ASSISTANT VIRTUEL EN FINANCES PERSONNELLES

Par

Lancelot DOMART

RAPPORT DE PROJET DE LOG791 PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE COMME EXIGENCE PARTIELLE À L'OBTENTION D'UN BACCALAURÉAT EN GÉNIE LOGICIEL

MONTRÉAL, LE 7 AOÛT 2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

Professeur Alain April, directeur de projet
Département de génie logiciel et TI à l'École de technologie supérieure

TABLE DES MATIÈRES

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

CELI : Compte d'épargne libre d'impôt

ETF : Exchange-Traded Fund

NLP : Natural language processing

REER : Régime enregistré d'épargne-retraite

Subreddits : communauté spécifique au sein de la plateforme Reddit

INTRODUCTION

L'époque actuelle est une époque hyperconnectée où les données disponibles en ligne sont massives. Il est possible d'apprendre ou de s'instruire sur à peu près n'importe quel sujet. De nombreuses personnes partagent leurs connaissances sur les réseaux sociaux afin de les rendre profitables à tous. Toutefois, la quantité d'informations est tellement grande qu'il est facile de s'y perdre. Il serait donc pertinent de concevoir un outil permettant de résumer, regrouper et répondre aux questions d'une personne sur un sujet donné.

En l'occurrence, le sujet ciblé est la finance, et plus précisément les finances personnelles. Par "finances personnelles", on entend la gestion du budget, l'épargne, l'investissement, la planification fiscale, la retraite, ainsi que la compréhension des différents produits financiers (ETF, actions, obligations, comptes enregistrés comme le CELI et le REER). L'objectif général est de faciliter la prise de décisions éclairées en fonction de chaque situation et des objectifs financiers, en simplifiant l'accès à l'information et en offrant des réponses claires et pertinentes afin de favoriser l'autonomie individuelle.

C'est dans ce contexte que se pose la problématique suivante : comment permettre à chacun d'accéder à des informations simples, pertinentes et personnalisées pour mieux gérer ses finances, en tirant parti de l'abondance de données disponibles en ligne ?

Ce projet spécial vise donc à créer un outil intelligent capable de filtrer, comprendre et résumer l'information issue de Reddit, afin de fournir des réponses pertinentes à toute personne souhaitant mieux gérer ses finances.

Ce rapport présente la conception et la mise en place de cet outil, en s'appuyant sur des techniques de traitement du langage naturel et de recherche sémantique. Il est structuré comme suit : la première section présente le contexte et l'état de l'art. Ensuite sont abordées l'analyse des besoins, la conception de la solution et les choix techniques. Enfin, les étapes d'implémentation, les tests réalisés et les perspectives d'évolution sont présentés.

CHAPITRE 1

Revue littéraire

1.1 Introduction

Ce chapitre présente une courte synthèse des travaux scientifiques, techniques et professionnels en lien avec les thèmes centraux du projet. L'objectif est de situer la problématique dans l'état actuel des connaissances, d'identifier les approches existantes, ainsi que les outils et méthodologies utilisés dans des contextes similaires. Une attention particulière est portée aux assistants virtuels appliqués aux finances personnelles, aux contributions du traitement du langage naturel (NLP) dans la structuration et l'exploitation de contenus non structurés, ainsi qu'à l'usage de Reddit comme source d'informations. Cette revue vise non seulement à dégager les tendances dominantes du domaine, mais également à mettre en lumière les limites, les lacunes méthodologiques et les opportunités d'innovation que la solution proposée entend explorer. Les différentes sections qui suivent sont organisées de manière thématique, afin de favoriser une lecture structurée et cumulative du corpus analysé.

1.2 Méthodologie de recherche

Cette revue de littérature repose sur une recherche documentaire structurée visant à identifier les travaux pertinents liés aux assistants virtuels, aux finances personnelles, au traitement du langage naturel (NLP) et à l'exploitation des données issues de Reddit.

Les sources ont été sélectionnées à partir de bases de données scientifiques reconnues, telles que Google Scholar, IEEE Xplore, ScienceDirect et arXiv. Quelques articles professionnels et rapports techniques ont également été consultés afin d'élargir la perspective à des cas d'usage concrets.

Les mots-clés utilisés incluent notamment : « assistant virtuel finance », « NLP personal finance », « dialogueur financier », « analyse Reddit finance », « modèle de langage et données

financières ». Ces termes ont été combinés selon les besoins pour explorer les différentes dimensions du sujet.

Les documents retenus ont été choisis en fonction de leur pertinence pour la problématique, de leur date de publication (majoritairement après 2018), ainsi que de leur clarté méthodologique. Les travaux purement théoriques ou trop éloignés du domaine des finances personnelles ont été écartés.

L'analyse des textes a permis d'identifier les principales approches utilisées, les outils technologiques mobilisés, ainsi que les limites relevées dans les études existantes. Cette base permet de situer la contribution du présent projet dans l'état actuel des connaissances.

1.3 État de l'art

La conception d'un assistant virtuel intelligent dans le domaine des finances personnelles repose sur l'articulation de plusieurs champs disciplinaires. Cette section examine trois axes principaux : les technologies utilisées dans les dialogueurs financiers, les apports du traitement du langage naturel (NLP) dans la personnalisation des services, ainsi que le potentiel offert par les sources de données non structurées, en particulier Reddit.

1.3.1 Dialogueurs en finance : technologies et enjeux

Les dialogueurs financiers se sont largement démocratisés au cours de la dernière décennie, à mesure que les institutions bancaires ont cherché à automatiser une partie de leur service client. Ces agents conversationnels sont généralement intégrés dans des applications mobiles ou des sites web, et sont capables d'interagir avec les utilisateurs pour leur fournir des informations sur leur solde, leurs transactions récentes, ou encore des conseils budgétaires de base. Des exemples notables incluent Erica (Bank of America), Eno (Capital One), ou encore des agents intégrés dans les plateformes de banques en ligne, comme Revolut ou N26 (Brandtzæg et Følstad, 2018).

