.

3.1 Spécification des besoins

La spécification des besoins implique une compréhension approfondie des besoins et des exigences afin de concevoir le système et d'en fournir une description opérationnelle. Cette étape permet de structurer l'ensemble du système et d'identifier les acteurs impliqués. Ensuite, nous détaillerons les exigences fonctionnelles et non fonctionnelles correspondantes, en fournissant une liste détaillée des fonctionnalités attendues du système ainsi que des contraintes et des critères de performance.

3.1.1 Identification des acteurs

«Un acteur est l'idéalisat^{ion} d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système.» Notre Platforme se compose d'un seul acteur suivant :

- **Utilisateur** : Une personne qui utilise la plateforme pour accéder à ses fonctionnalités et services.

3.1.2 Besoins fonctionnels

Les besoins fonctionnels sont les exigences spécifiques d'un système informatique qui décrivent les fonctionnalités et les actions qu'il doit être capable d'effectuer pour répondre aux

besoins des utilisateurs.

Les besoins fonctionnels associés à notre solution seront présentés comme suit :

- **Inscription** : Les utilisateurs s'inscrivent en fournissant les informations requises telles que leur nom, leur adresse e-mail et leur mot de passe.
- **Authentification** : Les utilisateurs peuvent se connecter en saisissant leur adresse e-mail et leur mot de passe pour accéder à leur espace personnel.
- **Réinitialisation de mot de passe** : Les utilisateurs peuvent réinitialiser leur mot de passe en fournissant un code de vérification envoyé par e-mail. Une fois le code et l'email validés, ils peuvent changer leur mot de passe.
- **Gestion de profil** : Les utilisateurs peuvent consulter et modifier leurs informations personnelles sur la plateforme.
- **Gestion des messages** : Les utilisateurs peuvent consulter leurs conversations avec d'autres utilisateurs et leur envoyer des messages.
- **Gestion des publications** : Les utilisateurs peuvent créer des publications, explorer les publications existantes, supprimer leurs propres publications, partager des publications, réagir aux publications, ajouter des commentaires et supprimer leurs propres commentaires. Ils peuvent également consulter les commentaires d'autres utilisateurs.
- **Recommandation de contenu personnalisé** : Les utilisateurs reçoivent des recommandations de contenu basées sur l'apprentissage approfondi, en fonction de leurs préférences et de leur historique d'interaction. Cela peut inclure des recommandations de publications, de contenus sponsorisés et d'actualités.

3.1.3 Besoins non fonctionnels

Les besoins non fonctionnels sont des exigences et des critères essentiels pour assurer la qualité et la performance d'un système, indépendamment de ses fonctionnalités spécifiques. Notre système doit respecter les exigences non fonctionnelles suivantes :

- **Convivialité** : le système doit être facile à utiliser et à comprendre pour les utilisateurs.
- **Évolutivité** : Le système doit être capable de s'adapter aux besoins futurs et de croître avec l'entreprise.
- **Sécurité** : Le système doit garantir la confidentialité, l'intégrité et la disponibilité des données.
- **Maintenabilité et scalabilité** : Le code source doit être lisible, compréhensible pour

assurer son état évolutif et extensible aux besoins de l'entreprise.

- **La performance :** Le système doit exécuter ses fonctionnalités de manière efficace et efficiente, en tenant compte de critères tels que la vitesse, le temps de réponse, la capacité de traitement et la consommation de ressources.

3.2 Étude analytique

Cette section concerne la présentation des diagrammes UML de notre application.

3.2.1 Diagramme de cas d'utilisation général

Les diagrammes de cas d'utilisation sont des diagrammes UML qui permettent d'avoir une vue d'ensemble statique du comportement fonctionnel d'un système logiciel.

Pour clarifier les exigences fonctionnelles et éliminer toute ambiguïté, nous allons représenter visuellement les attentes de l'utilisateur du système à l'aide d'un diagramme de cas d'utilisation général, comme illustré dans la figure 3.1 ci-dessous. Cette représentation visuelle facilitera la compréhension et la communication des besoins entre les parties prenantes.

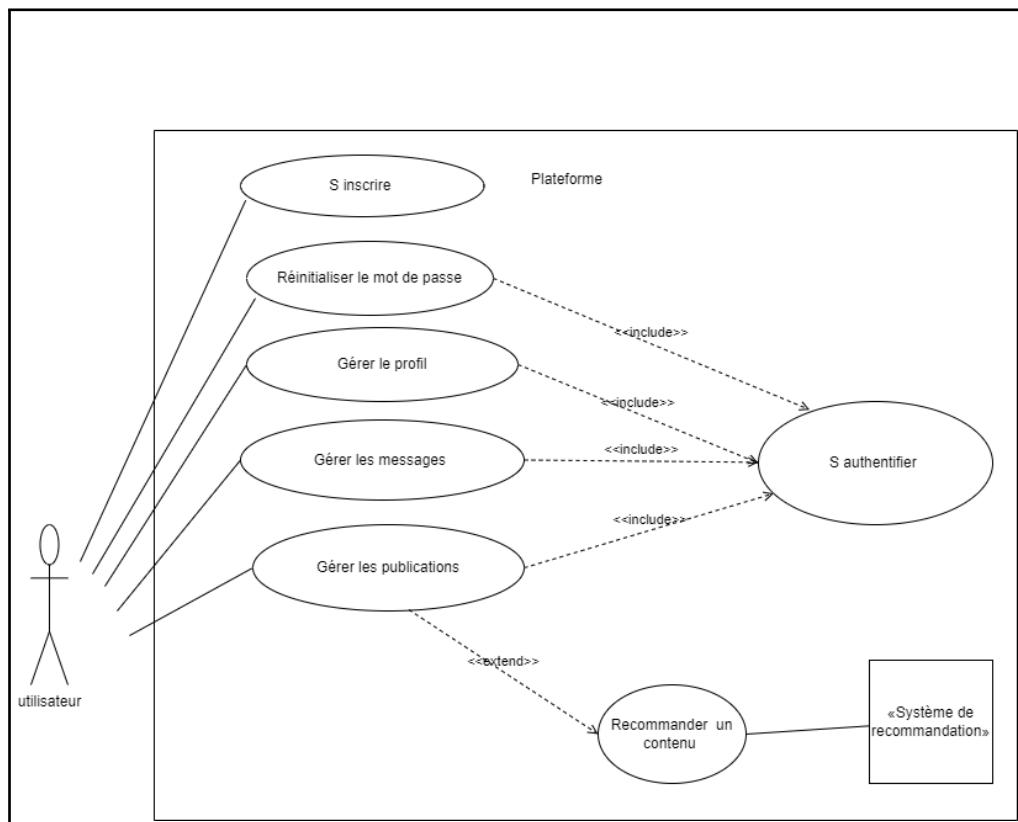


FIGURE 3.1 – Diagramme de cas d'utilisation général

3.2.2 Diagramme de classe global

Le diagramme de classes joue un rôle essentiel dans la modélisation orientée objet en offrant une représentation visuelle de la structure interne d'un système. Il permet de représenter de manière abstraite les objets qui interagissent pour accomplir les cas d'utilisation. Le diagramme de classes fournit une vue globale de l'organisation, de la structure et des relations entre les classes, ainsi que de leurs attributs, dans un système logiciel. Dans la figure 3.2 ci-dessous, nous présentons le diagramme de classe global.

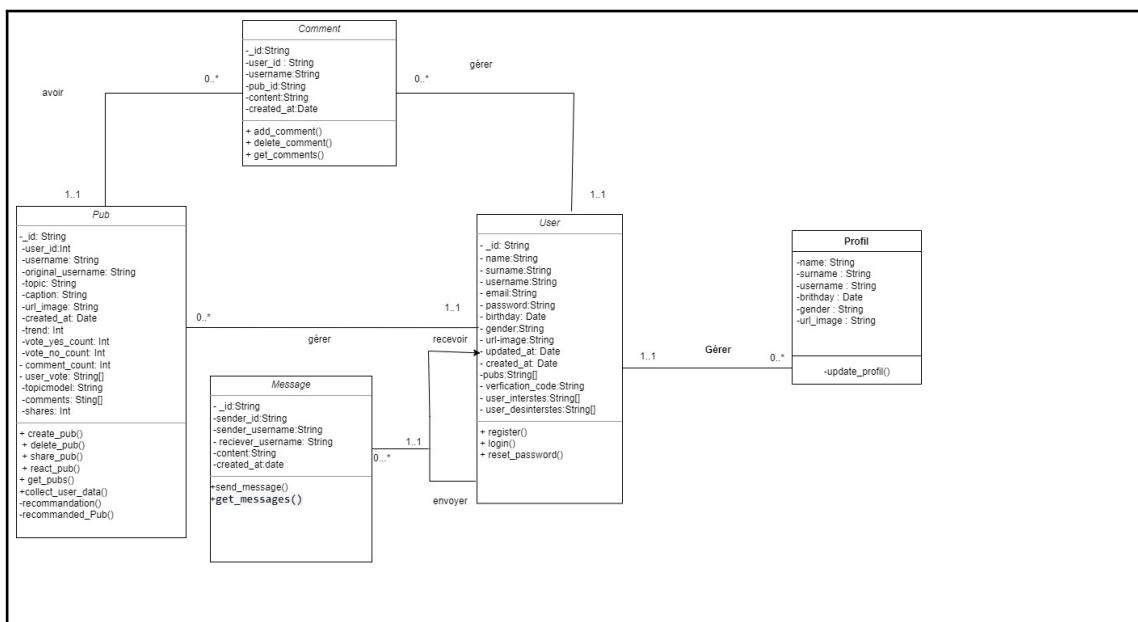


FIGURE 3.2 – Diagramme de classe global

3.3 Pilotage du projet avec Scrum

Cette section présente le Backlog du produit et la planification des releases.

3.3.1 Le Backlog du produit

Le tableau 3.1 représente le Product Backlog, qui est l'élément central de la méthodologie Scrum, jouant un rôle crucial dans la gestion du développement du produit. Il représente l'ensemble des fonctionnalités essentielles, appelées "user stories", qui décrivent en détail le produit souhaité. Dans notre contexte spécifique, le Backlog produit est structuré en utilisant les champs suivants :

- **ID :** Un identifiant unique attribué à chaque user story, permettant de les identifier de manière distincte.
- **Description(User story) :** Une description détaillée du travail à réaliser, spécifiant les fonctionnalités attendues et les exigences associées.
- **Priorité :** Un paramètre déterminant l'ordre dans lequel les tâches doivent être effectuées, en se basant sur leur valeur relative pour le produit.

TABLEAU 3.1 – Backlog de produit

ID	Fonctionnalité	Description	Priorité
1	Inscription	En tant qu'utilisateur, je souhaite pouvoir m'inscrire sur la plateforme .	Haute
2	Authentification	En tant qu'utilisateur, je souhaite pouvoir m'authentifier facilement sur la plateforme .	Haute
3.1	Réinitialisation de mot de passe par email	En tant qu'utilisateur, je souhaite avoir accès à un formulaire de réinitialisation de mot de passe pour pouvoir saisir mon adresse e-mail associée à mon compte .	Haute
3.2	Vérification de code	En tant qu'utilisateur, je souhaite pouvoir saisir le code de vérification reçu par e-mail et vérifier s'il est valide ou non pour récupérer mon compte .	Moyenne
3.3	Changement de mot de passe	En tant qu'utilisateur, je souhaite pouvoir changer mon mot de passe une fois le code de vérification validé .	Moyenne
4.1	consultation de profil	En tant qu'utilisateur, je souhaite pouvoir consulter mon profil .	Haute
4.2	Modification de profil	En tant qu'utilisateur, je souhaite pouvoir mettre à jour les informations de mon profil.	Moyenne
5.1	consultation des conversations	En tant qu'utilisateur, je veux pouvoir accéder à mes conversations .	Moyenne
5.2	Envoi de messages	En tant qu'utilisateur, je veux pouvoir envoyer un message à un autre utilisateur.	Moyenne

6.1	Creation des publications	En tant qu'utilisateur, je veux pouvoir créer des publications .	Haute
6.2	Explorer les publications	En tant qu'utilisateur, je veux pouvoir explorer les publications publiées sur la plate-forme.	Haute
6.3	Suppression des publications	En tant qu'utilisateur, je veux pouvoir supprimer mes propres publications .	Moyenne
6.4	Partage des publications	En tant qu'utilisateur, je veux pouvoir partager des publications .	Moyenne
6.5	Réaction aux publications	En tant qu'utilisateur, je veux pouvoir réagir aux publications .	Moyenne
6.6	Ajout de commentaires	En tant qu'utilisateur, je veux pouvoir ajouter des commentaires aux publications .	Moyenne
6.7	Suppression de commentaires	En tant qu'utilisateur, je veux pouvoir supprimer mes propres commentaires.	Faible
6.8	Consultation des commentaires	En tant qu'utilisateur, je veux pouvoir consulter les commentaires des autres utilisateurs.	Faible
7	Recommandation intelligente	En tant qu'utilisateur, je souhaite bénéficier d'une recommandation personnalisée basée sur mes préférences et mon historique de reaction, améliorant ainsi mon expérience.	Forte

Le Product Backlog est un outil dynamique, évoluant tout au long du projet. Il est constamment révisé et adapté en fonction des besoins changeants du produit, des retours des utilisateurs et des nouvelles découvertes. La gestion efficace du Product Backlog est essentielle pour la planification et l'exécution réussies des itérations ou des sprints dans le cadre de la méthodologie Scrum.

3.3.2 Planification des releases

Pour s'assurer que notre travail est bien organisé, il est essentiel d'organiser une réunion de planification une fois que le Backlog du Produit a été établi. Pour notre projet, nous avons

décidé de le diviser en deux releases distinctes, comme décrit ci-dessous dans la figure 3.3.

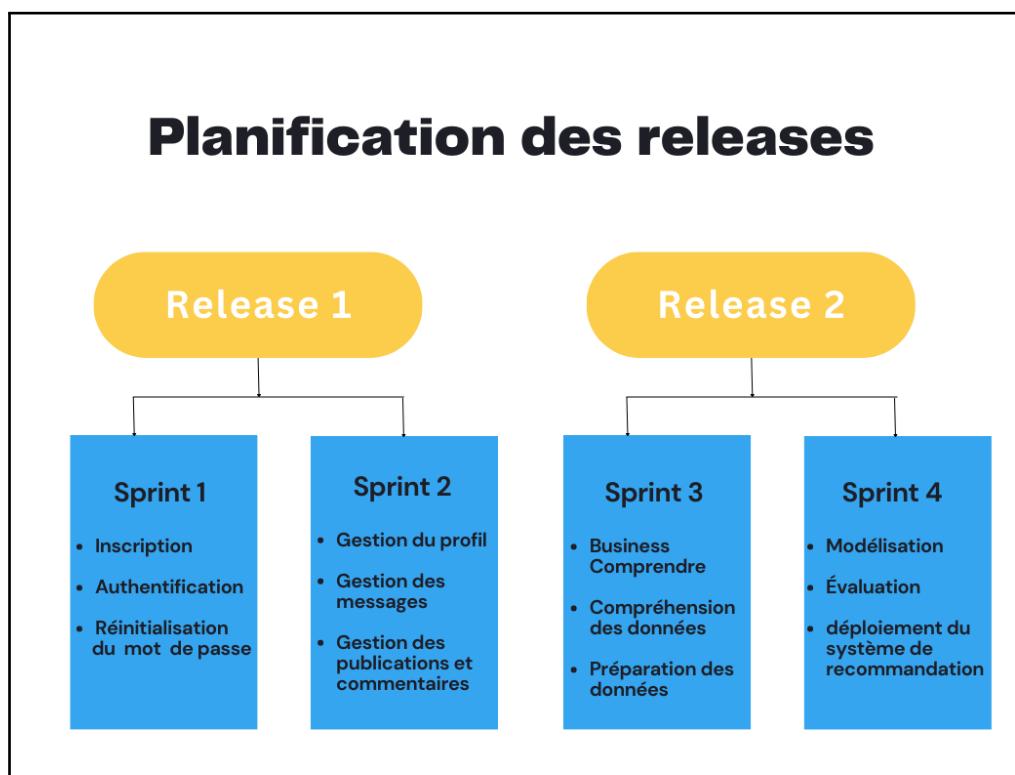


FIGURE 3.3 – Planification des releases

3.4 Architecture de la solution

Après l'identification des besoins et la planification du projet avec Scrum, parler de l'architecture de la solution s'impose. Nous commençons par l'architecture logique. Ensuite, nous dessinons le diagramme des paquetages global et nous terminons avec l'architecture physique de notre application.

3.4.1 Développement Full stack

Nous avons décidé de développer une application Full stack avec les technologies ReactJS et Flask.

Le développement Full stack fait référence au développement d'une application end-to-end, y compris la partie front-end et la partie back-end. Le front-end est présenté par les interfaces utilisateurs et le back-end se compose de la logique métier et des workflows de l'application. La figure suivante 3.4 illustre le Développement Full stack.

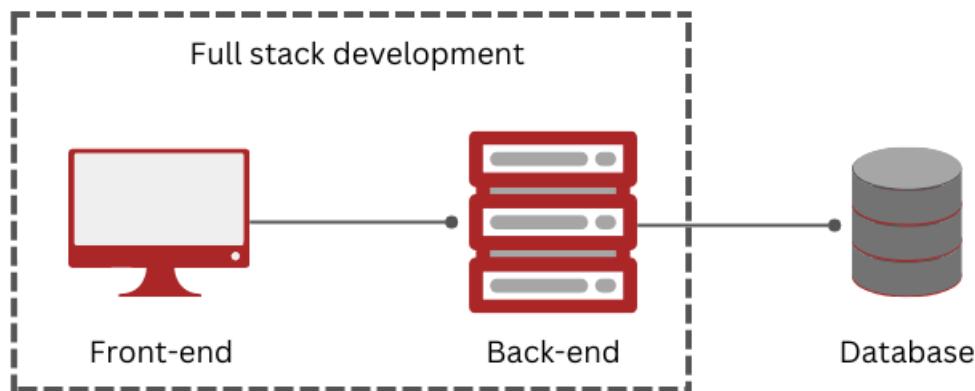


FIGURE 3.4 – Développement Full stack

3.4.2 Couche et niveau

Layers ou couches sont des séparations logiques du code. C'est une manière d'organiser le code d'une façon logique suivant leurs différentes fonctions.

De l'autre coté, Tiers ou niveaux sont les emplacements physique où les différentes layers sont déployés et s'exécutent.

3.4.3 Architecture logique du projet

Notre projet se compose d'un front-end avec ReactJs et d'un back-end avec flask :

- **Flask** : Est un framework web léger et flexible écrit en langage de programmation Python, Il est employé pour développer des applications web rapidement, offrant des fonctionnalités fondamentales telles que le routage, les requêtes HTTP et le rendu des templates. Le back-end est composé de 3 couches. Une couche présentation suivant le modèle MVC (modèle - vue - contrôleur). L'objectif global du MVC est de séparer les aspects traitement, données et présentation, et de définir les interactions entre ces trois aspects. Deux autres couches métier et accès aux données sont regroupées ensemble dans la couche logique métier [15].
- **ReactJS** : Est une bibliothèque JavaScript open-source développée par Facebook. Elle est principalement utilisée pour construire des interfaces utilisateur interactives et réactives. React permet de créer des composants réutilisables et modulaires qui se mettent à jour de manière efficace et efficace lorsque l'état des données change [16].

Le front-end et le back-end communiquent à travers les requêtes et les réponses à travers le protocole HTTP.

3.4.4 Diagramme de paquetages global

Notre système est composé principalement de trois paquetages : le Front-End, le Back-End et les procédures stockées, comme le montre la figure 3.5.

