



Université Ibn Tofail
Faculté des Sciences, Kénitra
Mémoire de Projet de Fin d'Études
Master Intelligence Artificielle et Réalité virtuelle

Assistant d'Entretien RH Basé sur l'IA

Établissement d'accueil : MintHR

Elaboré par : AIT AYAD Manal

Encadré par : **Mr. NOURI Anass** (Encadrant UIT)
 Mr. LATRACH Abdelhek (Encadrant externe)

Soutenu le 20-09-2024, devant le jury composé de :

- Mme TOUAHNI Raja (Faculté des sciences)
- MR MESSOUSSI Rochdi (Faculté des sciences)
- MR NOURI Anass (Faculté des sciences)
- MR BOUJIHA Tarik (ENSA)

Année Universitaire : 2023/2024

Table des matières

1.1	Motivation.....	8
1.1.1	Pourquoi ce sujet.....	8
1.2	Objectifs du projet.....	8
1.2.1	Statistiques et études récente.....	8
1.2.2	Bénéfices potentiels de l'automatisation.....	8
1.3	Contexte du projet.....	8
1.4	Présentation de la société.....	9
1.5	Problématique.....	9
1.6	Cibles du projet.....	10
1	État de l'Art	11
1.1	Recrutement automatisé.....	11
1.1.1	Technologies et outils.....	11
1.1.2	Avantages du recrutement automatisé.....	13
1.1.3	Défis et limitations.....	13
1.1.4	Études de cas et exemples.....	14
1.1.5	Tendances futures.....	14
1.1.6	Chatbots RH populaires.....	14
1.1.7	Tendances et innovations dans le recrutement automatisé.....	15
1.2	Chatbots et IA dans les RH.....	15
1.2.1	Recrutement et sélection.....	15
1.2.2	Onboarding et formation.....	16
1.2.3	Gestion des performances et développement.....	16
1.2.4	Engagement des employés et support.....	16
1.2.5	Défis et considérations éthiques.....	16
1.2.6	Tendances futures.....	16
1.2.7	Solutions d'intégration et de formation des employés.....	17
1.2.8	Outils d'évaluation des performances.....	18
2	Analyse des besoins	19
1.1	Analyse fonctionnelle.....	19
1.1.1	Identification des utilisateurs.....	19
1.1.2	Scénarios d'utilisation.....	19
1.2	Analyse technique.....	20
1.2.1	Contraintes techniques.....	20
1.2.2	Choix technologiques.....	20
1.3	Planification du projet.....	22
1.3.1	Diagramme de gantt.....	22
3	Développement	23
1.1	Configuration de l'environnement de développement.....	23
1.2	Développement du backend.....	23
1.2.1	Base de données.....	23
1.3	Développement du frontend.....	24
1.3.1	Frameworks et bibliothèques utilisés.....	24
1.4	Interface utilisateur.....	24

1.4.1	Candidats.....	24
1.4.2	Intégration du chatbot.....	26
1.4.3	FeedBack de la candidature.....	27
1.5	Èquipe RH.....	27
1.5.1	Gestion des offres d'emploi.....	27
4	Modèles et frameworks utilisés	31
1.1	Prétraitement des données.....	31
1.2	Exploration de modèles pour l'extraction de mots-clés.....	32
1.2.1	TF-IDF (Fréquence de terme-Fréquence inverse de document).....	33
1.2.2	TextRank.....	33
1.2.3	Bert.....	34
1.2.4	RAKE.....	34
1.2.5	KeyBERT.....	35
1.2.6	YAKE.....	35
1.3	Application d'extraction de mots-clés.....	35
1.3.1	KeyBert methodology.....	35
1.3.2	YAKE methodology.....	36
1.3.3	Textrank methodology.....	36
1.3.4	TF-IDF methodology.....	36
1.3.5	RAKE methodology.....	37
1.4	Méthodologie d'extraction et de classification des mots-clés.....	37
1.5	Modèles entraînés.....	38
1.5.1	Régression logistique.....	38
1.5.2	Forêt aléatoire.....	38
1.6	Méthodes d'optimisation.....	39
1.6.1	GridSearch.....	39
1.6.2	RandomSearch.....	39
1.7	Modèles de génération des questions.....	39
1.7.1	Modèle T5.....	39
1.7.2	Modèle BART.....	40
1.7.3	Modèle lmqg/t5-large-squad-qg.....	40
1.7.4	MStanford Question Answering Dataset.....	40
1.8	Les modèles d'évaluation des réponses.....	40
1.9	Framworks du chatbot.....	41
1.9.1	Rasa.....	41
1.9.2	DialogFlow.....	42
1.9.3	Microsoft Bot Framework.....	42
1.9.4	Botpress.....	43
1.9.5	IBM Watson.....	43
1.9.6	Wit.ai.....	44
1.9.7	ChatterBot.....	44
1.9.8	PandoraBot.....	45
1.9.9	Comparaison des frameworks du chatbot.....	45
5	Évaluation comparative des modèles : Étude comparative (Benchmarking)	46
1.1	Modèles d'extraction des mots-clés.....	46
1.2	Modèles de Génération de Questions.....	47
1.3	Model d'evaluation des réponses des utilisateurs.....	48
6	Challenges rencontrés	49
1.1	Challenges Techniques.....	49
7	Conclusion & Perspectives	51
1.1	Perspectives d'Amélioration Technologique du Chatbot.....	51