Sur le plan technique, la majorité de ces dialogueurs reposent sur des systèmes hybrides combinant traitement de règles, reconnaissance d'intentions (intent detection), et classification supervisée (Diederich, Brendel, et Kolbe, 2019). Toutefois, ces approches restent souvent limitées en termes de compréhension du contexte, d'adaptation au langage naturel, ou de capacité à gérer des requêtes complexes (Yin, Boyd-Graber, et Li, 2021). L'enjeu principal réside alors dans la capacité à dépasser une interaction purement transactionnelle pour offrir une véritable assistance décisionnelle, ce qui marque une transition vers ce que certains auteurs appellent la « conversation intelligence » (Zamora, 2017).

Un autre défi majeur concerne la gestion de la confiance utilisateur et la conformité réglementaire. Les dialogueurs opérant dans le domaine financier doivent respecter des exigences strictes en matière de protection des données personnelles (notamment le RGPD en Europe), de transparence algorithmique, et d'explicabilité des recommandations (Samek, Wiegand, et Müller, 2017 ; Ribeiro, Singh, et Guestrin, 2016). Cela complexifie considérablement leur déploiement à grande échelle, notamment dans les contextes soumis à des régulations financières strictes (European Banking Authority, 2021).

1.3.2 NLP et personnalisation des services financiers

Le traitement automatique du langage naturel (NLP) a connu des avancées majeures avec l'émergence de modèles préentraînés, tels que BERT, GPT, RoBERTa, ou encore T5 (Devoteam, 2025). Dans le contexte financier, des versions spécialisées ont été développées, comme FinBERT, afin d'adapter ces architectures à la terminologie et aux formulations propres aux textes économiques (Sharkey et Treleaven, 2024).

Ces modèles permettent non seulement d'extraire de l'information à partir de documents financiers (rapports annuels, actualités, publications d'utilisateurs), mais aussi de générer des réponses contextualisées ou d'analyser les intentions sous-jacentes dans une requête utilisateur. Certaines études ont mis en évidence leur capacité à améliorer la qualité des

recommandations financières, à condition de disposer de données représentatives et bien annotées (Sharkey et Treleaven, 2024).

La personnalisation constitue un enjeu central dans la conception de services intelligents. En intégrant les préférences de l'utilisateur, son profil de risque, ou encore son historique d'interaction, il devient possible d'adapter dynamiquement les réponses fournies. Cela suppose toutefois une capacité à modéliser des profils individuels de manière éthique, transparente et sécurisée, en tenant compte des biais potentiels introduits par les données d'entraînement (Devoteam, 2025).

Enfin, l'intégration de méthodes comme la recherche sémantique (semantic search) et la génération augmentée par la recherche (Retrieval-Augmented Generation – RAG) permettent d'associer la précision des systèmes de recherche à la flexibilité des modèles génératifs. Ce type d'architecture est particulièrement adapté à des contextes où l'information est hétérogène, fragmentée et non structurée (Devoteam, 2025).

1.3.3 Sources non structurées et Reddit

Les plateformes communautaires, comme Reddit, constituent des réservoirs d'informations particulièrement riches pour l'analyse des comportements, des opinions ou des questionnements liés aux finances personnelles. Des forums comme r/PersonalFinance, r/FinancialIndependence ou encore r/PersonalFinanceCanada rassemblent des milliers d'utilisateurs actifs qui échangent sur des sujets variés : investissement, épargne, fiscalité, planification de la retraite, produits financiers, etc.

L'exploitation de ce type de données présente un double intérêt. D'une part, elle offre un aperçu direct des préoccupations réelles des utilisateurs, exprimées dans un langage naturel, souvent spontané. D'autre part, elle permet de construire des cas d'usage concrets pour l'entraînement de modèles d'IA, en exposant ces derniers à une diversité de formulations, de styles et de contextes.

Toutefois, Reddit demeure une source de données difficile à traiter. Le contenu y est non structuré, parfois bruité, rédigé dans un registre informel, et sujet à des phénomènes linguistiques, tels que l'ironie, les sous-entendus ou les expressions idiomatiques. De plus, la pertinence et la fiabilité des informations partagées peuvent varier considérablement d'un utilisateur à l'autre.

Très peu de travaux se sont intéressés à l'analyse de Reddit dans une perspective d'assistance personnalisée. La plupart des études existantes se concentrent sur l'analyse des tendances de marché ou sur la détection d'événements boursiers, à l'image des recherches menées autour de l'affaire GameStop. L'utilisation de Reddit comme base de connaissances pour un dialogueur reste donc un champ largement ouvert, à fort potentiel, mais encore peu exploré sur le plan méthodologique.

1.4 Synthèse critique et identification des lacunes

L'analyse de l'état de l'art révèle des acquis significatifs, mais également des limites notables dans chacun des axes explorés.

Les dialogueurs financiers actuels (ex. Erica, Eno) demeurent principalement transactionnels et peu adaptés aux requêtes complexes ou personnalité. Ils reposent majoritairement sur des modèles hétérogènes (règles, classification) qui peinent à saisir les subtilités du langage utilisateur et à évoluer face à des contextes imprévus.

Les modèles de NLP spécialisés, tels que FinBERT, améliorent la reconnaissance du sentiment et la compréhension terminologique financière, mais montrent encore des lacunes en termes de sens implicite ou d'intentions sous-jacentes, surtout en l'absence d'indices explicites dans le texte. Par ailleurs, l'utilisation de Transformer complexes nécessite un fine-tuning rigoureux et pose d'importants défis de qualité des données et d'explicabilité.

Quant à Reddit, ses communautés offrent une mine d'informations authentiques sur les pratiques financières personnelles. Toutefois, le corpus est très hétérogène, bruyant, informel, et sujet à des enjeux d'ironie ou de biais contextuel.

1.5 Conclusion

Cette revue de littérature a mis en évidence trois enseignements principaux. Premièrement, les dialogueurs financiers existants déploient des capacités transactionnelles limitées et peinent à offrir un accompagnement personnalisé cohérent. Deuxièmement, bien que les modèles de NLP spécialisés, tels que FinBERT démontrent une puissance d'analyse et de classification supérieure, leur intégration dans des systèmes interactifs reste entravée par des exigences de fine-tuning, de qualité des données et de transparence explicative. Enfin, Reddit apparaît comme une source riche et authentique d'informations financières, mais sa nature bruitée et informelle complique fortement son exploitation.