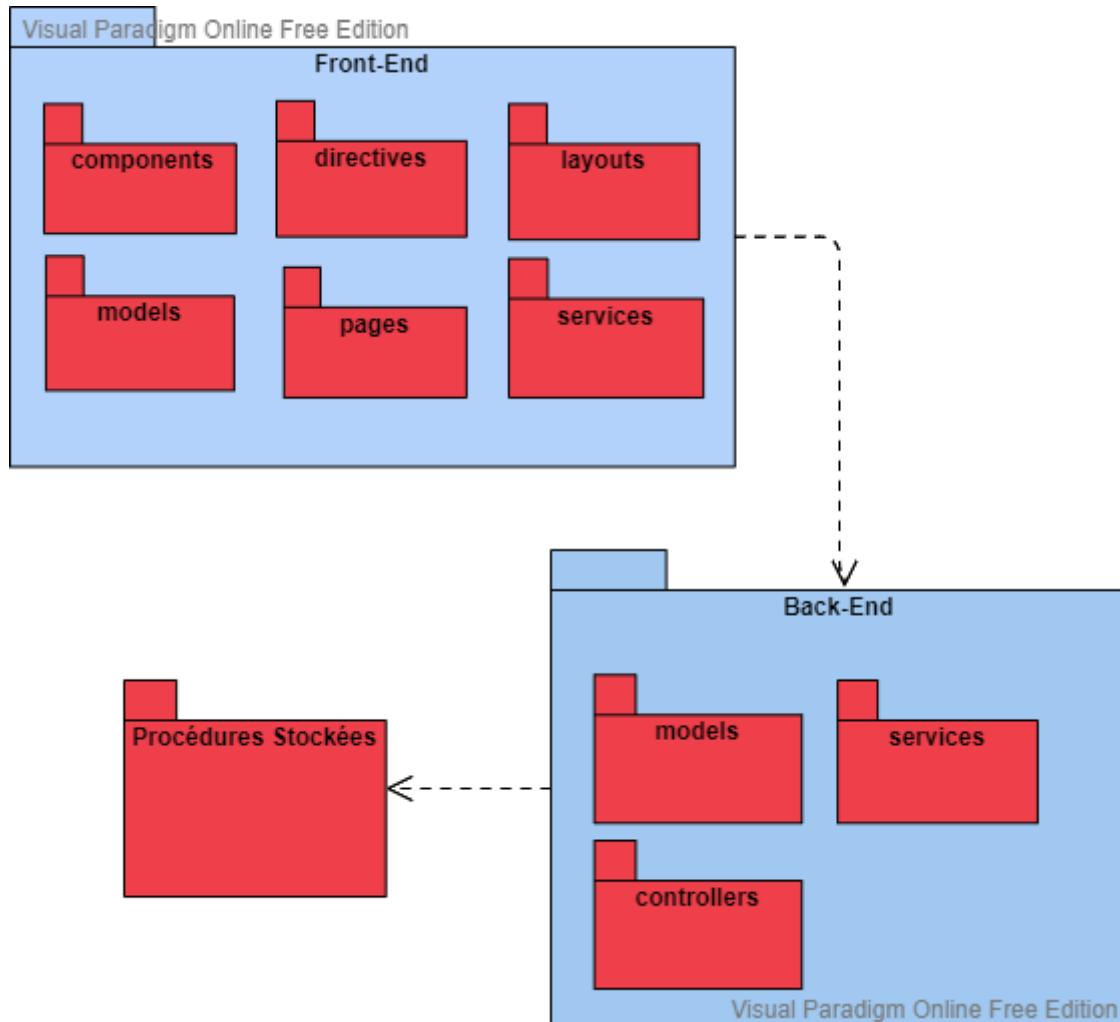


FIGURE 3.5 – Diagramme de Package Global

Nous expliquons les paquets :

1. Front-End :

- Components : Les composants sont les éléments de base de React. Ils encapsulent la logique et l'interface utilisateur d'une partie spécifique de l'application. Les composants peuvent être réutilisables et modulaires, ce qui facilite la construction d'interfaces utilisateur interactives.
- Pages : Dans React, les pages peuvent être considérées comme des composants spécifiques qui représentent des vues spécifiques de l'application. Elles peuvent regrouper plusieurs composants pour représenter une page complète.
- Services : Dans le contexte de React, les services font référence à des modules ou

des classes utilisées pour effectuer des opérations spécifiques, telles que l'appel à des API externes, la gestion des données, etc. Les services sont généralement utilisés pour séparer la logique métier de la logique de présentation dans les composants.

2. Back-End :

- Modèles (Models) : Les modèles dans Flask font référence à la couche de données de l'application. Ils représentent la structure des données et la logique de manipulation des données. Les modèles sont généralement définis à l'aide d'un ORM (Object-Relational Mapping) tel que Mongoengine.
- Services (Services) : Les services dans le backend Flask sont responsables du traitement des requêtes HTTP et de la manipulation des données. Ils contiennent la logique métier de l'application, interagissent avec les modèles et préparent les données à renvoyer en réponse. Les services sont généralement des classes ou des modules regroupant des fonctionnalités spécifiques.
- Contrôleurs (Controllers) : Flask suit le modèle de conception MVC (Modèle-Vue-Contrôleur), où les vues font office de contrôleurs. Les contrôleurs dans Flask gèrent la logique de l'application en coordonnant les actions entre les modèles et les vues. Ils agissent comme une couche intermédiaire entre les modèles et les vues pour traiter les actions de l'utilisateur.

3. Procédures Stockées

: MongoDB Atlas est un service de base de données cloud entièrement géré proposé par MongoDB. Il permet aux développeurs de déployer, de gérer et de scaler facilement des clusters MongoDB dans le cloud, sans avoir à se soucier de la gestion de l'infrastructure sous-jacente.

3.4.5 Architecture physique du projet

Dans cette partie, nous présentons l'architecture globale du projet afin d'expliquer la communication entre les différentes parties du système. Notre application adopte une architecture 3-tiers, avec une couche de présentation basée sur React, une couche métier utilisant Flask, et une couche d'accès aux données avec MongoDB. La Figure 3.6 montre l'architecture de notre application

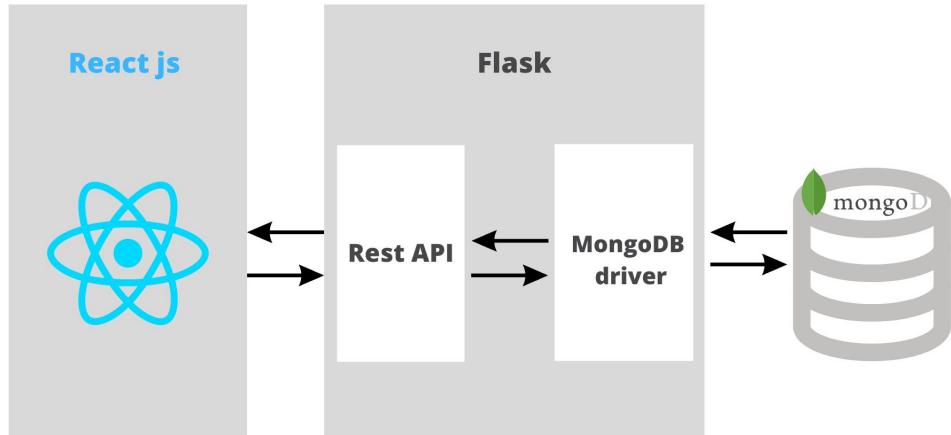


FIGURE 3.6 – Aperçu de l'architecture physique de la solution

3.5 Environnement de travail

Dans cette section, nous vous présentons notre environnement de travail logiciel et matériel durant ce stage.

3.5.1 Environnement matériel

Le portatif qui nous a été donnée par Pepolls comporte les caractéristiques suivantes :

TABLEAU 3.2 – Caractéristiques du Portatif

CPU	RAM	GPU	OS
Intel® Core™ i7-4610M Processor4M Cache, jusqu'à 3.70 GHz	16 Go	AMD Radeon™ HD 8670M	Windows 10

3.5.2 Environnement logiciel

Après avoir cité les caractéristiques du matériel, nous passons maintenant à l'environnement logiciel.

3.5.2.1 Éditeur de code

- **Visual Studio Code (VSCode) :** Éditeur de code source autonome qui s'exécute sur Windows, macOS et Linux. Il convient aux développeurs avec des extensions pour prendre

en charge à peu près n'importe quel langage de programmation. La figure 3.7 montre le logo de VsCode.



FIGURE 3.7 – Logo VsCode

- **Google Colab, ou Google Colaboratory :** Est un environnement de notebook Jupyter hébergé qui ne nécessite aucune configuration pour utiliser et fonctionne entièrement dans le cloud. Il peut être utilisé pour écrire et exécuter du code, principalement en Python, et est largement utilisé pour des travaux liés à l'apprentissage automatique, l'analyse de données et l'éducation. Vous pouvez exécuter des scripts, visualiser des données, partager vos résultats, et même accéder à des ressources matérielles puissantes (comme les GPU et TPU de Google) gratuitement. La figure 3.8 montre le logo de Google Colab.

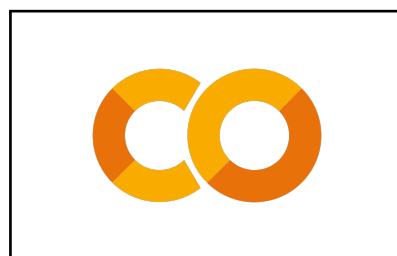


FIGURE 3.8 – Logo Google Colab

3.5.2.2 Client API

- **Thunder Client :** Est une extension pour Visual Studio Code qui sert d'outil de test d'API léger et rapide. Fonctionnant comme un client API, il fournit une interface conviviale pour créer et envoyer des requêtes HTTP/HTTPS. La figure 3.9 montre le logo de Thunder Client.

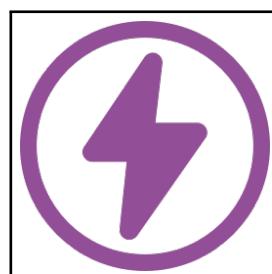


FIGURE 3.9 – Logo Thunder Client

3.5.2.3 Open API

- **Swagger :** Connue sous le nom d'OpenAPI, est un ensemble d'outils de spécification pour la conception, la construction, la documentation et le test d'API RESTful. Il fournit un moyen standard et langage-agnostique pour décrire la structure d'une API, permettant à la fois aux humains et aux machines de comprendre les capacités d'un service sans accès direct au code source. La figure 3.10 montre le logo de Swagger.

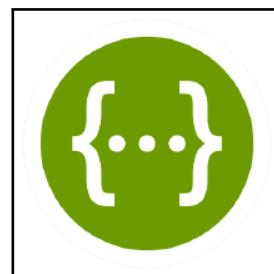


FIGURE 3.10 – Logo Swagger

3.5.2.4 Frameworks et Technologies de développement

- **ReactJS :** Est une bibliothèque JavaScript open-source développée par Facebook, utilisée pour construire des interfaces utilisateur interactives pour les applications web mono-page. La figure 3.11 montre le logo de ReactJs.

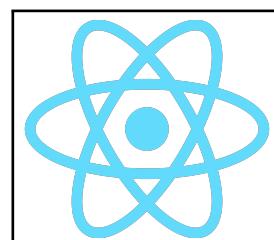


FIGURE 3.11 – Logo ReactJs

- **Flask :** Est un micro-framework web écrit en Python, reconnu pour sa simplicité et sa modularité. Il offre les outils de base nécessaires pour créer des applications web. La figure 3.12 montre le logo de Flask.



FIGURE 3.12 – Logo Flask

- **Bootstrap** : Boîte à outils frontale puissante, extensible et riche en fonctionnalités. Créer et personnaliser avec Sass, utiliser un système de grille et des composants pré-définis et donner vie aux projets avec de puissants plug-ins JavaScript. La figure 3.13 montre le logo de Bootstrap.



FIGURE 3.13 – Logo Bootstrap

- **Pusher** : Est une technologie de communication en temps réel pour les applications Web et mobiles, permettant des fonctionnalités telles que les notifications en direct et les mises à jour en temps réel. La figure 3.14 montre le logo de Pusher.



FIGURE 3.14 – Logo Pusher

- **TensorFlow** : Est une plateforme open-source dédiée à l'apprentissage profond (deep learning) qui permet de construire et d'entraîner des réseaux neuronaux. La figure 3.15 montre le logo de TensorFlow.



FIGURE 3.15 – Logo TensorFlow

3.5.2.5 Langages de programmation

- **Python 3** : Est la dernière version majeure du langage de programmation Python, caractérisé par sa syntaxe claire et lisible qui favorise la productivité des développeurs. Il est utilisé pour une grande variété d'applications, y compris le développement web,

l'analyse de données, l'apprentissage automatique, l'automatisation, et bien plus encore. La figure 3.16 montre le logo de Python.

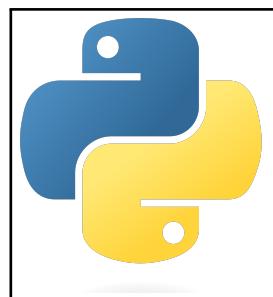


FIGURE 3.16 – Logo Python

- **JavaScript** : Est un langage de programmation de haut niveau, interprété, et principalement connu comme le langage de script pour les pages Web. Il permet d'ajouter des fonctionnalités interactives aux sites Web et est également utilisé côté serveur, notamment avec NodeJs. La figure 3.17 montre le logo de JavaScript.

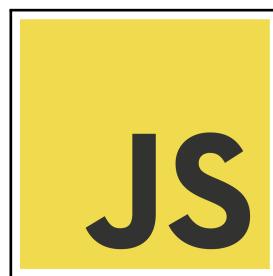


FIGURE 3.17 – Logo JavaScript

3.5.2.6 Base de donnée

- **MongoDB** : Est une base de données NoSQL orientée document qui stocke les données sous forme de documents BSON, une représentation binaire de JSON. La figure 3.18 montre le logo de MongoDB.



FIGURE 3.18 – Logo MongoDB

3.5.2.7 Outils DevOps

- **Git** : Est un système de contrôle de version distribué gratuit et open source conçu pour tout gérer, des petits aux très grands projets, avec rapidité et efficacité. La figure 3.19 montre le logo de Git.



FIGURE 3.19 – Logo Git

- **Gitlab** : Est un répertoire GIT web-based. Il permet de planifier des projets et de gérer le code source, l'approche DevOps CI/CD, et la sécurité [17]. En plus, il permet aux équipes de collaborer et de développer de meilleurs logiciels. La figure 3.20 montre le logo de GitLab.



FIGURE 3.20 – Logo GitLab

- **ClickUp** : Est une plateforme de gestion de projet conçue pour aider les équipes à rationaliser leur flux de travail et à augmenter leur productivité. Elle propose une gamme de fonctionnalités pour la gestion des tâches, la collaboration et le suivi des objectifs. La figure 3.21 montre le logo de ClickUp.



FIGURE 3.21 – Logo ClickUp

3.5.2.8 Outils de modélisation

- **Draw.io** : Est une application web de création de diagrammes et de schémas, offrant des fonctionnalités pour concevoir des organigrammes, des diagrammes de flux, des cartes mentales et bien d'autres. La Figure 3.22 montre le logo de Draw.io.

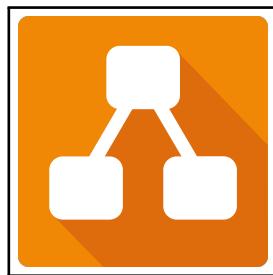


FIGURE 3.22 – Logo Draw.io

Conclusion

Les grandes lignes de ce chapitre sont l'identification des besoins, la réalisation d'une étude analytique, la planification du projet avec Scrum, l'élaboration de l'architecture de la solution et la présentation de l'environnement de travail. Nous terminons le chapitre de Planification et Architecture et nous détaillerons, dans les prochains chapitres, nos releases.

CHAPITRE 4

RELEASE 1

Introduction

Dans ce chapitre, nous allons examiner en détail les travaux effectués lors de la **première Release**. Cette release correspond à la phase de développement de la plateforme. Ce première release comprend deux sprints :

- **Sprint 1** : Authentification , Inscription , Réinitialisation de mot de passe.
- **Sprint 2** : Gestion de profil , messages , publications et commentaires .

Le développement de chaque sprint passe par les étapes suivantes : analyse, conception et réalisation.

4.1 Développement de sprint 1

Le premier sprint est planifié sur une période de trois semaines et se compose de trois éléments :

- **Authentification**
- **Inscription**
- **Réinitialisation de mot de passe**

Le tableau 4.1 illustre le Backlog du premier sprint.

TABLEAU 4.1 – Backlog du premier sprint

ID	Fonctionnalité	description	Priorité
1	Inscription	En tant qu'utilisateur, je souhaite pouvoir m'inscrire .	Haute
2	Authentification	En tant qu'utilisateur, je souhaite pouvoir m'authentifier facilement .	Haute
3	Réinitialisation de mot de passe par email	En tant qu'utilisateur, j'ai besoin d'accéder à un formulaire de réinitialisation de mot de passe afin de pouvoir saisir mon adresse e-mail liée à mon compte. Une fois que j'aurai reçu un code de vérification par e-mail, je pourrai le saisir et, après validation, changer mon mot de passe en toute sécurité.	Haute

4.1.1 Analyse et conception

Nous présentons les différents diagrammes de sprint 1.

4.1.1.1 Diagramme de cas d'utilisation du sprint 1

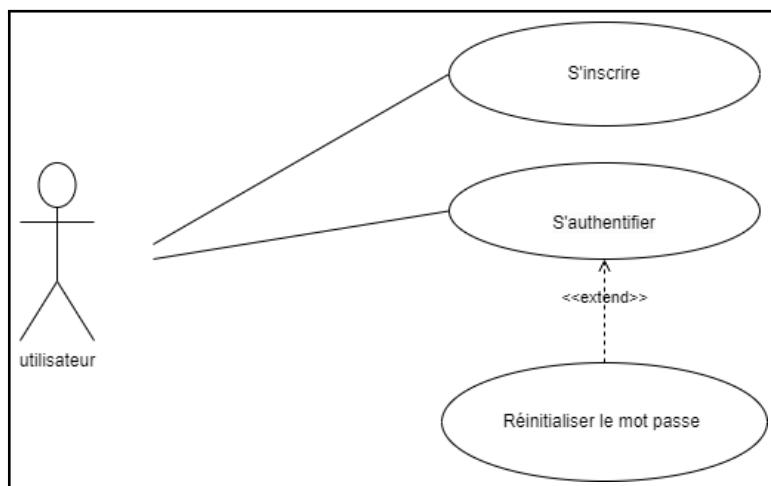


FIGURE 4.1 – Diagramme de cas d'utilisation du sprint 1

Le diagramme d'utilisation de la figure 4.1 représente les actions qu'un utilisateur peut effectuer sur la plateforme. Initialement, l'utilisateur a la possibilité de s'inscrire en créant un compte sur la plateforme. Une fois inscrit, l'utilisateur peut se connecter à son compte en utilisant ses identifiants de connexion. En cas d'oubli de son mot de passe, l'utilisateur a la possibilité de réinitialiser son mot de passe en suivant une procédure spécifique.

4.1.1.2 Diagrammes de séquence

Dans cette partie, nous aborderons les diagrammes de séquence du premier sprint.

Diagramme de séquence du cas d'utilisation «S'inscrire»

Le diagramme de séquence représenté dans la Figure 4.2 détaille le processus d'inscription d'un utilisateur. Au début, l'utilisateur accède à la page "Inscription" et entre son adresse e-mail et son mot de passe. Le système procède ensuite à la vérification de l'existence de l'adresse e-mail dans la base de données et à la validation du mot de passe. Si les informations fournies sont valides, le système génère un token pour l'utilisateur et crypte le mot de passe. Ensuite, l'utilisateur est redirigé vers le formulaire d'inscription 2, où il fournit les informations supplémentaires nécessaires. Une fois que toutes les informations sont validées,

le système enregistre les données de l'utilisateur, le redirige vers la page d'authentification et affiche un message de confirmation de son inscription.