1.2 Critiques Sociologiques et Limites de l'Automatisation.....	52
1.3 Conclusion.....	53

Table des figures

1	Logo de l'entreprise.....	9
2	Recrutement Automatisé.....	11
3	ATS.....	12
4	Les chatbots dans les domaines de recrutement.....	12
5	l'AI dans le recrutement.....	13
6	Mya chatbot.....	16
7	Olivia chatbot.....	16
8	Xor chatbot.....	16
9	Talla—an AI-driven platform.....	17
10	Lms365.....	17
11	Reflektive.....	17
12	beterworks.....	18
13	Django freamwork.....	20
14	Python freamwork.....	20
15	PostgreSQL database.....	21
16	React.....	21
17	Shadcn.....	21
18	JavaScript.....	21
19	Diagramme de gant.....	21
20	MCD.....	22
21	Page d'accueil.....	23
22	Page d'inscription.....	23
23	Page de login.....	25
24	Interface des offres.....	25
25	Profil utilisateur.....	26
26	Interaction avec le chatbot.....	26
27	feedback candidature.....	27
28	Gestion des offres d'emploi.....	28
29	Candidature pour chaque job.....	28
30	Ajout d'une offre.....	28
31	Modification d'une offre.....	29
32	Suppression d'une offre.....	29
33	Invitation pour le deuxième entretien.....	30
34	Notifications RH.....	30
35	TF-IDF.....	33
36	TextRank.....	33
37	Bert.....	34
38	keybert.....	35
39	Modèle T5[20].....	40
40	rasa freamwork.....	41
41	DialogFlow.....	42
42	Microsoft Bot framework.....	42
43	Botpress.....	43
44	IBM Watson.....	43

46	chatterBot.....	44
47	PandoraBot.....	45
48	Comparaison des performances des modèles d'extraction des mots clés.....	46
49	Comparaison des performances des modèles de génération des questions.....	47
50	Performance du model de l'évaluation des réponses utilisateurs.....	48

Dédicace

Je me dédie ce succès et je voudrais me remercier pour ce que je suis devenu. Vous priez parce que vous le méritez.

À mes côtés fidèles, à celles dont les prières ont été la raison de ma réussite et le baume de mes blessures, à mon ange dans la vie, ma mère. Que Dieu prolonge ta vie et accorde à ton cœur tout ce qu'il désire.

À celui dont je porte le nom, mon père, merci pour tout ce que tu m'as donné.

À ma petite sœur bien-aimée, à mon petit soutien, au meilleur de mes jours, et à celle qui m'a tendu la main dans mes moments de faiblesse, je te souhaite prospérité et réussite dans la vie.

Remerciements

Mon premier remerciement est à DIEU le Tout Puissant. Je remercie le Bon DIEU, Tout Puissant, de m'avoir donné la force et l'audace pour dépasser toutes les difficultés et réaliser ce modeste travail.

À mon encadrant de stage, M. LATRACH Abdelhek. Grâce à votre encadrement, votre professionnalisme et votre générosité, j'ai pu acquérir une expérience précieuse et enrichissante. Votre soutien quotidien et votre expertise ont été des éléments déterminants de ma progression. Vous avez contribué à faire de ce stage une période d'apprentissage fructueuse et mémorable. Je tiens également à remercier toute l'équipe de développement de MintHR, et en particulier Mme. EL GOUSSAIRY Meriem , pour leur aide, leurs précieux conseils, et leur suivi tout au long de la réalisation de ce projet.

J'offre mes sincères et chaleureux remerciements à mon encadrant de mémoire, Pr NOURI Anass. Le mérite d'un mémoire appartient certes à l'auteur, mais également à son directeur qui l'encadre. Dans mon cas, mon encadrant a été d'un soutien et d'une attention exceptionnels. Merci pour tous les jours et les heures que vous avez sacrifiées afin d'encadrer ce travail. Merci pour votre disponibilité à tout moment, prêt à répondre à mes questions.

Enfin, je tiens à exprimer ma profonde reconnaissance envers toutes mes professeures pour leur dévouement et leurs efforts tout au long de ces deux années d'études. Leurs enseignements et leur accompagnement ont été essentiels pour mon développement académique et professionnel.

Je souhaite également exprimer ma gratitude envers la faculté des sciences pour m'avoir offert cette précieuse opportunité d'explorer le monde professionnel. Cette expérience m'a permis d'acquérir des compétences pratiques et de mettre en application les connaissances théoriques acquises en classe.