Ces constats soulignent l'absence dans la littérature d'une démarche intégrée combinant dialogueur, NLP avancé et exploitation structurée de contenus Reddit. Le projet présenté dans ce rapport vise précisément à combler cette lacune, en proposant un système hybride capable de récupérer, analyser et restituer de manière contextualisée des informations issues de Reddit pour répondre aux besoins individuels en finances personnelles.

CHAPITRE 2

Proposition de solution

2.1 Introduction

Ce chapitre présente les décisions de conception et la solution proposée pour répondre à la problématique. Il expose comment les outils existants et les connaissances de l'état de l'art ont été adaptés et intégrés dans le système proposé.

2.2 Pipeline CI/CD

Afin d'assurer une livraison rapide des fonctionnalités dans le cadre du projet, l'approche DevOps a été privilégiée. Celle-ci permet d'automatiser les étapes de développement, de test et de déploiement grâce à l'intégration des pratiques CI/CD, facilitant ainsi des livraisons fréquentes, fiables et continues. Le serveur VPS, destiné à l'hébergement du projet, a d'abord été configuré pour permettre le déploiement du code. Une fois les dépendances nécessaires installées (NGINX, Docker, Certbot, etc.) et le déploiement initial effectué, un pipeline d'intégration et de déploiement continu a été mis en place à l'aide de GitHub Actions. Ce pipeline permet, à chaque mise à jour sur la branche principale, de valider automatiquement la qualité du code, puis de mettre à jour les conteneurs Docker via un processus de pull et de rebuild.

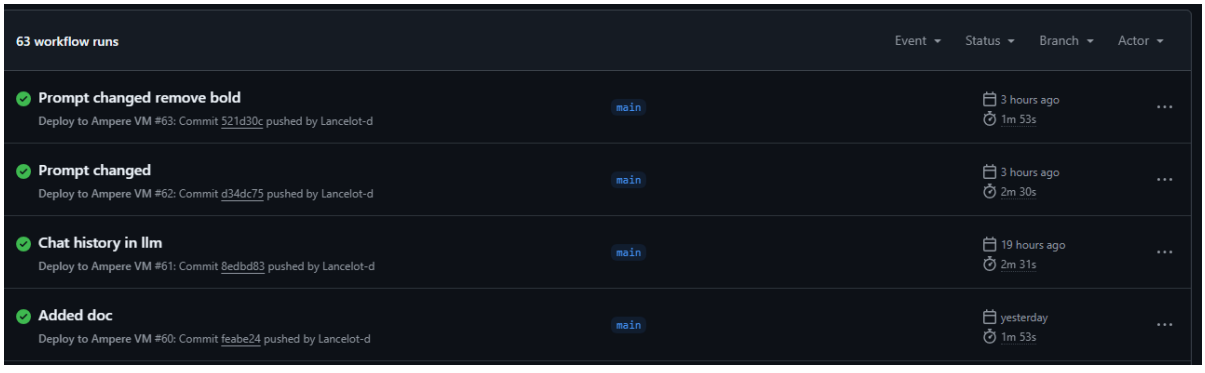

Figure 1 - Pipeline CI/CD GitHub

Cette automatisation a contribué à fluidifier le cycle de développement et à assurer un déploiement stable et rapide à chaque évolution du projet.

Le diagramme de déploiement, présenté à la figure 2, illustre l'infrastructure technique sur laquelle l'application est déployée, incluant les nœuds matériels, les conteneurs logiciels ainsi que les relations de communication entre eux.

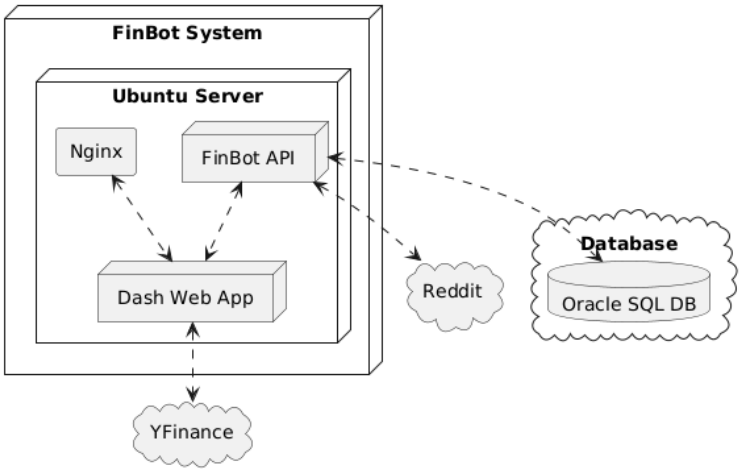

Figure 2 - Diagramme de déploiement

2.3 Pipeline de données

Le pipeline de données a été conçu dans le but d'interroger la plateforme Reddit afin d'en extraire le contenu des publications des utilisateurs ainsi que les réponses associées.

Pour réaliser cette tâche, une classe en logiciel libre disponible sur GitHub (https://github.com/datavorous/yars) a été utilisée. Cette dernière s'est avérée fonctionnelle avec seulement quelques ajustements mineurs. Néanmoins, un blocage de Reddit a rapidement été constaté en raison du volume important de requêtes émises. En moyenne, deux requêtes sont nécessaires par publication : une pour le contenu principal et une autre pour les commentaires, ce qui représente environ 200 requêtes sur une période inférieure à dix minutes.

Pour éviter ce blocage, une classe nommée « ProxyManager » a été développée. Celle-ci permet de gérer une rotation automatique des proxys afin de maintenir la continuité des requêtes. Ce mécanisme repose sur une liste de proxys associés à un compteur de succès, lequel s'incrémente après chaque requête effectuée avec succès. La liste est ensuite triée en ordre décroissant selon le taux de succès, et le système passe automatiquement au proxy suivant en cas d'échec.