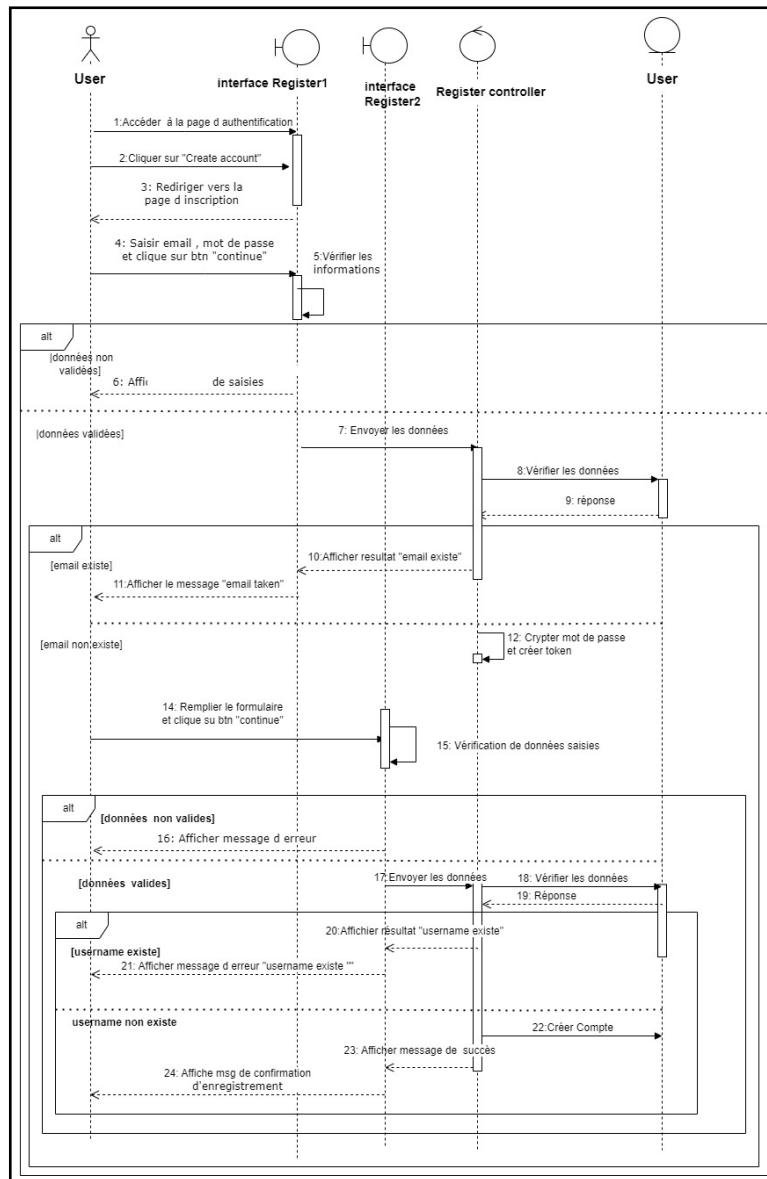


FIGURE 4.2 – Diagramme de séquence du cas d'utilisation «S'inscrire»

Diagramme de séquence du cas d'utilisation «S'authentifier»

Le diagramme de séquence représenté dans la Figure 4.3 illustre le processus d'authentification de l'utilisateur sur la plateforme. L'utilisateur accède à la page d'authentification et fournit son adresse e-mail et son mot de passe. Le système vérifie ensuite la validité de ces informations en les comparant avec celles stockées dans la base de données. Si l'adresse e-mail ou le mot de passe fourni n'est pas valide, le système affiche un message d'erreur approprié pour informer l'utilisateur. En revanche, si les informations d'authentification sont valides, l'utilisateur est redirigé vers la page d'accueil de la plateforme, où il peut accéder aux fonctionnalités et aux contenus réservés aux utilisateurs authentifiés.

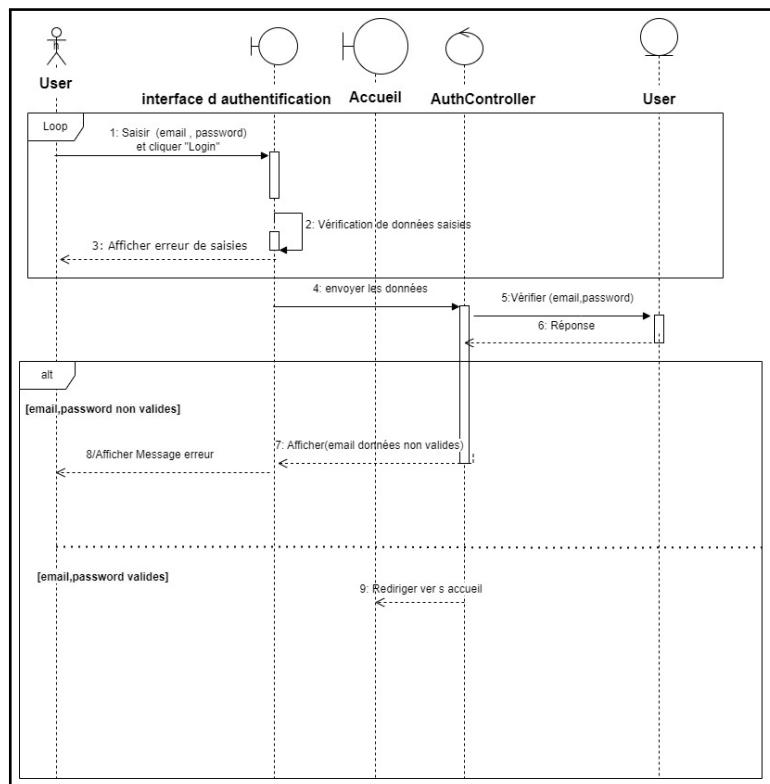


FIGURE 4.3 – Diagramme de séquence du cas d'utilisation «S'authentifier»

Diagramme de séquence du cas d'utilisation «Réinitialiser de mot de passe»

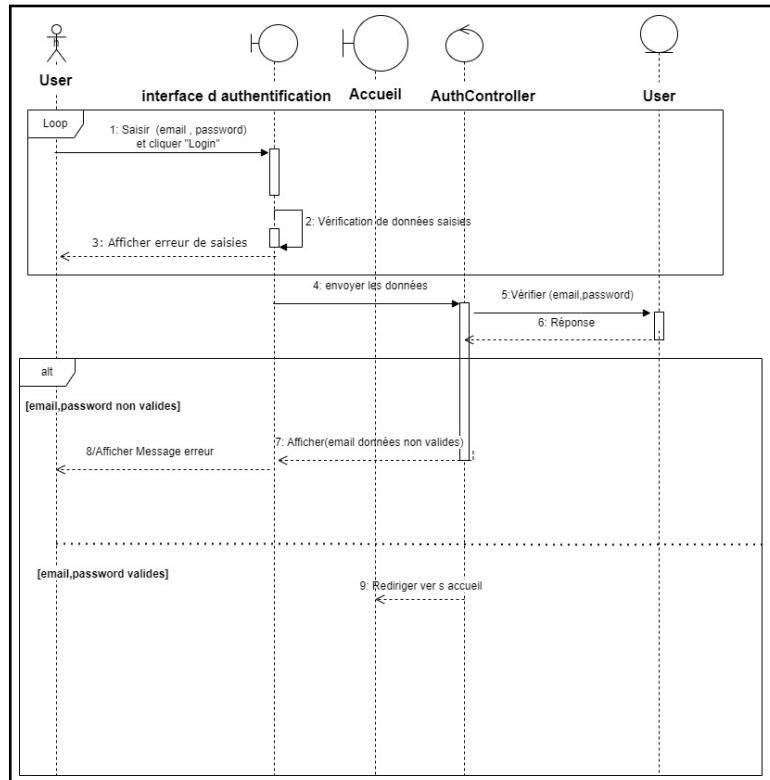


FIGURE 4.4 – Diagramme de séquence du cas d'utilisation «Réinitialiser de mot de passe»

Lors du processus de réinitialisation de mot de passe, l'utilisateur saisit son adresse e-mail sur la page dédiée, soumet le formulaire, et reçoit un code de vérification par e-mail. En entrant le code de vérification, le système vérifie sa validité. Si l'adresse e-mail n'est pas trouvée ou si le code de vérification est incorrect, un message d'erreur s'affiche. Cependant, si le code est valide, l'utilisateur peut entrer un nouveau mot de passe. Une fois que le nouveau mot de passe est confirmé, le système l'enregistre avec succès. la Figure 4.4 illustre le processus de réinitialisation du mot de passe.

4.1.2 Réalisation

Après avoir terminé la phase d'analyse et de conception de ce sprint, nous passons maintenant à la phase de réalisation.

4.1.2.1 Interface d'inscription

Nous présentons les interfaces de registration de l'utilisateur :

The screenshot shows a registration form titled "Create a new account". It includes fields for email ("djebbiofahmi@gmail.com"), password ("*****"), and password confirmation ("Verify your new password"). A blue "Continue" button is at the bottom, and a link "Already have an account ? Login" is below it. The Pepolls logo is in the top left corner.

FIGURE 4.5 – L'interface initiale d'inscription

The screenshot shows a continuation of the registration process. It asks "Let's get to know each other" and displays a placeholder profile picture with the text "upload your profile pictures". Below are fields for first name ("Fahmi"), last name ("DIOBBI"), middle name ("Djo"), date of birth ("13-08-1997"), and gender ("Male"). A blue "Continue" button is at the bottom, and a link "Already have an account ? Login" is below it. The Pepolls logo is in the top left corner.

FIGURE 4.6 – La deuxième interface d'inscription

La Figure 4.5 présente l'interface initiale de l'enregistrement où l'utilisateur peut saisir son adresse e-mail et son mot de passe. Cette interface permet à l'utilisateur de créer un compte en fournissant les informations nécessaires pour l'authentification et l'accès au système. Ensuite, la Figure 4.6 présente la deuxième interface d'inscription, où l'utilisateur est obligé d'ajouter des informations complémentaires dans les champs. Cette interface permet à l'utilisateur de fournir des détails supplémentaires, tels que son nom, prénom, date de naissance, etc.

4.1.2.2 Interface de l'authentification

La Figure 4.7 illustre l'interface d'authentification où l'utilisateur saisit son adresse e-mail et son mot de passe pour accéder à la plateforme.

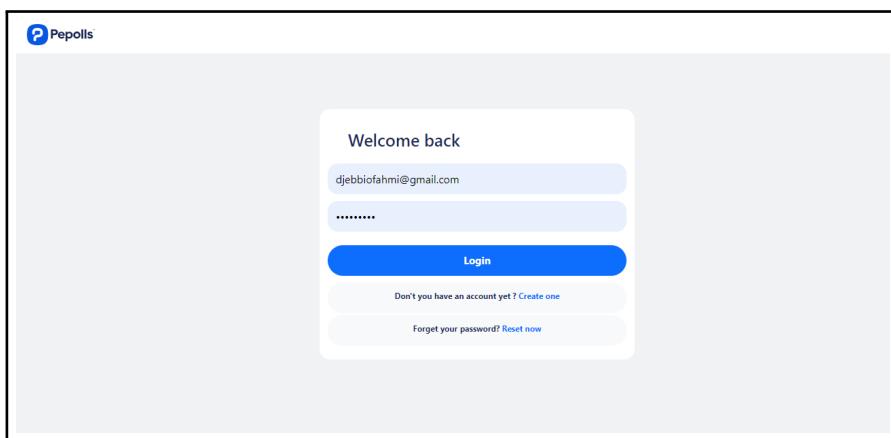
A screenshot of a web-based login form. At the top left is the Pepolls logo. The main area has a light gray background. In the center, there is a white rectangular card with rounded corners. At the top of the card, the text "Welcome back" is displayed. Below this, there are two input fields: the first contains the email address "djebbiofahmi@gmail.com" and the second contains a masked password ".....". Below the inputs is a large blue rectangular button with the word "Login" in white. Underneath the button, there is small text: "Don't you have an account yet? [Create one](#)" and "Forgot your password? [Reset now](#)".

FIGURE 4.7 – Interface de l'authentification

4.1.2.3 Interfaces de réinitialisation de mots de passe

Les interfaces suivantes sont destinées à la réinitialisation de mots de passe. La Figure 4.8 représente l'interface où l'utilisateur ajoute son adresse e-mail pour récupérer le mot de passe. La Figure 4.9 présente l'interface de vérification du code envoyé par e-mail.

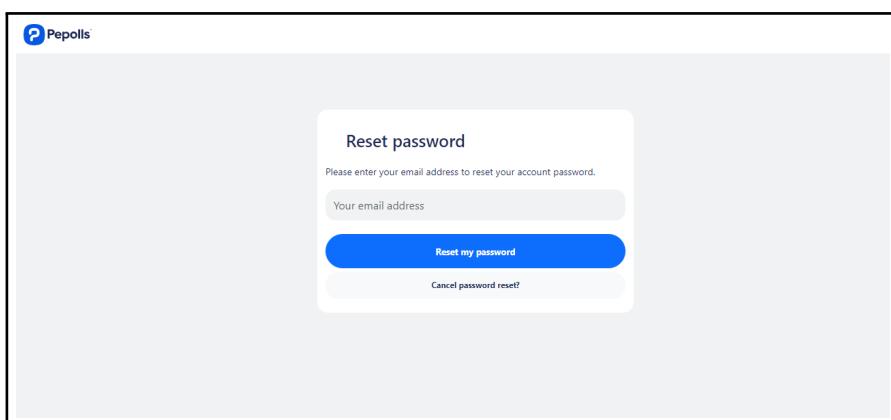
A screenshot of a password reset form. At the top left is the Pepolls logo. The main area has a light gray background. In the center, there is a white rectangular card with rounded corners. At the top of the card, the text "Reset password" is displayed. Below this, there is a small instruction: "Please enter your email address to reset your account password.". There is a single input field labeled "Your email address". Below the input field is a large blue rectangular button with the words "Reset my password" in white. At the bottom of the card, there is a link "Cancel password reset?".

FIGURE 4.8 – Interface de réinitialisation de mot de passe

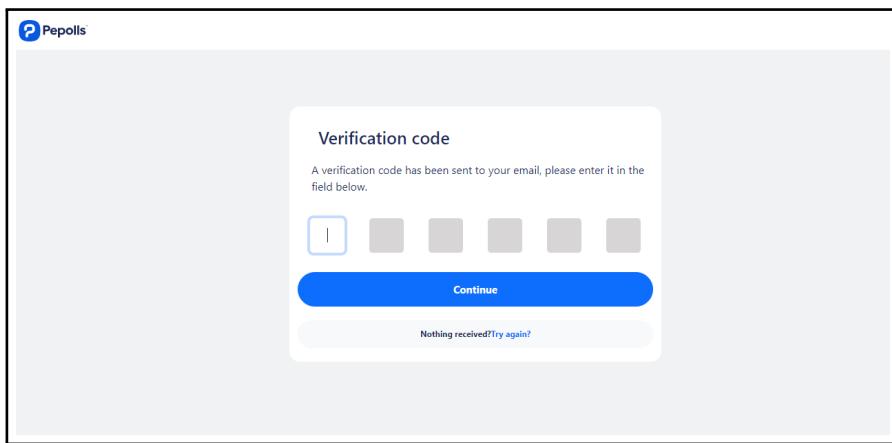


FIGURE 4.9 – Interface de vérification code

4.2 Développement du sprint 2

Le deuxième sprint s'étale sur une période de trois semaines et se décompose en 3 éléments :

- **Gestion du profil .**
- **Gestion des messages .**
- **Gestion des publications et commentaires .**

Le tableau 4.2 présente le Backlog du deuxième sprint.

TABLEAU 4.2 – Backlog du deuxième Sprint

ID	Fonctionnalité	Description	Priorité
4.1	consultation de profil	En tant qu'utilisateur, je souhaite pouvoir consulter mon profil .	Haute
4.2	Modification de profil	En tant qu'utilisateur, je souhaite pouvoir mettre à jour les informations de mon profil.	Moyenne
5.1	consultation des conversations	En tant qu'utilisateur, je veux pouvoir accéder à mes conversations .	Moyenne
5.2	Envoi de messages	En tant qu'utilisateur, je veux pouvoir envoyer un message à un autre utilisateur.	Moyenne
6.1	Creation des publications	En tant qu'utilisateur, je veux pouvoir créer des publications .	Haute

6.2	Explorer les publications	En tant qu'utilisateur, je veux pouvoir explorer les publications publiées sur la plate-forme.	Haute
6.3	Suppression des publications	En tant qu'utilisateur, je veux pouvoir supprimer mes propres publications .	Moyenne
6.4	Partage des publications	En tant qu'utilisateur, je veux pouvoir partager des publications .	Moyenne
6.5	Réaction aux publications	En tant qu'utilisateur, je veux pouvoir réagir aux publications .	Moyenne
6.6	Ajout de commentaires	En tant qu'utilisateur, je veux pouvoir ajouter des commentaires aux publications .	Moyenne
6.7	Suppression de commentaires	En tant qu'utilisateur, je veux pouvoir supprimer mes propres commentaires.	Faible
6.8	Consultation des commentaires	En tant qu'utilisateur, je veux pouvoir consulter les commentaires des autres utilisateurs.	Faible

4.2.1 Analyse et conception

Dans cette partie, nous allons présenter l'analyse et la conception du deuxième sprint.

4.2.1.1 Diagrammes de cas d'utilisation du sprint 2

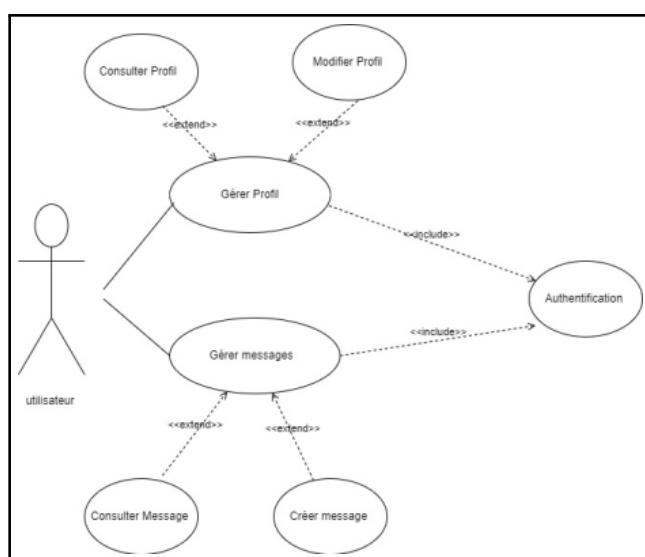


FIGURE 4.10 – Diagramme de cas d'utilisation du Gestion profil et messages

Le diagramme de cas d'utilisation de gestion de profil et de messagerie dans la Figure 4.10 représente les fonctionnalités disponibles pour l'utilisateur enregistré. Il lui permet de consulter et mettre à jour son profil en accédant aux informations qui y sont associées, telles que son nom, son adresse e-mail ou sa photo de profil. De plus, l'utilisateur enregistré peut accéder à ses conversations pour consulter les messages échangés avec d'autres utilisateurs et envoyer des messages à ces derniers. Ainsi, la plateforme facilite la communication et l'interaction entre les utilisateurs grâce à la fonctionnalité de messagerie intégrée.

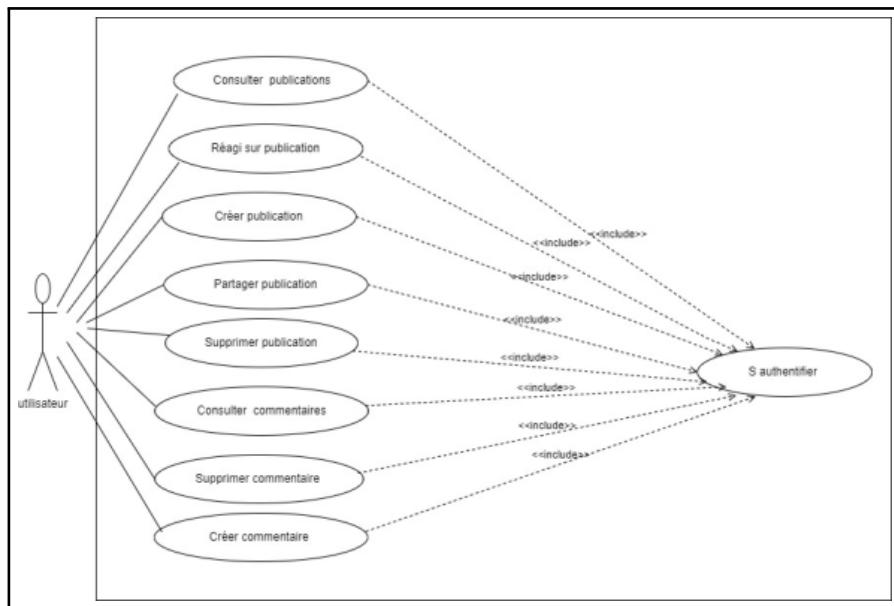


FIGURE 4.11 – Diagramme de cas d'utilisation du Gestion publications et commentaires

Le diagramme de cas d'utilisation de gestion des publications et des commentaires dans la Figure 4.11 offre plusieurs fonctionnalités à l'utilisateur enregistré. Tout d'abord, il permet à l'utilisateur de créer des publications, lui donnant ainsi la possibilité de partager du contenu sur la plateforme. Ensuite, l'utilisateur peut explorer les publications publiées par d'autres utilisateurs, lui permettant ainsi de découvrir et d'interagir avec différents contenus. L'utilisateur a également la possibilité de supprimer ses propres publications, offrant un contrôle sur le contenu partagé. De plus, l'utilisateur peut partager des publications, permettant ainsi de les diffuser à un plus large public. En ce qui concerne les réactions, l'utilisateur peut réagir aux publications en exprimant son appréciation, son soutien ou son désaccord. De plus, il peut ajouter des commentaires aux publications, permettant ainsi d'engager des discussions et d'interagir avec d'autres utilisateurs. L'utilisateur peut également supprimer ses propres commentaires et consulter les commentaires des autres utilisateurs, favorisant ainsi l'interaction et l'échange au sein de la plateforme.

4.2.1.2 Diagrammes de séquence

Dans cette partie, nous allons présenter les diagrammes de séquence du deuxième sprint.

Diagramme de séquence de la consultation de profil

Dans le diagramme de séquence 4.12 lorsque l'utilisateur clique sur le bouton "Edit Profil", le système envoie une requête pour afficher les données utilisateur. Le système répond avec un code de succès et récupère les informations du profil de l'utilisateur. En conséquence, l'interface de profil de l'utilisateur est affichée, permettant à l'utilisateur de consulter facilement ses informations personnelles.

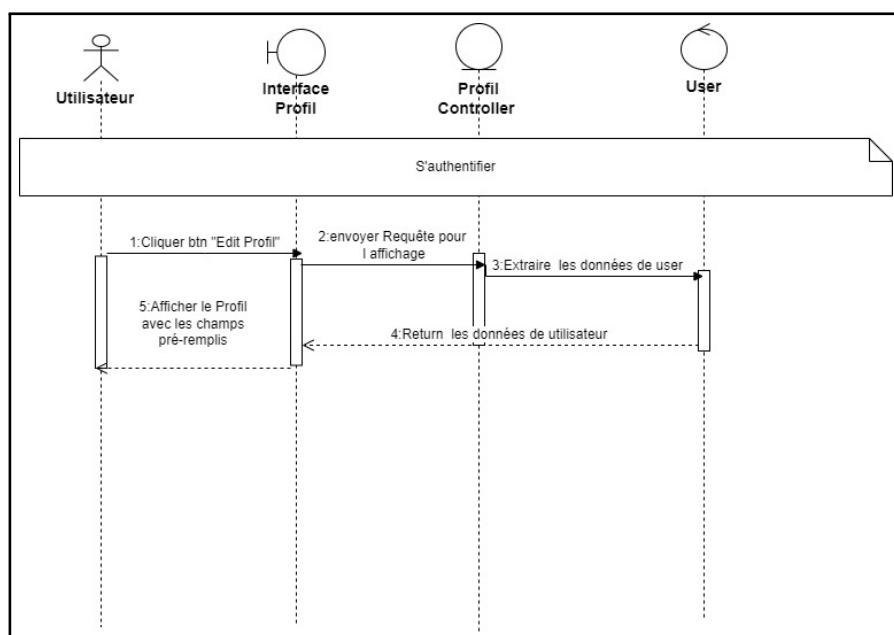


FIGURE 4.12 – Diagramme de séquence du cas d'utilisation «Consulter Profil»

Diagramme de séquence de la modification de profil

Le diagramme de séquence 4.13 représente le processus de modification du profil de l'utilisateur. Lorsque l'utilisateur accède à la page de modification de profil, le système affiche le formulaire avec les champs pré-remplis contenant les informations actuelles de l'utilisateur. Ensuite, l'utilisateur modifie les informations qu'il souhaite changer dans le formulaire. Après avoir effectué les modifications, l'utilisateur valide les changements. Le système enregistre alors les modifications apportées au profil de l'utilisateur. Enfin, le système affiche un message de confirmation indiquant que la modification du profil a été effectuée avec succès.

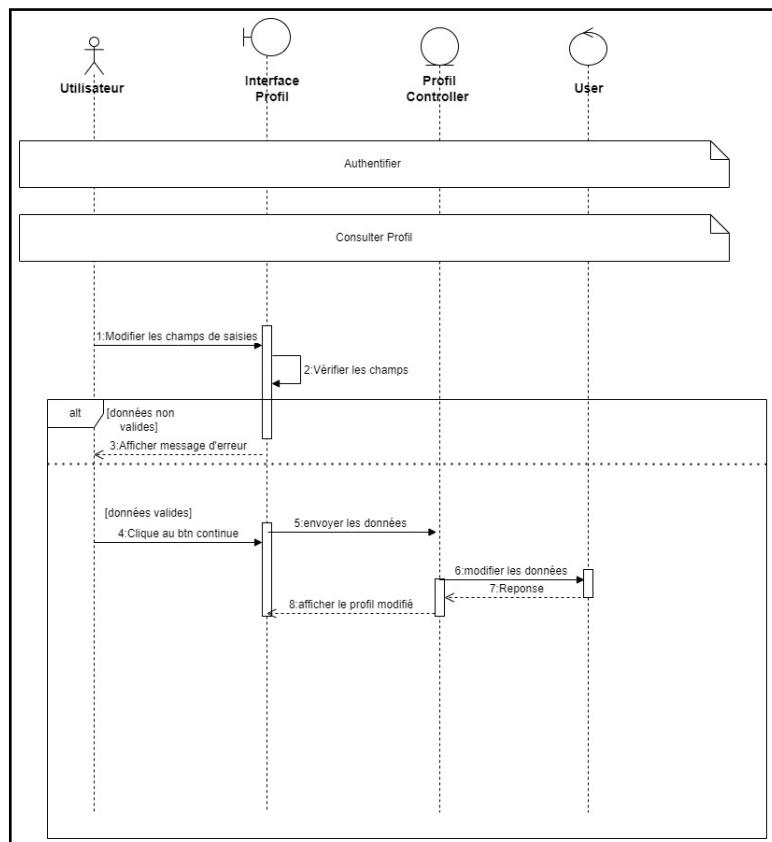


FIGURE 4.13 – Diagramme de séquence du cas d'utilisation «Modifier Profil»

Diagramme de séquence de la gestion des messages

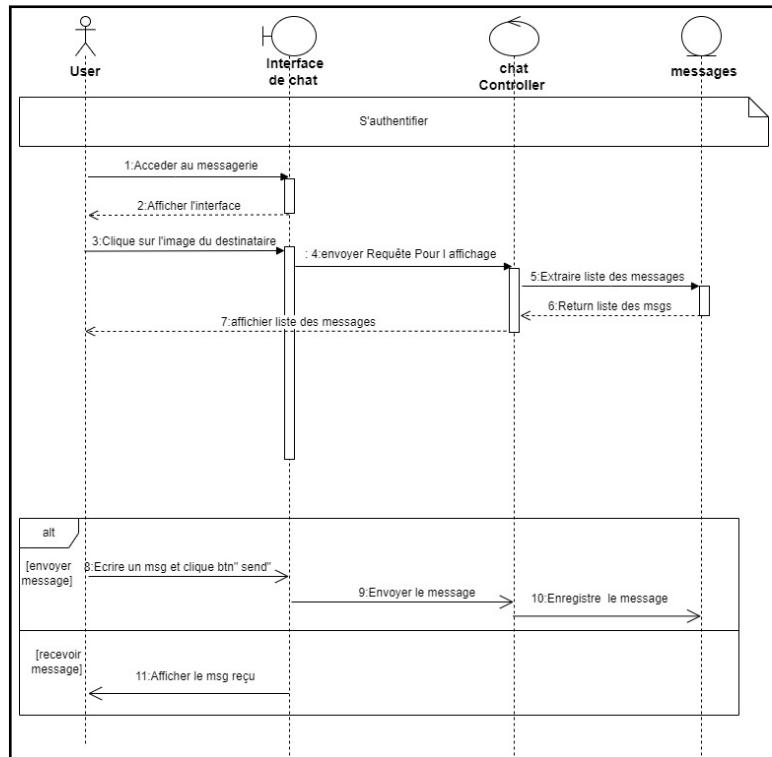


FIGURE 4.14 – Diagramme de séquence du cas d'utilisation «Gérer messages»

Le diagramme de séquence représenté dans la Figure 4.14 décrit le processus de gestion des messages. L'utilisateur accède d'abord à la messagerie de la plateforme, où il peut consulter les conversations existantes. En cliquant sur l'image du destinataire, l'utilisateur ouvre la conversation correspondante. Il peut ensuite saisir son message dans la zone de texte prévue à cet effet. Une fois le message rédigé, l'utilisateur clique sur le bouton "send" pour l'envoyer. Parallèlement, le système permet également à l'utilisateur de recevoir les messages en envoyant une requête d'affichage.

Diagramme de séquence de la création de publication

Le diagramme de séquence représenté dans la Figure 4.15 décrit le processus de la création de publication. L'utilisateur saisit les informations de la publication telles que le sujet, le texte et éventuellement une image. Une fois que toutes les informations nécessaires sont fournies, l'utilisateur valide la création de la publication. Le système enregistre alors la publication dans la base de données et affiche un message de confirmation à l'utilisateur. Dans le scénario alternatif, si l'utilisateur omet de remplir certains champs obligatoires, le système affiche un message d'erreur indiquant les champs manquants.

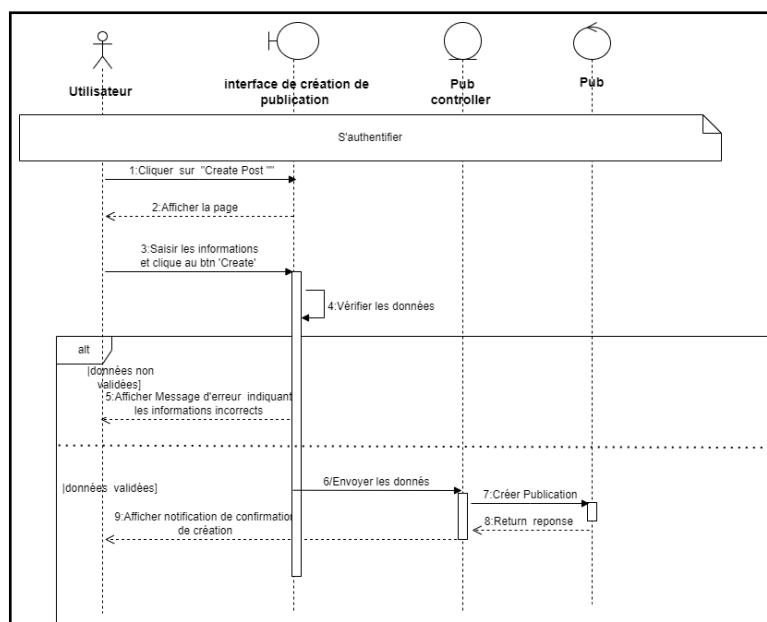


FIGURE 4.15 – Diagramme de séquence du cas d'utilisation «Créer publication»

Diagramme de séquence de l'exploration de publications

Le diagramme de séquence de l'exploration de publications dans la Figure 4.16 illustre le processus par lequel le système affiche la liste des publications disponibles. Initialement, l'utilisateur accède à la page d'exploration de publications et envoie une requête pour récupérer

les publications. Le système répond en extrayant les publications de la base de données et les affiche à l'utilisateur. L'utilisateur peut ensuite parcourir la liste des publications et visualiser les détails de chaque publication.

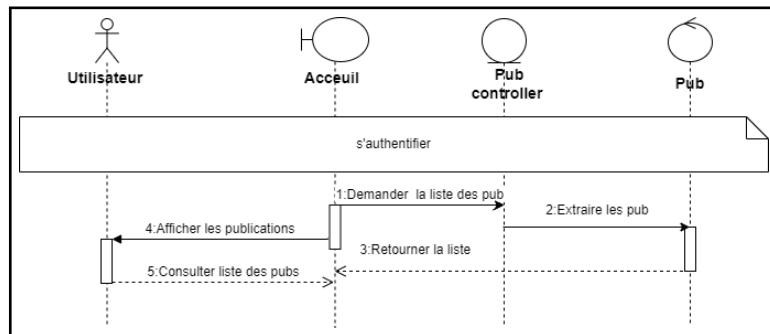


FIGURE 4.16 – Diagramme de séquence du cas d'utilisation «Explorer publication»

Diagramme de séquence du partage de publication

Le diagramme de séquence du partage de publication représente le processus par lequel un utilisateur sélectionne une publication à partager sur la plateforme. Initialement, l'utilisateur accède à la publication qu'il souhaite partager et envoie une requête de partage au système. Le système vérifie les autorisations de partage de l'utilisateur et affiche un message de confirmation indiquant que la publication a été partagée avec succès. Ce diagramme de séquence dans la figure 4.17 met en évidence les étapes clés du processus de partage de publication, permettant à l'utilisateur de diffuser du contenu auprès d'autres utilisateurs sur la plateforme.

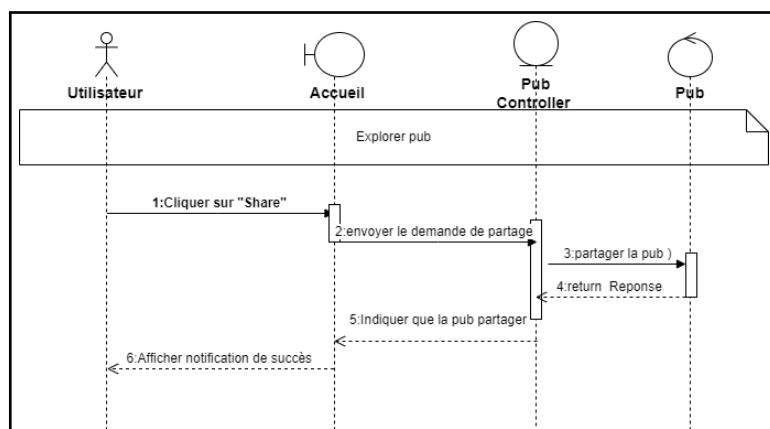


FIGURE 4.17 – Diagramme de séquence du cas d'utilisation «Partager publication»

Diagramme de séquence de la gestion des commentaires

Le diagramme de séquence de la gestion des commentaires dans la figure 4.18 illustre le processus par lequel un utilisateur peut ajouter et supprimer des commentaires. Initialement,

l'utilisateur accède à la page des commentaires et soumet son commentaire en cliquant sur le bouton "add comment". Le système vérifie le commentaire et l'ajoute à la page ou à l'élément correspondant, renvoyant un message de confirmation avec un statut 200 OK(un code de réponse HTTP qui indique que la requête du client a été traitée avec succès par le serveur.) Pour supprimer un commentaire, l'utilisateur accède à la page des commentaires, localise son commentaire et clique sur le bouton "x" pour le supprimer. Le système supprime le commentaire de la page ou de l'élément associé, renvoyant un message de confirmation avec un code de succès par le serveur. En cas d'erreur lors de l'ajout ou de la suppression du commentaire, le système renvoie un message d'erreur avec un statut 500(un code de réponse HTTP) pour indiquer un problème de traitement de la requête.

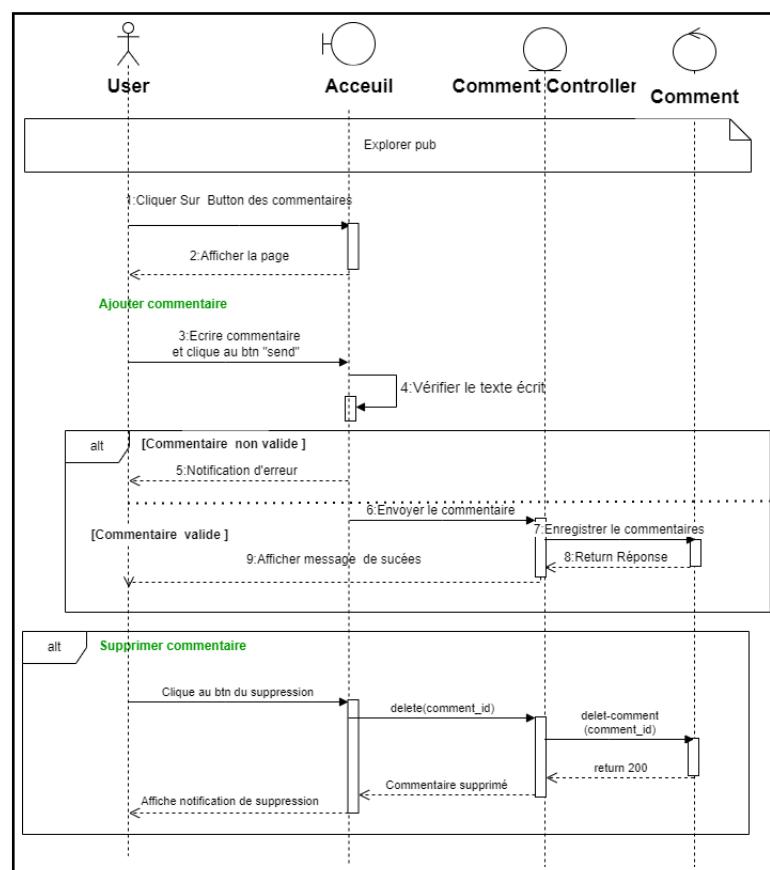


FIGURE 4.18 – Diagramme de séquence du cas d'utilisation «Gérer commentaires»

Diagramme de séquence pour l'action "Réagir à une publication"

Lorsqu'un utilisateur interagit avec une publication sur notre plateforme en sélectionnant une réaction (like, dislike, trend), une requête est envoyée au serveur. Le serveur enregistre alors la réaction dans la base de données et renvoie une réponse de succès pour confirmer l'enregistrement. Le compteur de réactions correspondant à la publication est mis à jour, augmentant de 1 pour la réaction sélectionnée. L'utilisateur a la possibilité d'annuler sa réaction

ou de la modifier ultérieurement. En cas d'annulation, le serveur supprime la réaction de la base de données et ajuste le compteur de réactions en conséquence. Si l'utilisateur souhaite modifier sa réaction, le serveur met à jour la réaction dans la base de données et ajuste les compteurs de réactions précédentes et actuelles. Ainsi, l'utilisateur peut réagir, annuler ou modifier sa réaction sur une publication. Ce diagramme de séquence dans la figure 4.19 met en évidence les étapes clés du processus de réaction à une publication.