PROXY	SUCCESS_REQUEST_COUNT
45.186.6.104:3128	823
98.8.195.160:443	117
204.157.185.4:999	103
181.224.244.50:999	102
186.167.80.234:8090	102
45.22.209.157:8888	101
190.121.145.115:999	96

Figure 3 - CSV pour la rotation de proxy

L'exécution du pipeline est assurée en tant que tâche d'arrière-plan (background task) intégrée à une API développée avec FastAPI. Cette tâche est planifiée pour s'exécuter toutes les huit heures. Les données extraites sont ensuite stockées dans une base de données SQL hébergée sur Oracle Cloud. Chaque publication est identifiée par un identifiant unique ainsi qu'une date d'insertion.

2.4 API pour le backend

2.4.1 Architecture générale

L'API backend est structurée autour de trois composants principaux :

- FastAPI Application : Point d'entrée principal qui orchestre les différents services et expose l'endpoint /complete_message/ pour les requêtes utilisateur ;

- Système de persistance : Un DAO (Data Access Object) implémentant le patron architectural « Singleton » pour gérer les interactions avec la base de données Oracle Cloud, assurant une connexion unique et optimisée ; et
- Pipeline de traitement asynchrone : Un système de tâches planifiées utilisant APScheduler pour automatiser la collecte et la mise à jour des données toutes les 8 heures.

Le diagramme de classe, présenté à la figure 4, synthétise l'ensemble des éléments abordés ci-dessus, en illustrant leurs relations et leur organisation structurelle.

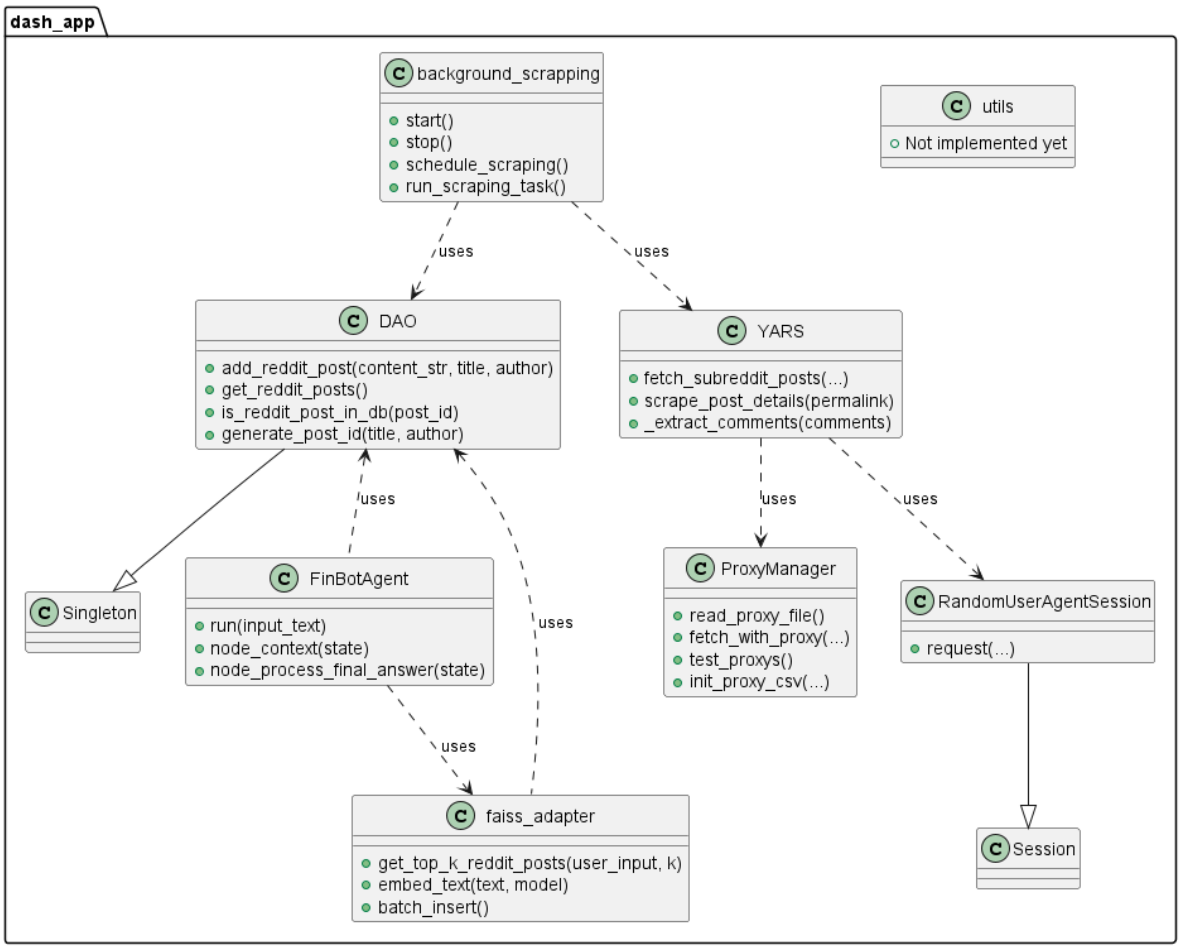

Figure 4 - Diagramme de classe API

2.4.2 RAG

L'indexation vectorielle est réalisée à l'aide de FAISS, en raison de sa simplicité d'implémentation et de sa capacité à générer un fichier d'index local (reddit_faiss.index). Cette méthode permet d'éviter la complexité associée à l'utilisation d'une base de données vectorielle externe et facilite le déploiement.

Le modèle d'« embedding » paraphrase-MiniLM-L3-v2 a été retenu pour son compromis efficace entre performance et légèreté (22 Mo). Sa taille réduite le rend adapté aux environnements soumis à des contraintes de ressources serveur limitées.

Le pipeline de recherche contextuelle s'articule autour de quatre étapes séquentielles : 1) vectorisation de la requête; 2) recherche de similarité dans l'index FAISS; 3) récupération des dix publications les plus pertinentes; et 4) extractions d'informations factuelles à l'aide d'un prompting structuré auprès du modèle de langage.