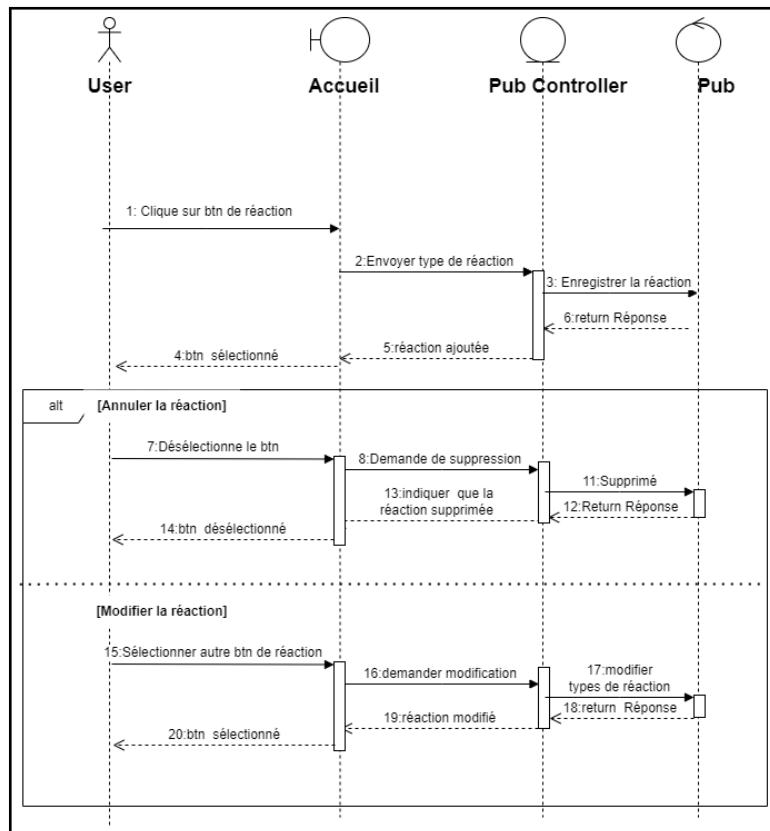


FIGURE 4.19 – Diagramme de séquence du cas d'utilisation «Réagir sur publication»

4.2.2 Réalisation

Nous entamons maintenant la phase de réalisation après avoir achevé l'analyse et la conception de ce sprint.

4.2.2.1 Interface d'édition de profil

L'interface de modification du profil de l'utilisateur est illustrée dans la Figure 4.20. Cette interface permet à l'utilisateur de mettre à jour ses informations personnelles, telles que son nom, son prénom, etc. Elle offre une interface conviviale pour que l'utilisateur puisse facilement effectuer des modifications sur son profil.

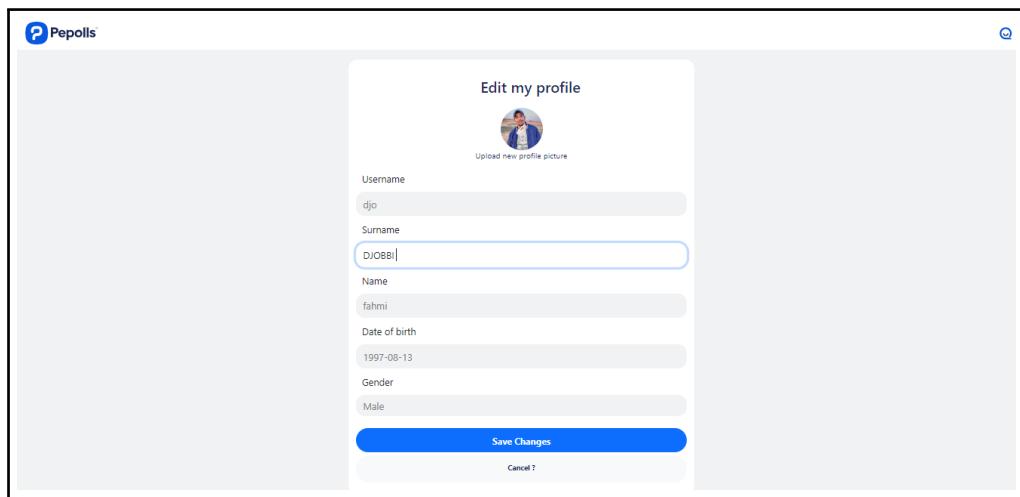


FIGURE 4.20 – Interface d'édition de profil

4.2.2.2 Interface de messagerie "Talk"

La Figure 4.21 présente l'interface de messagerie de l'onglet "Talk" de notre réseau social. Cette interface permet aux utilisateurs de communiquer entre eux en envoyant des messages en temps réel. Elle offre aux utilisateurs une expérience pratique et conviviale pour rester connectés et échanger facilement au sein de notre réseau social.

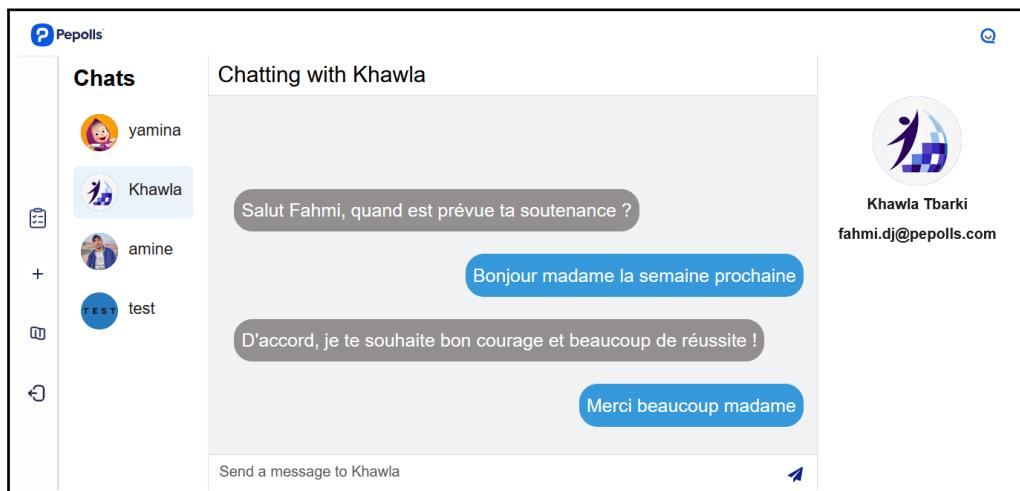


FIGURE 4.21 – Interface d'édition de profil

4.2.2.3 Interfaces d'ajout de publication

La Figure 4.22 présente l'interface d'ajout de publication sur Pepolls. Cette interface permet aux utilisateurs de partager leurs idées en créant un post. Ils peuvent inclure une image, une légende et choisir un sujet pour leur publication. De plus, les utilisateurs ont la possibilité d'annuler cette action s'ils décident de ne pas publier leur contenu.

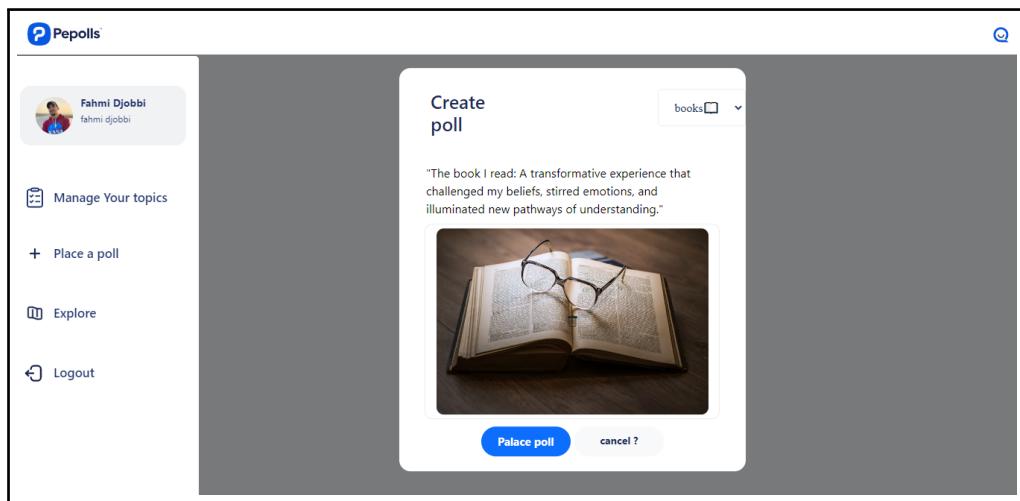


FIGURE 4.22 – Interface d'ajout de publication

Une fois la publication ajoutée, on peut observer sa forme sur la plateforme à la figure 4.23. Elle comprend le nom de l'utilisateur et des boutons d'interaction tels que "Like", "Dislike", "Trend", ainsi que des boutons pour commenter et partager la publication.



FIGURE 4.23 – Interface d'ajout de publication

4.2.2.4 Interface d'ajout de commentaire

La Figure 4.24 présente l'interface qui permet aux utilisateurs d'ajouter des commentaires à une publication et de répondre à des commentaires existants.

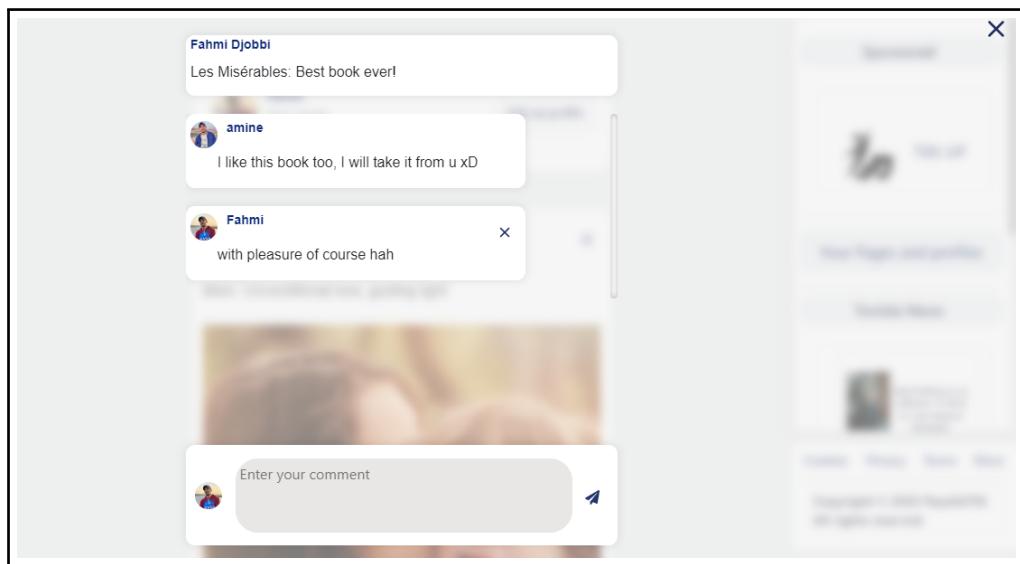


FIGURE 4.24 – Interface d'ajout de commentaire

4.2.2.5 Interface de suppression de publication

L'interface de suppression de publication, illustrée dans la Figure 4.25, permet aux utilisateurs de supprimer leurs propres publications sur Pepolls.

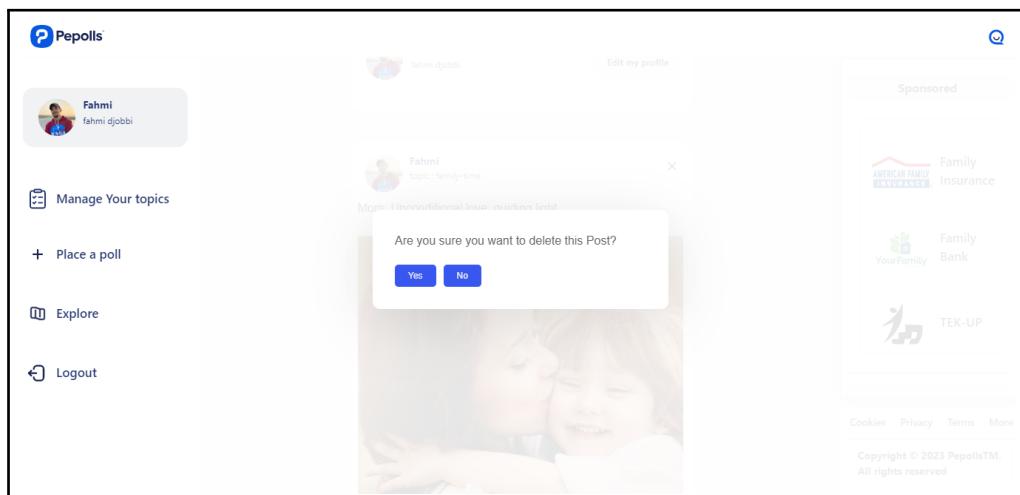


FIGURE 4.25 – Interface de suppression de publication

4.2.2.6 Interfaces du flux d'actualités

Les figures 4.26 et 4.27 présentent l'interface du flux d'actualités de notre plateforme. Ces interfaces permettent aux utilisateurs de visualiser les publications, les articles et les activités récentes partagés par les autres utilisateurs et les publicités. Le flux d'actualités offre une vue globale du contenu généré par la communauté.

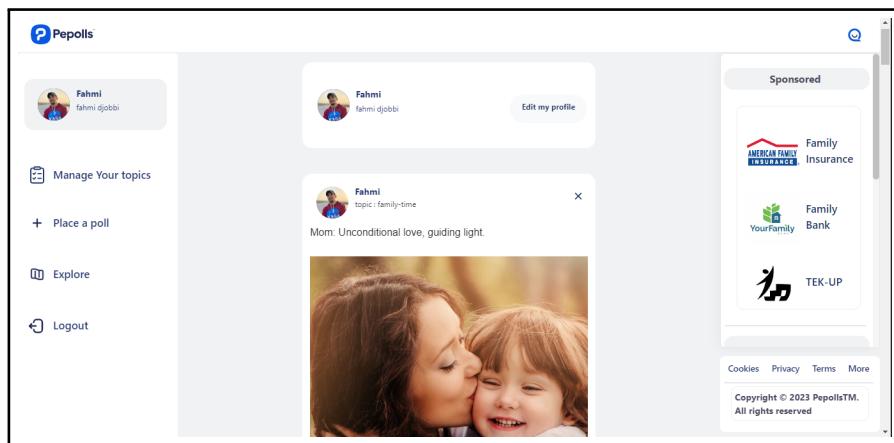


FIGURE 4.26 – Interfaces du flux d'actualités

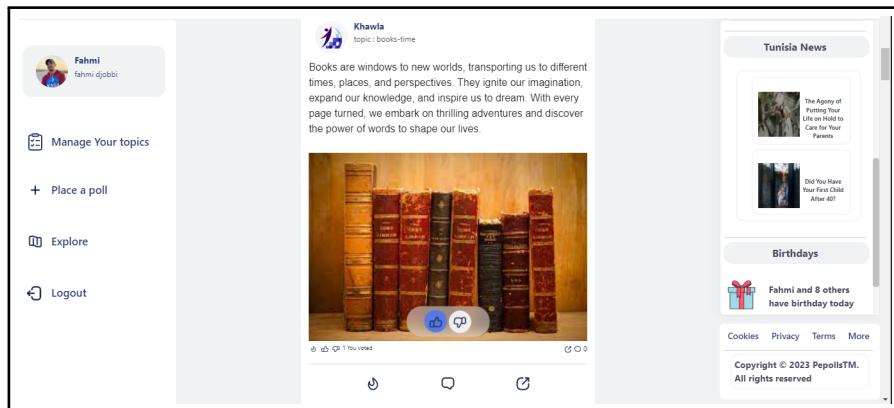


FIGURE 4.27 – Interfaces du flux d'actualités

4.2.2.7 Interface de partage de publication

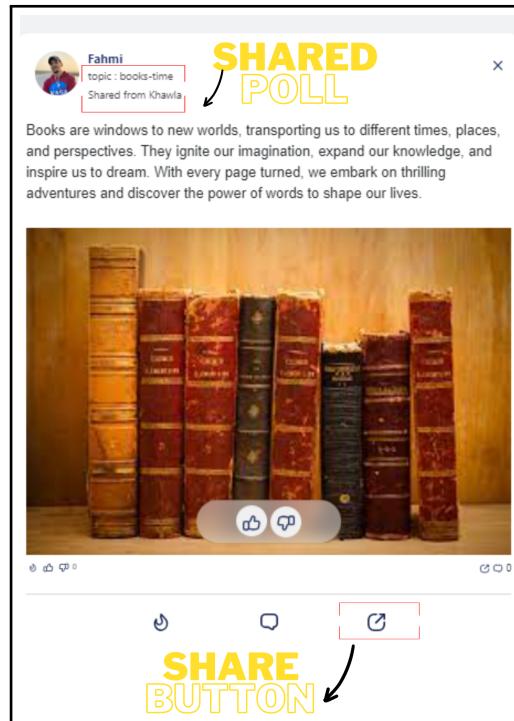


FIGURE 4.28 – Interface de partage de publication

La Figure 4.28 présente l'interface d'une publication partagée. Cette interface facilite la découverte et l'engagement avec le contenu partagé par d'autres utilisateurs sur notre plate-forme.

Conclusion

Cette plateforme offre de nombreuses fonctionnalités qui permettent aux utilisateurs de vivre une expérience similaire à celle d'un réseau social. Ils peuvent interagir les uns avec les autres tout en garantissant la sécurité de la collecte des informations des utilisateurs dans notre base de données. Dans ce chapitre, nous avons examiné toutes les étapes de création de cette release, de l'analyse et la conception des sprints à la réalisation. Dans le chapitre suivant, nous détaillerons la mise en place de notre système de recommandation NLP pour notre partie d'intelligence artificielle.

CHAPITRE 5

RELEASE 2

Introduction

Dans ce chapitre, nous nous concentrerons sur la description détaillée de la méthodologie appliquée et du travail accompli au cours de la deuxième release. Cette partie sera spécifiquement dédiée à l'analyse de la deuxième phase de réalisation, qui correspond à la mise en place du système de recommandation basé sur le traitement du langage naturel (NLP), le volet intelligence artificielle du projet.

5.1 Choix méthodologique

Dans cette section, nous exposons la méthodologie mise en place lors de notre deuxième release. Nous avons opté pour la méthodologie Cross-Industry Standard Process for Data Mining (CRISP-DM). Le choix de cette approche est motivé par sa pertinence et son efficacité pour les projets de Data Science (DS). En effet CRISP-DM, offre un cadre robuste pour la gestion des données et l'application des techniques de NLP. La Figure 5.1 illustre les différentes phases de cette méthodologie CRISP [18].

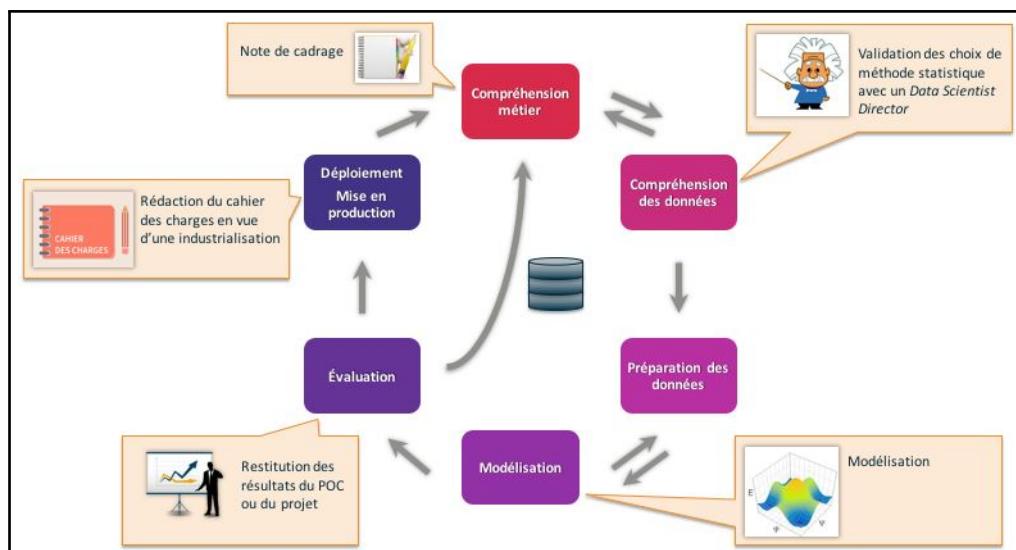


FIGURE 5.1 – Schéma de la méthodologie CRISP

5.1.1 Compréhension du Business

Cette étape de démarrage implique de comprendre en profondeur le problème métier que vous essayez de résoudre. Cela implique :

- Identification des objectifs et des critères de réussite du projet.

- Évaluation de la situation actuelle, y compris les ressources disponibles, les contraintes et les hypothèses.
- Développement d'un plan de projet pour atteindre les objectifs.