La mise à jour des données s'effectue par traitement par lots complets via la fonction batch_insert(), qui reconstruit intégralement l'index. Cette stratégie, privilégiant la simplicité de maintenance et la cohérence des données, est retenue au détriment d'une optimisation théorique permise par des mises à jour incrémentales.

2.4.3 LLM

L'intégration du modèle de langage s'appuie sur LangChain et utilise Meta-Llama-3-8B-Instruct via l'API de Together. Cette configuration permet de bénéficier des performances d'un modèle avancé sans nécessiter l'hébergement local d'une infrastructure gourmande en ressources.

La logique conversationnelle est encapsulée au sein de la classe FinBotAgent, qui repose sur un graphe d'état (StateGraph) structuré autour de deux nœuds principaux. Le nœud node_context est chargé d'extraire et de traiter le contexte pertinent à partir des publications

Reddit indexées, tandis que le nœud node_process_final_answer génère la réponse finale en combinant ce contexte avec les capacités d'analyse financière du modèle.

Des instructions structurées sont employées afin de définir explicitement le rôle du modèle en tant qu'expert financier et d'assurer un ton professionnel. Ces instructions intègrent conditionnellement le contexte issu de Reddit et orientent la génération vers des recommandations concrètes et directement exploitables.

La gestion asynchrone des traitements permet le traitement parallèle de multiples publications Reddit, ce qui optimise les performances globales du système. Cette architecture contribue à maintenir des temps de réponse satisfaisants, y compris en présence de volumes de données contextuelles importants.

2.5 Application Web

L'application web est développée à l'aide de la bibliothèque Python Dash, qui s'inspire de Streamlit tout en offrant une personnalisation plus poussée. L'utilisation de cette solution, plutôt que d'un front-end conventionnel tel que Angular ou React, vise à simplifier le développement en tirant parti de composants déjà intégrés. Par ailleurs, le choix d'une bibliothèque en Python permet de concentrer l'ensemble du développement dans un seul langage, ce qui améliore la maintenabilité du système et facilite l'intégration entre les différents modules, tous implémentés en Python.

L'application est composée de deux modules distincts : le premier permet d'obtenir des statistiques sur des symboles boursiers, tandis que le second propose une interface d'interaction avec l'agent conversationnel encapsulé dans l'API. Le visuel global est présenté à la figure 5.

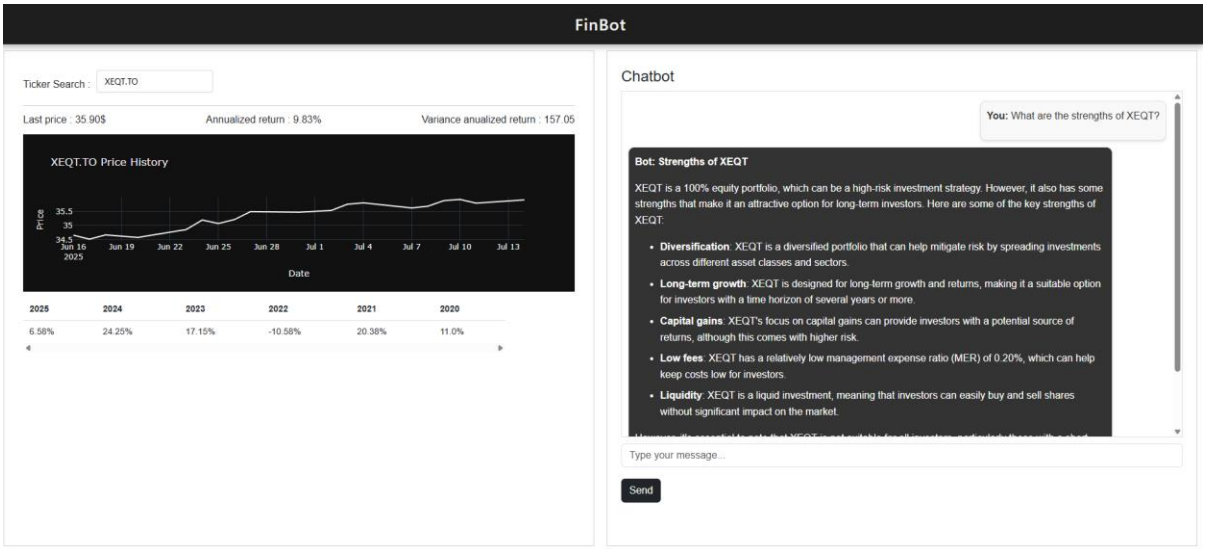

Figure 5 - Capture d'écran de l'application Web

L'architecture de l'application web peut être visualisée à la figure 6.

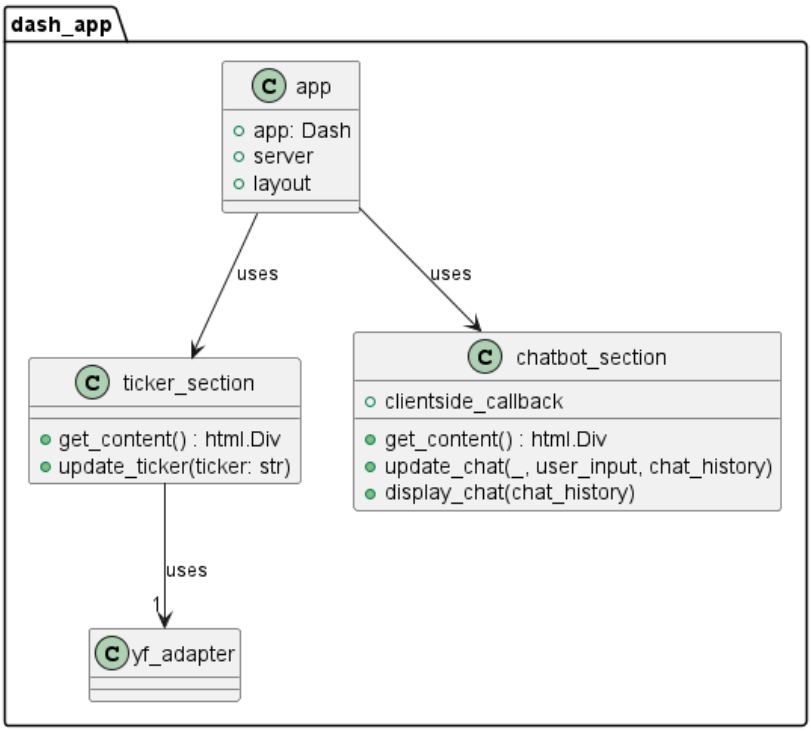

Figure 6 - Diagramme de classe de l'application Web

CHAPITRE 3

Présentation des résultats et analyse critique

3.1 Introduction

Ce chapitre présente les principaux résultats obtenus après la mise en place de la solution proposée. Il décrit les performances du système, la qualité des réponses générées, ainsi que le comportement observé dans différents cas d'usage. Une analyse critique permet ensuite de dégager les forces et les limites du prototype. Enfin, des pistes d'amélioration et des perspectives de développement futur sont discutées.

3.2 Résultats obtenus

3.2.1 Fonctionnement général du système

Lorsqu'une requête est soumise via l'interface web, elle est transmise à l'API back-end. Celle-ci déclenche un graphe « LangGraph » composé de deux nœuds. Le premier nœud vectorise la question à l'aide du modèle d'« embedding », recherche les publications Reddit les plus similaires, puis extrait les informations factuelles pertinentes. Le second nœud utilise ces données comme contexte pour générer la réponse finale.

Le système fonctionne de manière stable et fluide. Le temps de réponse moyen observé est d'environ 30 secondes, incluant l'ensemble des étapes (recherche, génération, affichage). La majorité des requêtes aboutissent à une réponse sans erreur, avec un taux de succès supérieur à 95 % dans les tests ad hoc réalisés.

3.2.2 Qualité des réponses générées

Les réponses générées sont, dans l'ensemble, cohérentes et en lien avec les publications Reddit identifiées comme pertinentes. Le système parvient généralement à reformuler l'information

de manière claire et structurée, tout en respectant le ton professionnel défini dans les instructions du modèle de langage.

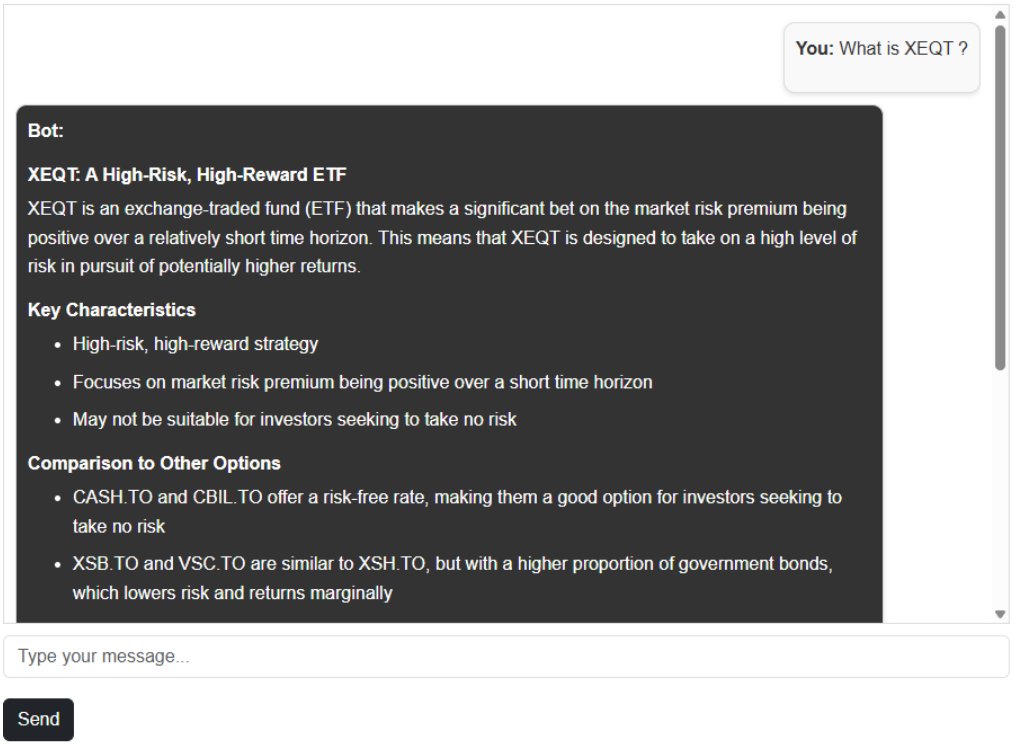

Figure 7 – Exemple de réponse

La figure 7 présente un exemple de réponse complète obtenue à partir d'une requête utilisateur. Dans ce cas, le système a correctement extrait les éléments essentiels des publications Reddit sélectionnées et les a synthétisés en une réponse concise et exploitable.

Certaines limites ont toutefois été observées dans des cas où les publications indexées manquaient de clarté ou lorsque les formulations étaient ambiguës. Cela peut entraîner des réponses vagues, partiellement incomplètes ou encore des hallucinations, comme illustré dans la figure 8.