5.1.2 Compréhension des Données

Cette phase consiste à comprendre les données que vous allez utiliser. Cela comprend :

- Collecte de données initiales.
- Exploration des données pour identifier les tendances, les valeurs aberrantes et les relations entre les variables.
- Vérification de la qualité des données pour identifier les problèmes potentiels liés à l'exactitude, la complétude ou la pertinence des données.

5.1.3 Préparation des Données

Cette phase implique tous les travaux nécessaires pour construire le jeu de données final à partir des données brutes. Cela peut impliquer :

- Nettoyage des données pour corriger ou éliminer les données incorrectes ou incomplètes.
- Intégration des données en fusionnant les données de différentes sources.
- Transformation des données pour les adapter aux outils de modélisation.
- Réduction des données en sélectionnant les variables pertinentes ou en réduisant la taille de l'échantillon.

5.1.4 Modélisation

Dans cette phase, vous sélectionnez et appliquez diverses techniques de modélisation pour les données. Les étapes peuvent comprendre :

- Sélection de la technique de modélisation qui correspond le mieux à l'objectif .
- Construction du modèle en utilisant les données de formation et les hyperparamètres.
- Évaluation du modèle pour vérifier s'il répond aux objectifs du projet.

5.1.5 Évaluation

Lors de cette phase, vous évaluez le modèle de manière approfondie et vérifiez s'il répond aux objectifs métier. Cela peut comprendre :

- Évaluation des résultats pour déterminer si le modèle répond aux critères de réussite du projet.
- Révision du processus pour identifier les domaines à améliorer.
- Détermination de la prochaine étape, qu'il s'agisse du déploiement du modèle ou de l'itération du processus pour améliorer le modèle.

5.1.6 Déploiement

Lors de cette phase finale, vous mettez le modèle en production et suivez de près son rendement pour vous assurer qu'il produit les résultats souhaités. Cela peut comprendre :

- Planification du déploiement, y compris les tâches nécessaires pour intégrer le modèle dans les systèmes existants.
- Surveillance du modèle pour vérifier qu'il fonctionne comme prévu et pour détecter les changements qui pourraient affecter son rendement.
- Maintenance du modèle pour garantir qu'il continue à produire des résultats précis et pertinents au fil du temps.

5.2 Élaboration du sprint 3

Le troisième sprint est planifié sur une période de quatre semaines et se compose de trois éléments :

- **Business comprendre**
- **Compréhension des Données**
- **Préparation des Données**

5.2.1 Business comprendre

➤ **Définition des objectifs et des critères de réussite du projet :** Notre principal objectif est la mise en place d'un système de recommandation alimenté par l'intelligence artificielle en utilisant des modèles NLP. Ce système a le potentiel d'améliorer significativement l'expérience des utilisateurs et de stimuler leur engagement sur les plateformes de médias sociaux, tout en fournissant aux annonceurs des outils plus efficaces pour atteindre leur audience cible.

- **Évaluation de l'état actuel incluant les ressources disponibles, les contraintes et les suppositions :** Notre plateforme actuelle utilise un fil d'actualité statique. Les ressources dont nous disposons sont assez basiques et limitées. Nos contraintes majeures sont le manque de ressources et de données.
- **Élaboration d'un plan de projet pour atteindre les objectifs :** Notre plan d'action comprend la collecte de données par le biais du scraping, le développement du système de recommandation et, enfin, son intégration à notre plateforme existante. Pour garantir le succès du projet, nous nous engageons à atteindre une précision élevée dans les recommandations générées par le système. Cela se traduira par une amélioration de l'expérience de l'utilisateur et d'augmenter l'interaction sur notre plateforme.

5.2.2 Compréhension des Données

Dans cette phase, nous nous concentrons sur la compréhension des données.

5.2.2.1 Analyse des données réelles de l'application Pepolls

Pour commencer, nous avons effectué des analyses sur les données réelles de l'application de médias sociaux Pepolls pour mieux comprendre le type de données à récupérer et nos besoins spécifiques. Comme nous le constatons dans les diagrammes et graphiques de la figure 5.3, des sujets importants apparaissent dans le domaine de la technologie, notamment les startups, ainsi que dans les domaines des relations et de la famille. De plus, des thèmes populaires tels que les films, l'anime et les livres (relevant du style de vie) sont également présents avec une forte prévalence chez les hommes et un vote majoritairement en faveur du "oui". En outre, un volume considérable de données de type textuel a également été observé dans la figure 5.2, justifiant ainsi notre choix de nous concentrer sur le traitement du langage naturel (NLP) pour ce projet.



FIGURE 5.2 – Graphique de polls par type d'activités

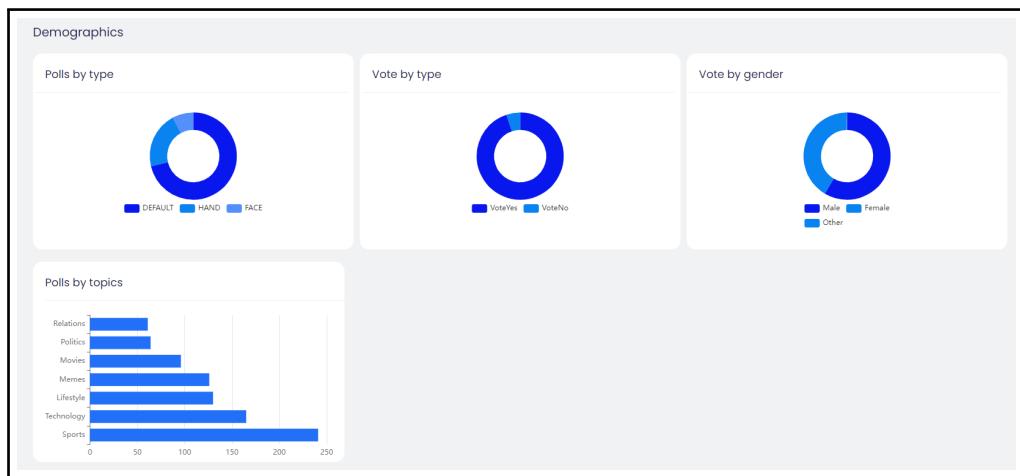


FIGURE 5.3 – Graphique de polls par topics

5.2.2.2 Extraction de données (Scraping)

Nous avons décidé de collecter nos données à partir de Reddit, comme le montre la Figure 5.4, une plateforme de partage de contenu connue pour son organisation en sous-forums, appelés "subreddits", qui traitent de sujets spécifiques. En exploitant son API open source, nous avons réussi à collecter 25 180 entrées de données de type Comma-Separated Values (CSV).

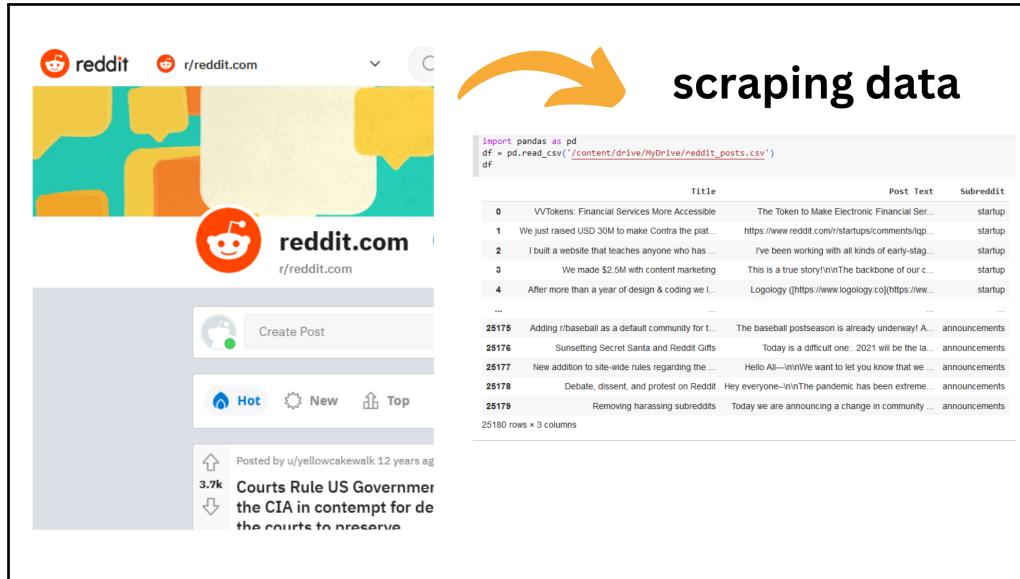


FIGURE 5.4 – Scraping data à partir de Reddit

Notre ensemble de données est composé de 4 colonnes et de 25 180 lignes, comme illustré dans la figure 5.5 ci-dessous.

	Title	Post Text	Subreddit
0	VVTokens: Financial Services More Accessible	The Token to Make Electronic Financial Ser...	startup
1	We just raised USD 30M to make Contra the plat...	https://www.reddit.com/r/startups/comments/lqp...	startup
2	I built a website that teaches anyone who has ...	I've been working with all kinds of early-stag...	startup
3	We made \$2,5M with content marketing	This is a true story!\n\nThe backbone of our c...	startup
4	After more than a year of design & coding we l...	Logology ([https://www.logology.co](https://ww...)	startup
...
25175	Adding r/baseball as a default community for t...	The baseball postseason is already underway! A...	announcements
25176	Sunsetting Secret Santa and Reddit Gifts	Today is a difficult one: 2021 will be the la...	announcements
25177	New addition to site-wide rules regarding the ...	Hello All—\n\nWe want to let you know that we ...	announcements
25178	Debate, dissent, and protest on Reddit	Hey everyone—\n\nThe pandemic has been extreme...	announcements
25179	Removing harassing subreddits	Today we are announcing a change in community ...	announcements
25180 rows × 3 columns			

FIGURE 5.5 – Les données extraites

Le tableau 5.1 suivant présente une vue détaillée de nos colonnes de données :

TABLEAU 5.1 – Description des colonnes de nos données

Colonne	Description
Index	un identifiant unique attribué à chaque ligne de notre ensemble de données.
Title	La colonne "title" contient le titre associé à chaque enregistrement de notre ensemble de données. Ce titre peut représenter le sujet, le contenu ou la nature de l'élément spécifique auquel il est associé.
Post text	La colonne "Post text" contient le texte complet associé à chaque enregistrement de notre ensemble de données. Elle offre une perspective plus approfondie sur le contenu et les informations partagées dans chaque poste.
Subreddit	La colonne "subreddit" indique le sous-forum (ou subreddit) auquel chaque enregistrement de notre ensemble de données est associé. Les subreddits sont des catégories thématiques spécifiques sur Reddit où les utilisateurs partagent du contenu lié à un sujet particulier.

5.2.2.3 Exploration des données

L'histogramme dans la figure 5.6 présente une visualisation des subreddits dans notre ensemble de données, avec le pourcentage et le nombre de chaque topic. Parmi tous les subreddits, nous avons choisi de nous concentrer sur cinq sujets principaux : "family", "anime",

"startup", "books" et "announcement". Il est intéressant de noter que les sujets "family" et "startup" sont les plus représentés dans notre ensemble de données, ce qui souligne leur importance et leur pertinence dans notre analyse.

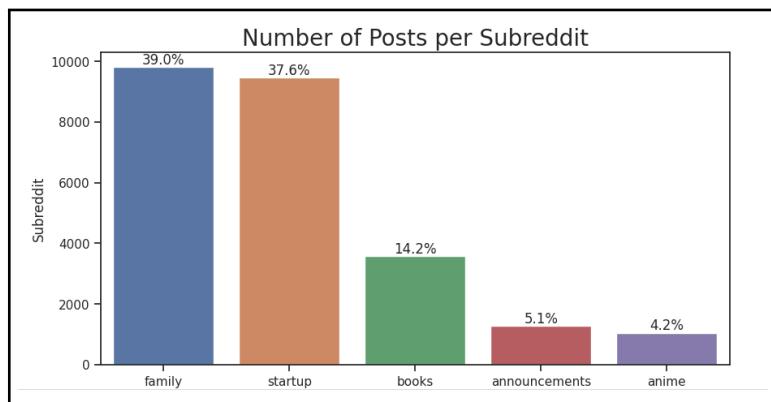


FIGURE 5.6 – Histogramme de subreddit

Le diagramme circulaire présenté dans la figure 5.7 suivante offre une vue plus claire du pourcentage de chaque sujet "subreddit" dans notre ensemble de données extrait. Cela nous permet de visualiser facilement la répartition des sujets et de comprendre leur importance relative dans notre analyse. Le sujet "family" représente le pourcentage le plus élevé dans nos données, avec une valeur de 39 %.

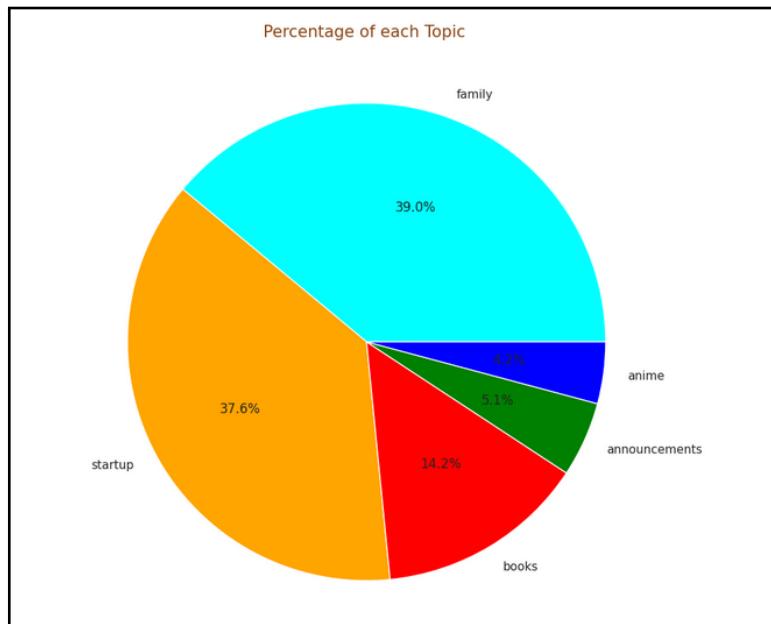


FIGURE 5.7 – Diagramme circulaire du pourcentage de subreddit

Nous avons utilisé une visualisation en treemap pour représenter la répartition proportionnelle de chaque sujet dans notre ensemble de données. Cette représentation graphique nous permet d'observer visuellement les différences de taille entre les sujets en fonction de leur fréquence respective. De plus, nous avons utilisé un heatmap pour afficher la corrélation

entre les titres et les sujets correspondants, ce qui nous permet d'identifier les tendances et les relations entre les sujets et les contenus publiés. Ces analyses sont présentées dans la Figure 5.8.

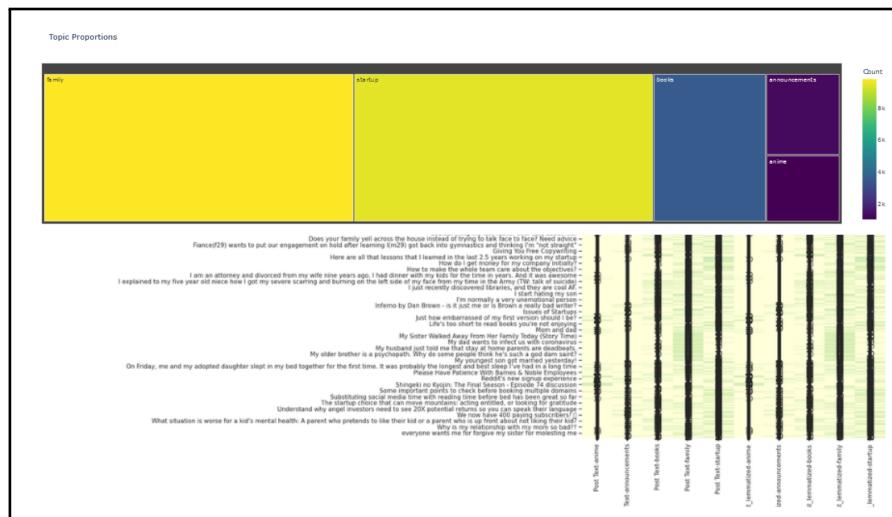


FIGURE 5.8 – Une visualisation en Treemap et Heatmap

Nous avons utilisé un scatterplot pour visualiser la fréquence des sujets de subreddit en fonction de leur index dans la figure 5.9 . Cette représentation graphique nous permet de voir la répartition des sujets et d'identifier les sujets les plus fréquents dans notre ensemble de données extraites

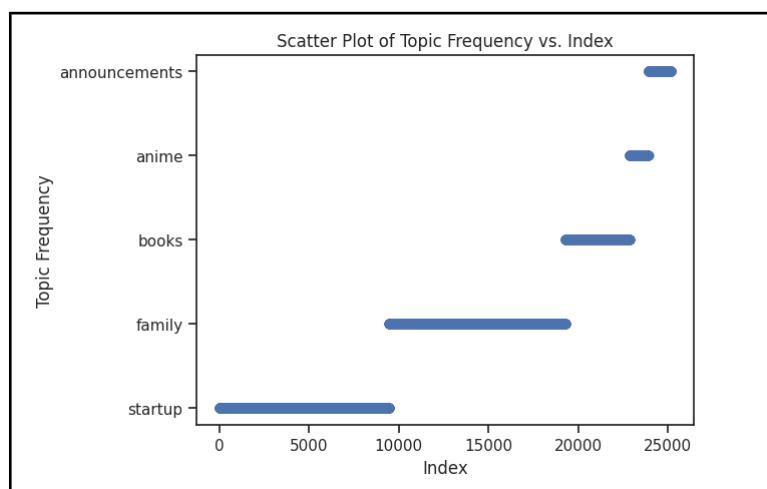


FIGURE 5.9 – Un scatterplot pour la fréquence de subreddit

À la fin de cette phase, nous visualisons les informations contenues dans notre ensemble de données. Nous examinons les types de données pour chaque colonne, ainsi que le nombre de valeurs manquantes. Nous constatons que toutes les colonnes sont de type "objet". En outre, nous fournissons une description détaillée de chaque colonne, en mettant en évidence la valeur la plus fréquente ("top") et sa fréquence correspondante. Ces analyses sont présentées dans la Figure 5.10.

```
[ ] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25180 entries, 0 to 25179
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Title       25180 non-null   object 
 1   Post Text   25180 non-null   object 
 2   Subreddit   25180 non-null   object 
dtypes: object(3)
memory usage: 590.3+ KB

[ ] df.describe().transpose()

   count    unique           top   freq
Title     25180    2508  Here's What Happened Last Week on Social Media...    50
Post Text  25180    2518    The Token to Make Electronic Financial Ser...    10
Subreddit  25180        5          family  9810
```

FIGURE 5.10 – Informations sur notre données

En conclusion, notre ensemble de données extraites est de bonne qualité, mais nécessite un nettoyage préalable pour la procédure NLP. Nous notons un léger déséquilibre dans la distribution des sujets, ce qui est compréhensible puisqu'il s'agit de données réelles.

5.2.3 Préparation des Données

Dans cette étape de préparation des données, nous allons procéder à diverses manipulations et transformations afin de rendre les données prêtes pour l'analyse et le traitement ultérieur.

5.2.3.1 Nettoyage de texte

Dans la phase de nettoyage de texte, nous avons effectué plusieurs opérations pour préparer le texte. Tout d'abord, nous avons converti le texte en minuscules pour assurer une cohérence. Ensuite, nous avons éliminé la ponctuation pour se concentrer uniquement sur les mots. Nous avons également vérifié que tous les caractères restants sont alphabétiques, en éliminant ainsi les caractères spéciaux et les chiffres. Enfin, nous avons supprimé les mots non pertinents en utilisant une liste prédéfinie de mots vides (stop words) en anglais. La Figure 5.11 illustre cette étape en montrant le flux de nettoyage de texte, où les différentes opérations sont appliquées pour obtenir un texte nettoyé et prêt pour l'analyse.