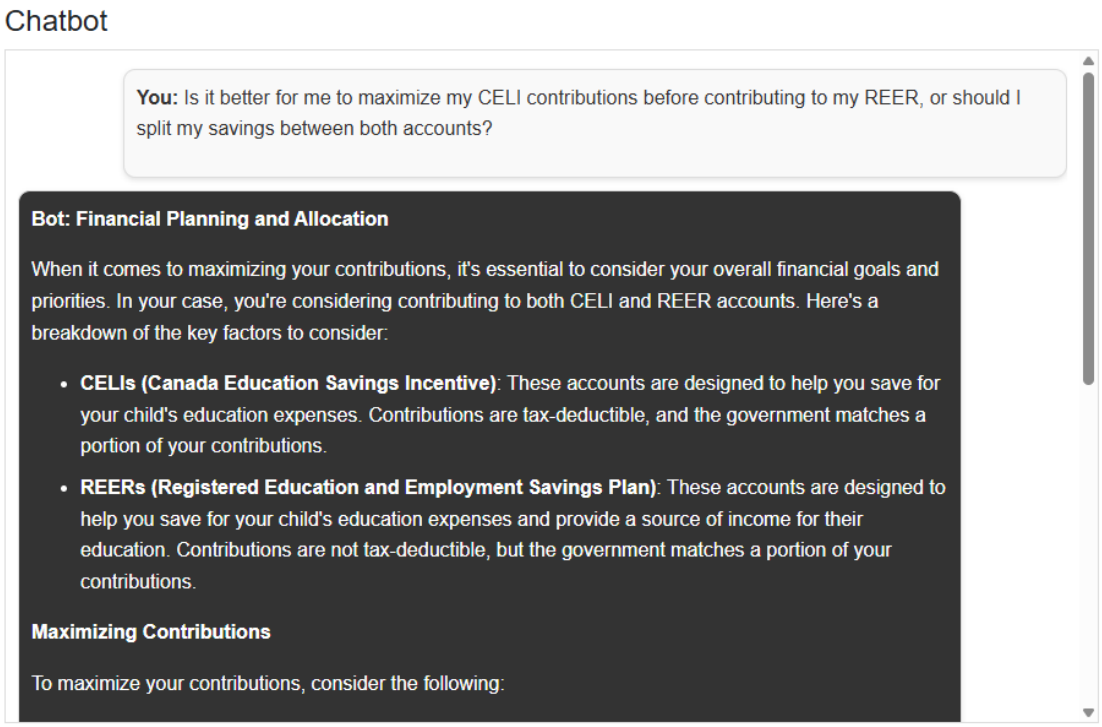

Figure 8 – Exemple d'hallucination

Dans l'ensemble, la qualité des réponses est satisfaisante pour un usage exploratoire ou informatif, mais une validation humaine pourrait être nécessaire avant d'en faire un usage décisionnel.

3.2.3 Performance du pipeline de données

La collecte des publications Reddit a débuté le 2025-02-19. Depuis cette date, un total de 26 444 publications a été insérées dans la base de données, provenant de plusieurs « subreddits » spécialisés en finances personnelles au Canada.

Le pipeline s'exécute automatiquement trois fois par jour et permet de récupérer en moyenne 182 nouvelles publications par jour, selon l'activité sur les « subreddits » ciblés. Les publications sont filtrées avant insertion afin d'exclure les contenus peu informatifs (liens, promotions, hors sujet ou publication trop courtes).

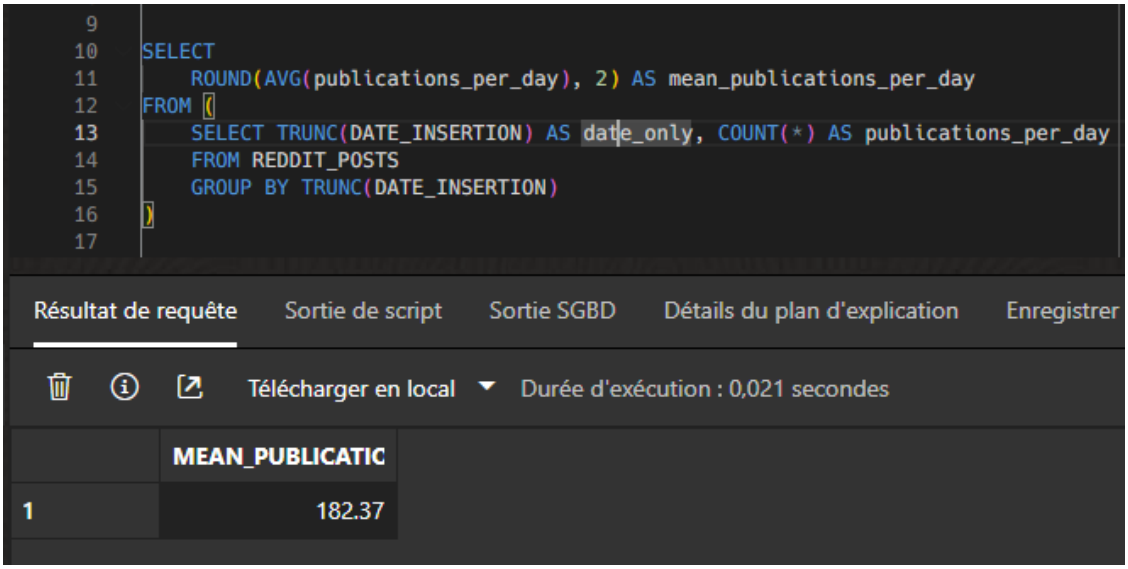

Figure 9 - Requête SQL pour calculer la moyenne des nouvelles publications par jour

Les données sont stockées dans une base relationnelle avec un identifiant unique, le contenu et la date d'insertion. Ce format facilite leur indexation et leur exploitation lors des requêtes utilisateur.

La stabilité du pipeline repose sur un système de rotation de proxys, qui permet de limiter les risques de blocage et d'assurer une extraction continue sans interruption majeure.

3.2.4 Robustesse de l'infrastructure

La séparation entre le front-end et le back-end renforce la résilience du système. En cas d'indisponibilité du back-end, le site web demeure accessible, permettant une continuité partielle du service.

L'hébergement sur Oracle Cloud de la base de données contribue également à la robustesse de l'ensemble, en limitant les points de défaillance uniques et en facilitant la maintenance indépendante des composants.

3.3 Synthèse des solutions investiguées

Les différentes composantes du système ont été testées et évaluées afin d'assurer leur adéquation avec les objectifs du projet. Le pipeline de collecte de données, basé sur une extraction régulière depuis Reddit, s'est révélé efficace pour alimenter la base avec un volume conséquent d'informations pertinentes.

L'indexation vectorielle avec FAISS a permis d'obtenir une recherche rapide et précise des publications en lien avec les requêtes utilisateur. L'utilisation du modèle d'embedding paraphrase-MiniLM-L3-v2 s'est avérée un bon compromis entre performance et légèreté.