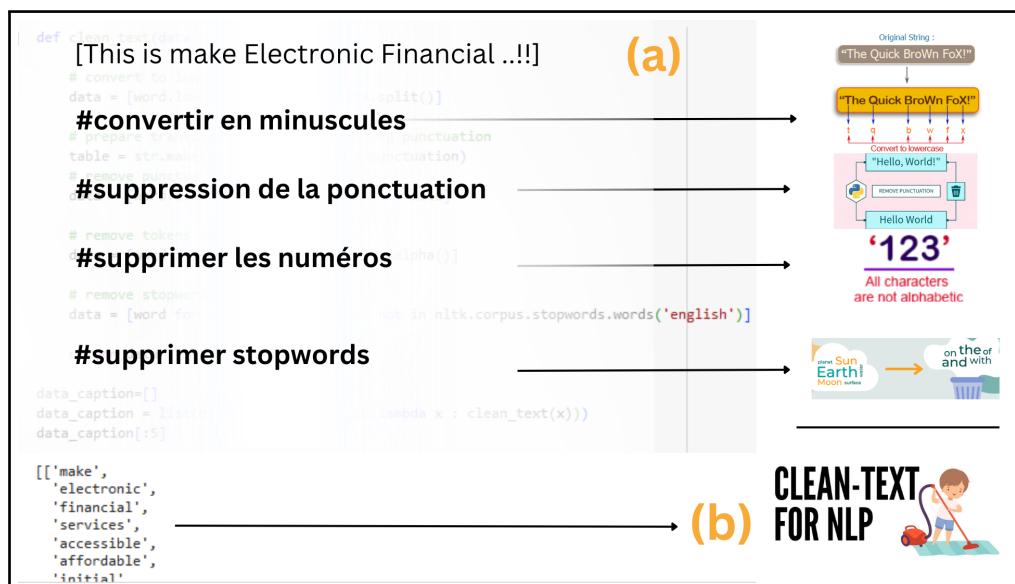


FIGURE 5.11 – Nettoyage de texte, (a) avant nettoyage, (b) après nettoyage

5.2.3.2 L'élimination des valeurs manquantes et indéfinies

Dans cette étape, nous avons également effectué l'élimination des valeurs manquantes et indéfinies dans notre jeu de données, comme indiqué dans la Figure 5.12. Cela garantit que nous travaillons uniquement avec des données complètes et valides, ce qui est essentiel pour une analyse précise.

missing_data		
Count and percentage of missing values for the columns:		
	Count	Percentage
Title	0	0.0
Post Text	0	0.0
Subreddit	0	0.0

FIGURE 5.12 – l'élimination des valeurs manquantes et indéfinies

5.2.3.3 Dévision des données

La division des données en ensembles d'entraînement et de test est essentielle pour évaluer les performances du modèle sur des données inconnues et éviter le surajustement. Une répartition courante est de 80 % pour l'entraînement et 20 % pour les tests. La Figure 5.13 illustre cette division des données.

```
[ ] print(train_df.shape)
print(test_df.shape)

(20144, 4)
(5036, 4)
```

FIGURE 5.13 – Taille de données de train et de test

5.2.3.4 Lemmatisation

L'étape de lemmatisation est une autre étape essentielle de la préparation des données. Elle consiste à réduire les mots à leur forme de base, appelée lemmes, afin de normaliser le texte. Cette étape permet de regrouper les différentes formes d'un même mot et de réduire la complexité du vocabulaire [19]. La Figure 5.14 présente cette étape de lemmatisation.

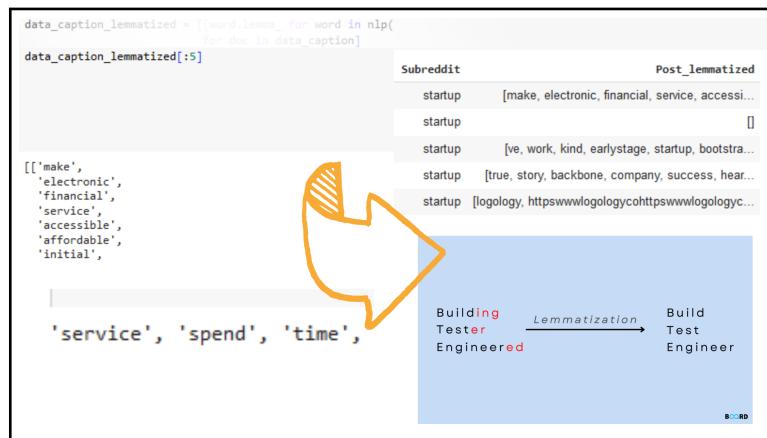


FIGURE 5.14 – Lemmatisation

5.2.3.5 ID2Word, Tokenization et Padding des séquences

Cette étape de préparation des données comprend des sous-étapes importantes : ID2Word, Tokenization et Pad Sequence. Ces techniques ont été définies au chapitre 2 du rapport. La figure 5.15 suivante présente l'utilisation de ces techniques dans notre projet.

```
df['Post_lemmatized'] = train_df['Post_lemmatized'].apply(lambda x: x.split())
create Dictionary
word = gensim.corpora.Dictionary(df['Post_lemmatized'])

#remove TOKEN FREQUENCY
for id, freq in word.doc2bow(df['Post_lemmatized']):
    if freq <= 10000:
        word._id2word[id] = None

#corpus
corpus = [word.doc2bow(text) for text in df['Post_lemmatized']]

#human readable format of corpus (id-frequency)
print([(id,word[id], freq) for id, freq in corpus[0].items()])

[ ] tokenizer = Tokenizer(num_words=10000)
tokenizer.fit_on_texts(df['Post_lemmatized'])
sequences = tokenizer.texts_to_sequences(df['Post_lemmatized'])
X = pad_sequences(sequences, maxlen=100)

[word_tokenize(text)]
['Hope', ',', 'is', 'the', 'only', 'thing', 'stronger', 'than', 'fear', '!', '#', 'Hope', '#', 'Amal.N']
```

FIGURE 5.15 – ID2Word, Tokenization et Padding des séquences

5.2.3.6 Visualisation avec WordCloud

nous avons utilisé un WordCloud pour visualiser les mots les plus fréquemment utilisés dans notre jeu de données. La Figure 5.16 présente le WordCloud résultant de notre analyse.

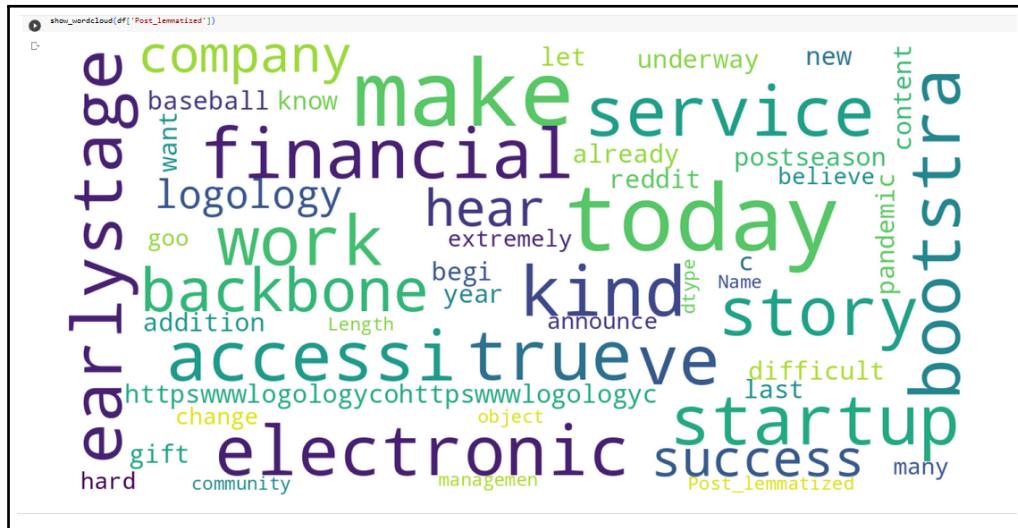


FIGURE 5.16 – les mots les plus fréquemment utilisés dans notre Data

La Figure 5.17 présente les WordClouds qui analysent les mots les plus fréquemment utilisés dans chaque sujet.

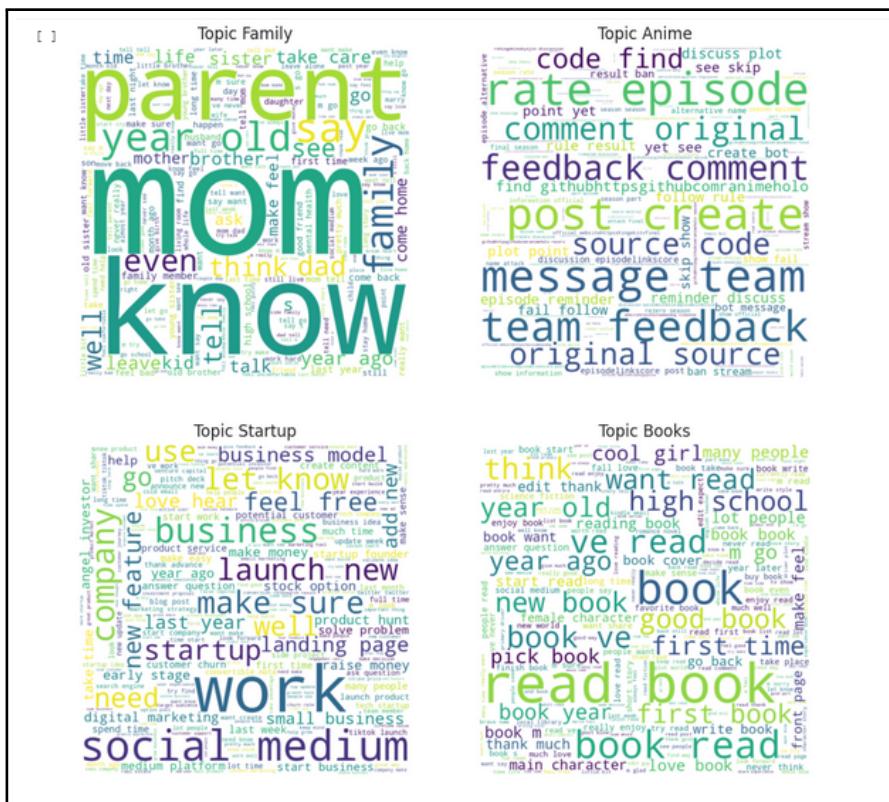


FIGURE 5.17 – les mots les plus fréquemment utilisés dans chaque sujet

5.3 Élaboration du sprint 4

Le quatrième sprint est planifié sur une période de quatre semaines et se compose de trois éléments :

- **Modélisation**
- **Évaluation**
- **Déploiement du système de recommandation**

5.3.1 Modélisation

Dans la phase de modélisation, nous avons utilisé six types de modèles pour construire notre modèle de traitement du langage naturel (NLP). Nous avons commencé par deux modèles d'apprentissage automatique, Latent Dirichlet Allocation (LDA), connus pour leur capacité de réduction de dimension et de modélisation des sujets. Dans la figure 5.18, nous pouvons observer la réduction de dimension de nos données sur les axes de composants 1 et 2. Après visualisation, nous remarquons que les tâches de couleur rouge sont les plus grandes, représentant le topic "family". De plus, nous constatons que certaines tâches de couleur verte sont incluses dans les tâches de couleur bleu clair, ce qui indique une certaine proximité entre les topics "books" et "anime" dans certains mots.

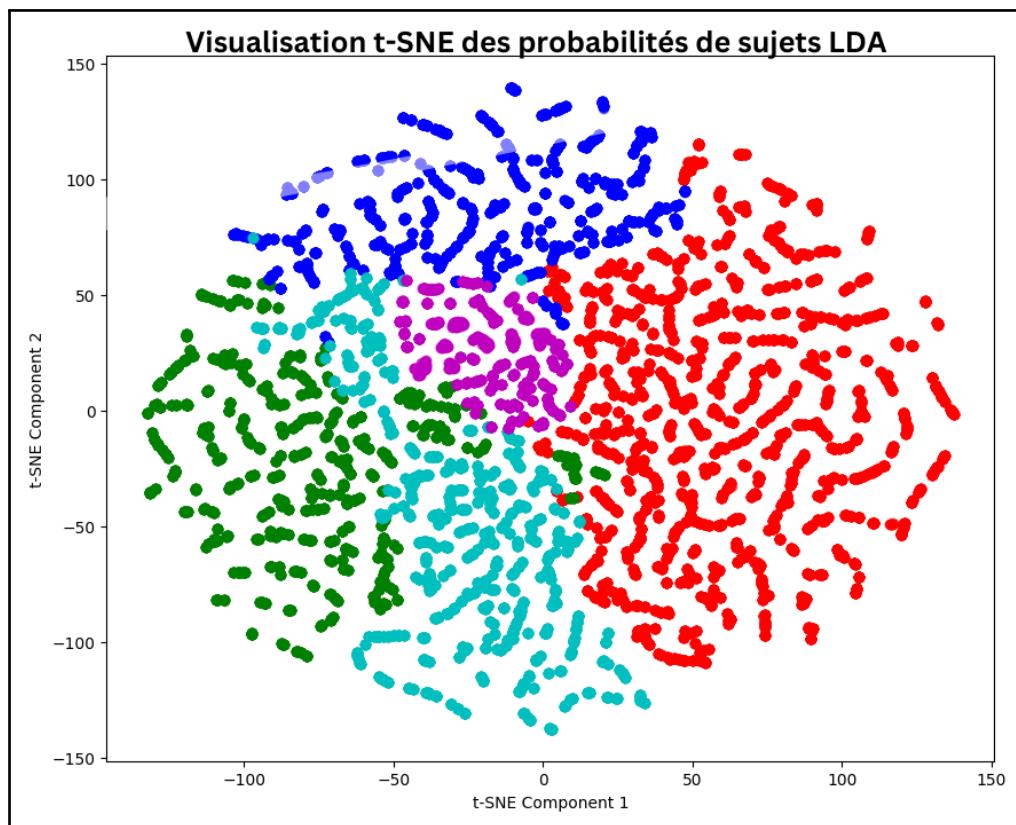


FIGURE 5.18 – Réduction de dimensions avec LDA

Ensuite, nous avons utilisé DecisionTree avec une profondeur maximale de 7, ce choix étant motivé par sa capacité à entraîner plusieurs arbres en parallèle.

Puis, nous avons expérimenté les modèles d'apprentissage profond, tels que CNN (Convolutional Neural Network) avec une couche d'embedding spécifique au NLP, utilisant l'optimiseur Adam, la rétropropagation par le biais de l'entropie croisée comme fonction de perte, avec 50 époques d'entraînement. Nous avons également utilisé les modèles LSTM (Long Short-Term Memory), BLSTM (Bidirectional LSTM) et GRU (Gated Recurrent Unit). Le chapitre 2 détaille l'utilité de chaque modèle.

[] model16.summary()		
LSTM		
Model: "sequential_1"	Model: "sequential"	
Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 500, 100)	1000000
lstm (LSTM)	(None, 128)	117248
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 5)	645
Total params: 1,117,893	Total params: 1,088,965	
Trainable params: 1,117,893	Trainable params: 1,088,965	
Non-trainable params: 0	Non-trainable params: 0	
GRU		
Model: "sequential_3"	Model: "sequential"	
Layer (type)	Output Shape	Param #
embedding_3 (Embedding)	(None, 500, 100)	1000000
bidirectional_1 (Bidirectional)	(None, 256)	234496
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 5)	1285
Total params: 1,235,781	Total params: 1,064,773	
Trainable params: 1,235,781	Trainable params: 1,064,773	
Non-trainable params: 0	Non-trainable params: 0	
CNN		
[] model1.save('BLSTM_NLP_Model_reddit.h5')		

FIGURE 5.19 – Un aperçu des paramètres utilisés dans chaque modèle

De plus, comme illustré dans la figure 5.19, nous avons ajouté une couche de dropout pour éviter l'overfitting du modèle avec un pourcentage de 20 % .

Aussi, on a choisi d'utiliser Adam et la cross-entropy en raison de leur performance élevée dans les tâches de classification. Adam offre un taux d'apprentissage adaptatif qui favorise une convergence rapide du modèle, tandis que la cross-entropy est une fonction de perte adaptée à la mesure de la différence entre les prédictions et les étiquettes de classe réelles. Leur efficacité et leur applicabilité étendue en font des choix populaires pour résoudre de nombreux problèmes de classification.

Nous avons choisi la fonction d'activation Softmax pour notre modèle car elle est couramment utilisée pour la classification multiclasse. Softmax attribue des probabilités à chaque classe, permettant ainsi de sélectionner la classe la plus probable pour chaque exemple. De plus, nous avons utilisé un batch size de 64 pour l'entraînement de notre modèle. Cette valeur

est un compromis entre l'efficacité du calcul et la capacité à généraliser sur l'ensemble des données.

Pour la réalisation de ces modèles, plusieurs bibliothèques ont été utilisées, notamment TensorFlow, Keras, spaCy, NumPy, Matplotlib, Pandas, Pickle, scikit-learn, os, NLTK, glob et Seaborn, ainsi que d'autres bibliothèques.

5.3.2 Évaluation

Dans cette partie d'évaluation, nous analyserons les résultats de nos modèles afin de déterminer si nous avons atteint nos critères de réussite en termes de performance commerciale. Nous évaluerons les différentes mesures de performance de nos modèles, telles que l'accuracy, la précision, le rappel, la F-mesure et matrice de confusion. Dans le chapitre 2, nous avons détaillé les mesures d'évaluation.

5.3.2.1 Score de cohérence et perplexité de modèle LDA

Nous avons utilisé le modèle LDA pour réduire la dimension de notre ensemble de données. Le score de cohérence de 0,4368 indique que les thèmes générés sont relativement cohérents et significatifs, ce qui permet une interprétation modérée des résultats. De plus, le score de perplexité négatif de -7,2704 suggère une performance prédictive raisonnable du modèle. Ces résultats témoignent de l'efficacité du modèle LDA dans la réduction de la dimension de notre ensemble de données tout en conservant une certaine cohérence et une capacité prédictive satisfaisante. Comme illustré dans la figure 5.20.

```
# PRINT THE SCORES
print("Coherence Score:", coherence_score)
print("Perplexity Score:", perplexity_score)

Coherence Score: 0.4368441171880728
Perplexity Score: -7.270399936712759
```

FIGURE 5.20 – Score de cohérence et perplexité de modèle LDA

5.3.2.2 Étude comparative des résultats de classification des sujets

Ce tableau 5.2 présente les résultats d'évaluation de l'exactitude (accuracy) de nos cinq modèles.

TABLEAU 5.2 – Comparaison de l'exactitude (accuracy) des 5 modèles

Modèle	Accuracy
GRU	0.99
LSTM	0.97
BLSTM	0.98
CNN	0.95
DecisionTree	0.65

Les modèles de deep learning, tels que GRU, LSTM, BLSTM et CNN, ont obtenu des performances globalement élevées dans l'analyse des résultats. GRU donne le meilleur score de 0.99 , ce qui indique son efficacité particulière dans le traitement du langage naturel complexe grâce à ses fonctionnalités sophistiquées.Suivi par BLSTM avec 0.98, LSTM avec 0.97.Même si CNN (Convolutional Neural Network) est initialement conçu pour le traitement d'images, il a obtenu un score de 0.95, démontrant ainsi sa capacité à être utilisé efficacement dans des tâches de NLP grâce à l'utilisation de couches d'embedding. Ces résultats mettent en évidence l'efficacité des modèles de deep learning dans le traitement du langage naturel complexe. En revanche, le modèle Decision Tree a donné des résultats moins satisfaisants avec un score de 0.65, probablement en raison de son inadaptation aux problèmes complexes de NLP.

Dans la poursuite de l'évaluation, nous avons comparé les modèles GRU et Decision Tree (Figure 5.21). Nous avons constaté que le modèle GRU affiche une précision, un rappel et un F1-score parfaits avec une valeur de 1, ce qui signifie que toutes ses prédictions positives sont correctes et qu'il identifie tous les exemples positifs. En revanche, le modèle Decision Tree présente des valeurs plus faibles de précision, de rappel et de F1-score, indiquant des performances moins satisfaisantes.