La génération de réponses via le modèle Meta-Llama-3-8B-Instruct, orchestré par LangChain, a démontré une capacité satisfaisante à synthétiser et contextualiser les informations issues des données collectées.

Enfin, l'architecture modulaire et la séparation claire entre les couches front-end et back-end ont favorisé la robustesse et la maintenabilité du système.

Malgré ces points positifs, certaines limites persistent, notamment en termes de latence et de gestion des erreurs, qui feront l'objet d'améliorations futures.

3.4 Analyse critique

Le système développé, qui est un prototype expérimental, répond globalement aux objectifs fixés en matière d'assistance virtuelle pour les finances personnelles, en exploitant efficacement les données issues de Reddit et les capacités des modèles de langage avancés. La modularité de l'architecture et la séparation front-end/back-end ont contribué à assurer la stabilité et la maintenabilité.

Cependant, plusieurs points méritent une attention critique. Tout d'abord, la latence des réponses, bien qu'acceptable pour un usage exploratoire, limite l'efficacité dans un contexte

interactif en temps réel. Cette contrainte technique est liée aux traitements successifs de recherche vectorielle et de génération de texte.

Ensuite, la qualité des réponses dépend fortement de la richesse et de la pertinence des données collectées. Les informations disponibles sur Reddit étant parfois incomplètes ou imprécises, le système peut générer des réponses partielles ou approximatives.

Par ailleurs, l'absence actuelle de mécanismes robustes de gestion des erreurs et de rétroaction utilisateur peut nuire à la confiance et à l'expérience globale.

Enfin, l'extensibilité reste un enjeu important, notamment en cas d'augmentation significative du volume de données ou du nombre d'utilisateurs, ce qui nécessitera des optimisations et éventuellement l'intégration de solutions plus performantes.

Ces limites ouvrent des pistes d'amélioration pour renforcer la fiabilité, la réactivité et la pertinence de l'assistant virtuel.

3.5 Perspectives et travaux futurs

Plusieurs améliorations pourraient être envisagées afin de renforcer et d'étendre les capacités du système.

L'implémentation d'outils d'observabilité, tels que Grafana, permettrait de mieux superviser la santé des services, d'identifier rapidement les anomalies et d'optimiser les performances.

Le front-end pourrait être repensé pour améliorer l'ergonomie et l'expérience utilisateur, notamment par l'ajout de retours explicites en cas d'erreur ou lors du traitement des requêtes.

L’intégration d’agents spécifiques dédiés à la récupération de données financières précises, en complément des informations issues de Reddit, serait susceptible d’enrichir les réponses et d’apporter une valeur ajoutée plus fiable et spécialisée.

Par ailleurs, afin de mieux gérer l’augmentation des volumes de données et garantir des temps de réponse rapides, le recours à une base de données vectorielle externe plus performante pourrait être envisagé.

Ces évolutions contribueraient à rendre le système plus robuste, extensible et mieux adapté aux besoins évolutifs des utilisateurs.

CONCLUSION

Ce rapport de projet spécial a présenté la conception et le développement d'un prototype expérimental d'assistant virtuel dédié aux finances personnelles, reposant sur l'exploitation des données non structurées issues de Reddit et sur des techniques avancées de traitement du langage naturel.

Le système proposé combine un pipeline automatisé de collecte et d'indexation des publications, ainsi qu'une intégration de modèles de langage récents pour générer des réponses contextualisées.

Les résultats obtenus démontrent la faisabilité d'une telle solution et soulignent à la fois ses forces, notamment en termes de robustesse et de modularité, et ses limites, principalement liées aux performances et à la qualité variable des données disponibles.

Enfin, les perspectives évoquées offrent des pistes prometteuses pour améliorer l'extensibilité, l'expérience utilisateur et la précision des réponses, ouvrant ainsi la voie à un outil plus complet et mieux adapté aux besoins individuels en matière de gestion financière.

Ce travail constitue une étape importante dans le développement d'assistants virtuels financiers plus intelligents, personnalisés et accessibles à l'avenir.

ANNEXE I

BIBLIOGRAPHIE

Brandtzæg, P. B., & Følstad, A. (2018). Chatbots: Changing user needs and motivations. Interactions, 25(5), 38–43. https://www.researchgate.net/publication/327191388_Chatbots_changing_user_needs_and_motivations

Diederich, S., Brendel, A. B., & Kolbe, L. M. (2019). On conversational agents in the financial domain: Analyzing user expectations toward chatbots. In Proceedings of the Pacific Asia Conference on Information Systems (PACIS). https://www.researchgate.net/publication/329739413_On_Conversational_Agents_in_Information_Systems_Research_Analyzing_the_Past_to_Guide_Future_Work

European Banking Authority. (2021). Guidelines on ICT and security risk management. https://www.eba.europa.eu

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135–1144). https://www.researchgate.net/publication/305999024_Why_Should_I_Trust_You_Explaining_the_Predictions_of_Any_Classifier

Samek, W., Wiegand, T., & Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296. https://arxiv.org/abs/1708.08296

Yamato, Y., Boyd-Graber, J., & Li, Y. (2021). Towards realistic conversational agents: Managing coherence and context in dialogues. Transactions of the Association for Computational Linguistics, 9, 1–19. https://doi.org/10.1162/tacl_a_00367

Zamora, J. (2017). Rise of the chatbots: Finding a place for artificial intelligence in role-playing games. XRDS: Crossroads, The ACM Magazine for Students, 24(3), 30–35. https://dl.acm.org/doi/abs/10.1145/3703401

Devoteam. (2025, 4 février). LSTM, Transformers, GPT, BERT : guide des principales techniques en NLP. https://www.devoteam.com/fr/expert-view/lstm-transformers-gpt-bert-guide-des-principales-techniques-en-nlp/

Sharkey, E., & Treleaven, P. (2024, 24 avril). BERT vs GPT for financial engineering (arXiv:2405.12990). arXiv. https://arxiv.org/abs/2405.12990