GRU					DECISION TREE				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	1.00	0.99	1.00	218	0	0.65	0.76	0.70	995
1	1.00	1.00	1.00	264	1	0.67	0.18	0.28	338
2	1.00	1.00	1.00	678	2	0.64	0.77	0.70	946
3	1.00	1.00	1.00	1906	3	0.47	0.13	0.20	139
4	1.00	1.00	1.00	1970	4	1.00	0.84	0.91	100
accuracy			1.00	5036	accuracy			0.66	2518
macro avg	1.00	1.00	1.00	5036	macro avg	0.69	0.54	0.56	2518
weighted avg	1.00	1.00	1.00	5036	weighted avg	0.65	0.66	0.63	2518

FIGURE 5.21 – Comparaison de la précision, du rappel et du f1score entre GRU et DescionTree

Dans la figure 5.22, nous montrons les résultats de classification en fonction de la matrice de confusion. Lors de la comparaison de la matrice de confusion entre les modèles GRU et Decision Tree, nous avons observé des différences significatives. La matrice de confusion du modèle GRU présente une répartition précise des prédictions, avec un nombre élevé de vrais positifs et de vrais négatifs. En revanche, la matrice de confusion du modèle Decision Tree révèle des erreurs plus fréquentes, avec un nombre plus élevé de faux positifs et de faux négatifs.

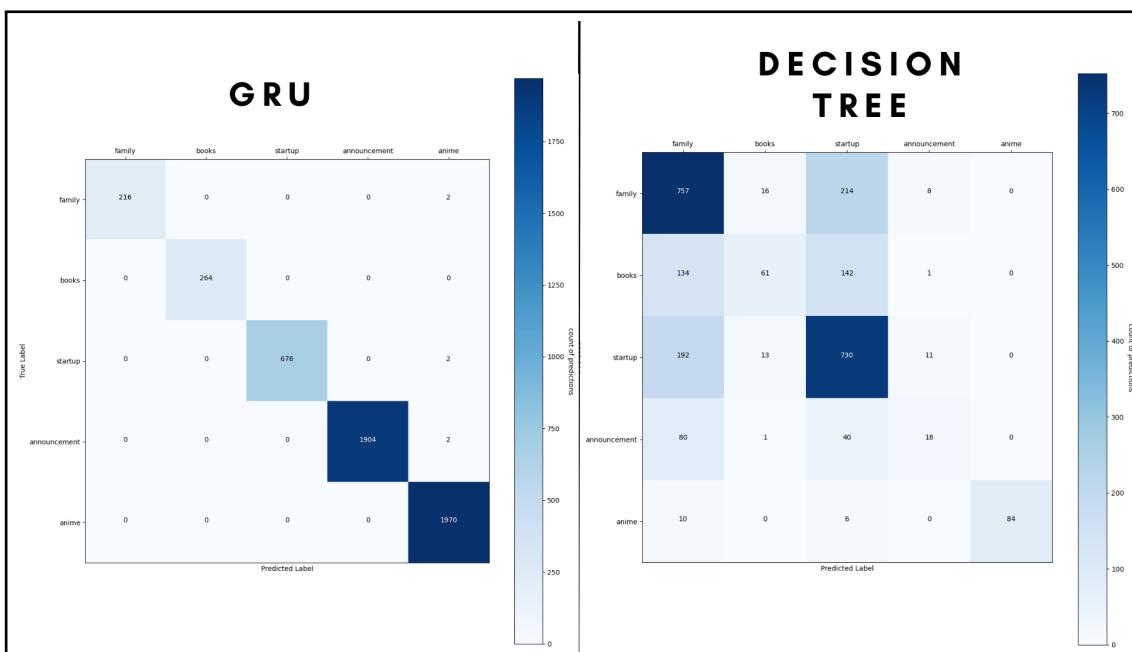


FIGURE 5.22 – Comparaison de la matrice de confusion entre GRU et DescionTree

5.3.2.3 Analyse des domaines à améliorer et identification des prochaines étapes

Suite à une évaluation détaillée de notre projet, nous avons réalisé un test de recommandation après avoir sauvegardé le modèle. Les résultats obtenus étaient précis et corrects, ce qui répond à nos besoins commerciaux. La figure 5.23 suivante illustre le test effectué :

```
= """ I spent the last 12 months in a remote wilderness lodge with no internet
cell service, and spent all my time reading. I did a ranking and mini-reviews of the 40ish books I read in that time """
['spend', 'last', 'month', 'remote', 'wilderness', 'lodge', 'internet', 'cell', 'service', 'spend', 'time', 'read', 'rank', 'minireview', 'book', 'read', 'tim
print("Predicted topic:", predicted_topic)

1/1 [=====] - 0s 20ms/step
[[7.0682617e-08 1.5034338e-08 9.9999511e-01 1.8991492e-11 4.7996091e-06]]
Predicted topic: books
```

FIGURE 5.23 – Test de modèle

Sur la base de ces résultats concluants, nous avons décidé de passer à l'étape de déploiement du modèle.

5.3.3 Déploiement du système de recommandation

Dans cette partie, nous aborderons le déploiement de notre système de recommandation au sein de la plateforme que nous avons développée.

5.3.3.1 Analyse et conception

Avant de procéder au déploiement de notre système de recommandation NLP basé sur le contenu, il est important de présenter le diagramme du cas d'utilisation, tel qu'illustré à la figure 5.24. Ce diagramme détaille le fonctionnement de notre système, qui commence par l'identification des publications préférées des utilisateurs. Ensuite, nous analysons le contenu de ces publications pour en extraire le sujet principal et suggérer un contenu similaire susceptible d'intéresser les utilisateurs.

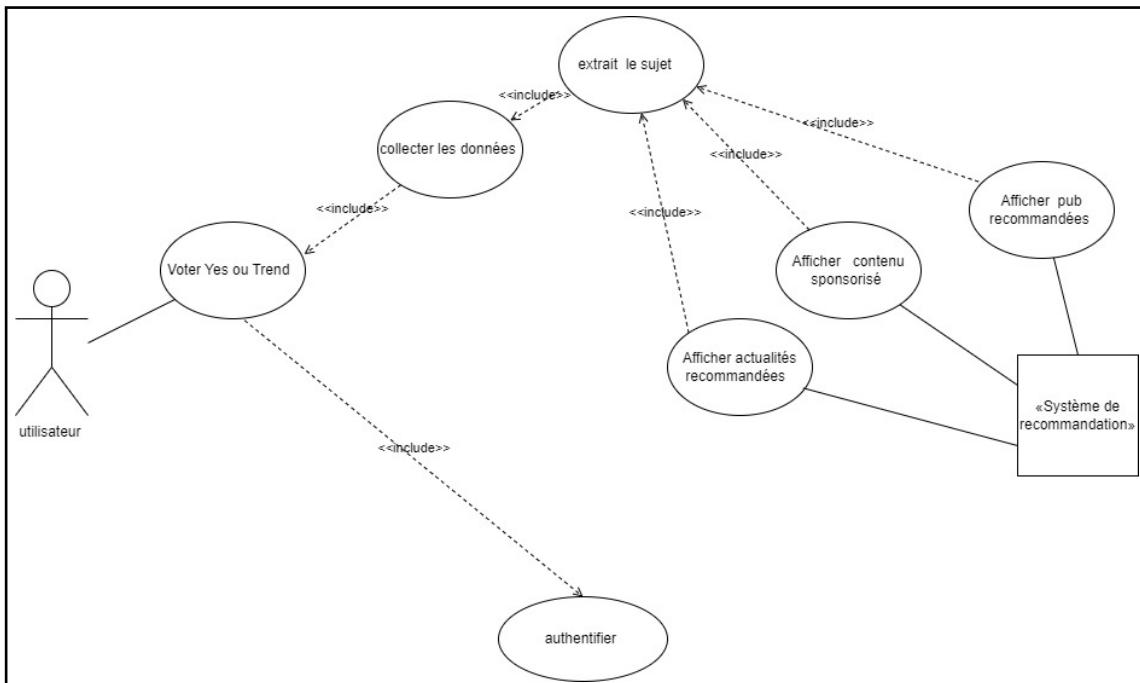


FIGURE 5.24 – Diagramme de cas d'utilisation de recommandation

5.3.3.2 Processus de recommandation en back-end



FIGURE 5.25 – Processus de recommandation en back-end

Le processus de recommandation en back-end comprend cinq étapes essentielles, comme le montre la figure 5.25 ci-dessus :

1. L'API *user_data* récupère les informations sur les interactions d'un utilisateur spécifique. En analysant toutes les publications de la collection *PUB* et en utilisant une boucle, les informations sont enregistrées dans un jeton (*token*).

2. L'API *recommandation* prend ce jeton, le déchiffre et le passe au modèle d'intelligence artificielle pour prédire les sujets d'intérêt de l'utilisateur. La Figure 5.26 présente le fonctionnement de l'API de recommandation.
3. Les sujets d'intérêt sont ensuite enregistrés dans la collection *user_interest* de la base de données.
4. L'API *get_recommendation* vérifie les intérêts de l'utilisateur dans la base de données, puis recherche des publications similaires à ses intérêts et qui n'ont pas été publiées par cet utilisateur.
5. Enfin, l'API *get_ads* recherche des publicités dans un fichier JavaScript Object Notation (JSON) qui sont similaires aux intérêts de l'utilisateur. De plus, une API spécifique au magazine *New York Times* peut recommander des articles similaires aux intérêts de l'utilisateur.

Ces étapes permettent de mettre en place un processus complet de recommandation en back-end, fournissant ainsi des suggestions personnalisées en fonction des intérêts de chaque utilisateur.

```
[ 'book', 'window', 'new', 'world', 'transport', '', 'different', 'time', 'place', 'perspective', 'ignite', 'imagination', 'expansive', 'knowledge', 'inspire', '', 'dream', '', 'page', 'turn', 'embark', 'thrill', 'adventure', 'discover', 'power', 'word', 'shape', 'live']  
1/1 [=====] - 0s 416ms/step  
[[2.5685958e-06 2.2957857e-07 9.9998021e-01 4.9019707e-07 1.6510130e-05]]  
Predicted topic: books  
127.0.0.1 - - [01/Jun/2023 22:42:30] "POST /api/v1/auth/recommendation HTTP/1.1" 200 -  
127.0.0.1 - - [01/Jun/2023 22:42:31] "POST /api/v1/auth/recommended-posts HTTP/1.1" 200 -
```

FIGURE 5.26 – API Recommandation

5.3.3.3 Processus de recommandation en front-end

Lorsque l'utilisateur est enregistré sur la plateforme de médias sociaux, il navigue à travers son fil d'actualité. Au moment où il interagit avec un "Poll" en y réagissant, notre système de recommandation est déclenché après un certain nombre de clics pré-déterminé. Le flux d'actualités est ensuite mis à jour avec les publications recommandées par notre système, et les API d'annonces et d'articles sont également lancées pour effectuer une mise à jour toutes les minutes (la fréquence de mise à jour des publicités et des articles est également contrôlée). La Figure 5.27 présente le déploiement de notre système de recommandation.

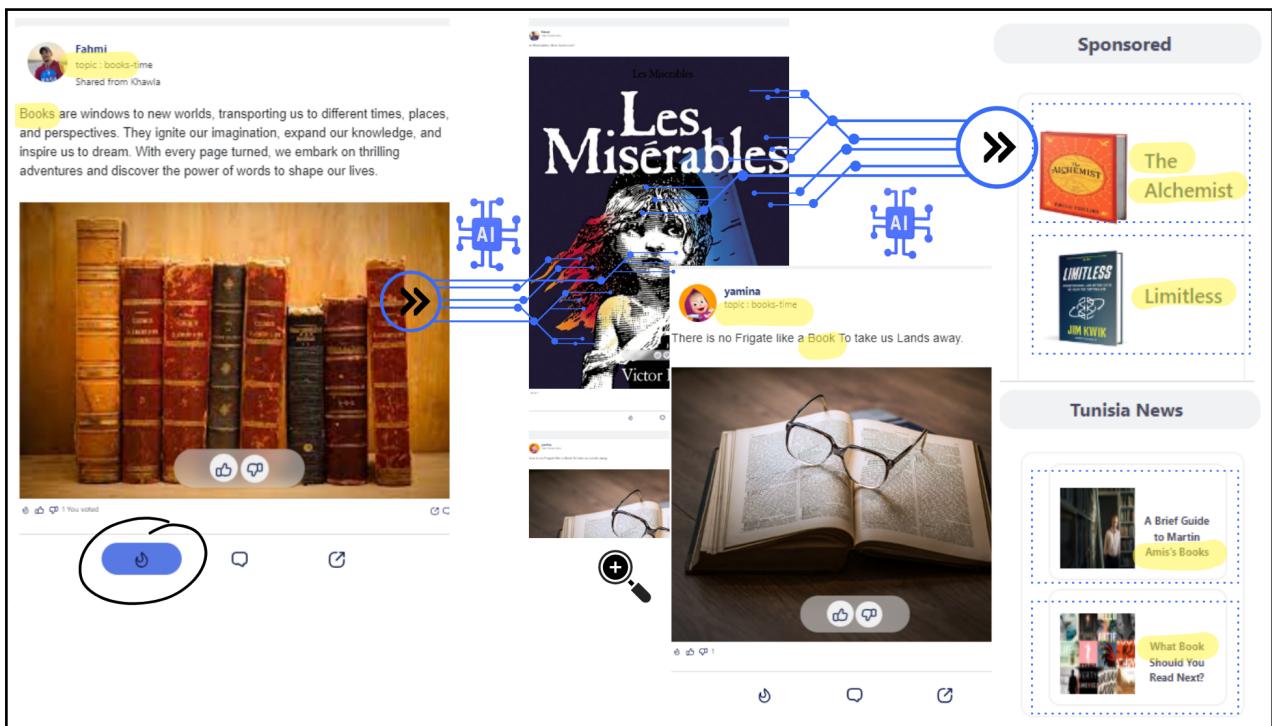


FIGURE 5.27 – Déploiement du système de recommandation

Conclusion

Dans ce dernier chapitre, nous avons discuté de la réalisation de notre système de recommandation en utilisant la méthodologie CRISP. Nous avons réussi à développer une version incrémentale qui répond aux exigences du client et qui peut être utilisée dans un environnement de production.

CONCLUSION GÉNÉRALE

Le projet de développement d'une plateforme de recommandation intelligente a été initié par Pepolls dans le but d'améliorer son réseau social, et a été réalisé au sein de l'équipe IA. Il offre une solution de flux d'actualités dynamique visant à proposer des publications et des publicités personnalisées à chaque utilisateur. Ce rapport détaille les différentes étapes de la mise en place de cette plateforme. Dans un premier temps, nous présentons le contexte général du projet, puis nous décrivons les systèmes de recommandation. Nous planifions ensuite le projet et le développons en plusieurs phases.

Nous avons assumé la responsabilité complète du développement de l'application web, de l'architecture à la conception, jusqu'à l'implémentation et le déploiement. Pour garantir une gestion efficace du projet, nous avons adopté les méthodologies agiles Scrum et CRISP-DM. Le projet a été divisé en deux releases, comprenant chacune deux sprints, en fonction des priorités définies par le client.

Bien que notre projet de développement d'une plateforme de recommandation intelligente ait réussi à transformer un fil d'actualité statique en un fil d'actualité dynamique, il reste encore des défis à relever pour exploiter pleinement le potentiel de l'IA. L'un de ces défis majeurs est le développement d'un modèle de détection des contenus inappropriés afin de maintenir un environnement sûr et conforme à nos valeurs. En outre, nous avons identifié plusieurs opportunités d'amélioration et de développement futur pour notre plateforme. Par exemple, nous envisageons d'intégrer un système de recommandation spécifique aux images et aux vidéos, ainsi que d'améliorer les fonctionnalités de notification et de messagerie de Talk. Nous sommes confiants que ces évolutions contribueront à la croissance continue et au succès de notre réseau social intelligent.

RÉFÉRENCES

- [1] Site web pepolls. <https://www.pepolls.com/>, 2023. [Accès le 1-mars-2023].
- [2] Scrum. <https://www.scrum.org/>. [Accès le 05-mars-2023].
- [3] Uml. <https://www.lucidchart.com/pages/fr/langage-uml>, 2023. [Accès le 20-mars-2023].
- [4] Tuzhilin Adomavicius, G. Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *ieee transactions on knowledge and data engineering*, 2005. 17(6), 734-749.
- [5] Martin J. H. Jurafsky, D. *Speech and language processing*. pearson education limited., 2020.
- [6] Idir Benouaret. Un système de recommandation contextuel et composite. <https://theses.hal.science/tel-01767997/>, 2017. [Accès le 04-mai-2023].
- [7] Charu C. Aggarwal. *Recommender systems*. http://pzs.dstu.dp.ua/DataMining/recom/bibl/1aggarwal_c_c_recommender_systems_the_textbook.pdf, 2016. [Accès le 08-mai-2023].
- [8] A. H. F. Laender R. B. Pinheiro and W. B. da Silva. *Natural language processing in recommender systems : A systematic review*, 2019.
- [9] Schütze H Manning, C. D. *Foundations of statistical natural language processing*. The MIT Press Series in Artificial Intelligence, 1999.
- [10] François Chollet. *Deep learning with python*. Manning Publications, 2018.
- [11] T. Mikolov et al. Efficient vectorization of words for large-scale word2vec, 2016.

- [12] Y Kim. Convolutional neural networks for sentence classification. proceedings of the 2014 conference on empirical methods in natural language processing, 2014. 1746–1751.
- [13] Schmidhuber J Hochreiter, S. Long short-term memory. neural computation. Neural Computation, 1997. 9(8), 1735-1780.
- [14] tensorflow. tf.keras.layers.gru. www.tensorflow.org/, 2023. [Accès le 1-avril-2023].
- [15] Shalabh Aggarwal. Flask framework cookbook. Published by Packt Publishing Ltd. Livery Place.35 Livery Street.Birmingham.B3 2PB, UK, 2019.
- [16] Alex Banks and Eve Porcello. Learning react modern patterns for developing react apps. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2020.
- [17] gitlab devops,. <https://about.gitlab.com/fr-fr/>. [Accès le 10-avril-2023].
- [18] SPSS. Crisp-dm consortium. crisp-dm 1.0 : Step-by-step data mining guide., 2000.
- [19] Nltk. <https://www.nltk.org/>, 2023. [Accès le 6-avril-2023].
- [20] Inc SCRUMstudy™, a brand of VMEd. A guide to the scrum body of knowledge (sbok™ guide)-third edition, 2017.
- [21] React.dev. <https://react.dev/learn>, 2023. [Accès le 12-avril-2023].
- [22] Flask. <https://flask.palletsprojects.com/en/2.3.x/>. [Accès le 15-avril-2023].
- [23] Nodejs. A propos de node.js. <https://nodejs.org/fr/about/>. [Accès le 20-avril-2023].
- [24] Keras. <https://keras.io/>. [Accès le 25-avril-2023].
- [25] Research paper classification systems based on tf-idf. <https://hcis-journal.springeropen.com/articles/10.1186/s13673-019-0192-7>. [Accès le 28-avril-2023].
- [26] Jack Stouffer Daniel Gaspar. Mastering flask web development second edition. Published by Packt Publishing Ltd. Livery Place.35 Livery Street.Birmingham, 2018.
- [27] Ben Auffarth. Artificial intelligence with python cookbook. Published by Packt Publishing Ltd.Livery Place.35 Livery Street.Birmingham. B3 2PB, UK, 2020.
- [28] Bootstrap Community. <https://getbootstrap.com/docs/4.1/getting-started/introduction/>, [Accès le 02-mai-2023].
- [29] Shalabh Aggarwal. Flask framework cookbook. Published by Packt Publishing Ltd.Livery Place.35 Livery Street.Birmingham. B3 2PB, UK, 2019.
- [30] Idir Benouaret. Applied machine learning at facebook. <https://abelay.github.io/6828seminar/papers/hazelwood:ml.pdf>, 2017. [Accès le 06-mai-2023].