Introduction

À l'ère numérique actuelle, les technologies de l'information et de la communication transforment radicalement nos pratiques professionnelles dans de nombreux domaines. Parmi ces transformations, l'intelligence artificielle (IA) et le traitement du langage naturel (NLP) se distinguent comme des outils puissants capables de révolutionner des processus complexes et souvent chronophages. En particulier, dans le domaine du recrutement, qui est crucial pour le succès de toute organisation, l'optimisation et l'automatisation des processus sont devenues essentielles. Les entreprises cherchent constamment des solutions innovantes pour améliorer l'efficacité des entretiens d'embauche, réduire les biais humains et offrir une expérience plus fluide et agréable aux candidats.

L'intégration de chatbots intelligents capables de mener des entretiens automatisés représente une avancée significative vers cette optimisation. Grâce aux capacités de l'IA, il devient possible de rationaliser le processus de recrutement tout en assurant une évaluation plus objective des candidats. Ce projet se propose d'explorer ces opportunités en développant un système novateur d'entretien basé sur un assistant virtuel, avec pour objectif d'améliorer l'efficacité et l'équité du recrutement.

1.1 Motivation

1.1.1 Pourquoi ce sujet

Le recrutement est crucial pour la croissance et la réussite des entreprises, car il permet d'attirer et de sélectionner les talents qui contribueront à leur développement. Cependant, ce processus est souvent semé d'embûches. La gestion d'un grand nombre de candidatures peut être particulièrement chronophage et inefficace, rendant difficile la sélection des meilleurs profils. Par ailleurs, l'évaluation des compétences est souvent biaisée par des critères subjectifs, ce qui peut mener à des décisions de recrutement imparfaites. Réduire ces biais est essentiel pour favoriser la diversité et l'inclusion au sein des équipes, une priorité pour de nombreuses entreprises. L'automatisation du recrutement apparaît ainsi comme une solution prometteuse. En automatisant certaines tâches, le processus de sélection devient plus fluide et objectif, permettant aux recruteurs de se concentrer sur les aspects les plus stratégiques et humains de leur métier.

1.2 Objectifs du projet

1. **Automatisation des tâches répétitives** : Utiliser des chatbots et des systèmes d'IA pour automatiser le tri des candidatures, la présélection et la programmation des entretiens.
2. **Évaluation objective des compétences** : Développer des algorithmes pour évaluer les compétences des candidats de manière objective, basée sur des données et des critères prédéfinis.
3. **Amélioration de l'expérience candidat** : Créer une interface utilisateur interactive et réactive pour améliorer l'expérience de candidature et fournir des réponses rapides et pertinentes aux candidats.
4. **Réduction des biais inconscients** : Implémenter des mécanismes pour minimiser les biais inconscients dans le processus de recrutement, en s'appuyant sur des évaluations basées sur les compétences et les performances plutôt que sur des critères subjectifs.

1.2.1 Statistiques et études récente

Selon une étude réalisée par LinkedIn [47] en 2023, 67% des recruteurs déclarent que trouver des candidats qualifiés est le plus grand défi du recrutement. De plus, le processus manuel de tri et d'évaluation des candidatures peut prendre en moyenne 23 heures par semaine pour les recruteurs, selon une étude de Glassdoor. Une autre enquête menée par PwC en 2022 a révélé que 77% des entreprises estiment que l'amélioration de l'efficacité du recrutement est une priorité stratégique.

1.2.2 Bénéfices potentiels de l'automatisation

L'automatisation du recrutement présente de nombreux avantages. En premier lieu, elle permet de gagner du temps en automatisant les tâches répétitives telles que le tri des CV et la présélection des candidats. Cela permet aux recruteurs de se concentrer sur des tâches à plus forte valeur ajoutée, comme les entretiens en profondeur et la stratégie de recrutement. De plus, l'utilisation de chatbots et d'algorithmes d'intelligence artificielle peut réduire les biais humains, en évaluant les compétences des candidats de manière plus objective et basée sur des critères prédéfinis. Enfin, l'automatisation peut améliorer l'expérience des candidats en fournissant des réponses rapides et précises à leurs questions et en rendant le processus de candidature plus fluide et interactif.

1.3 Contexte du projet

Le recrutement et la gestion des ressources humaines sont des éléments cruciaux pour assurer le bon fonctionnement de toute organisation. Il est essentiel de bien sélectionner les meilleurs candidats pour un poste donné, ce qui implique une maîtrise approfondie des techniques de recherche et de sélection des candidats qualifiés, l'évaluation des compétences, et la prise de décisions éclairées en matière d'embauche.

Traditionnellement, ces tâches reposaient sur l'intervention humaine, ce qui, bien que souvent efficace, pouvait entraîner des erreurs, des coûts élevés, des délais prolongés et des risques de biais subjectifs inhérents à la nature humaine.

Avec l'avènement de la transformation numérique, l'intelligence artificielle (IA) et les technologies de traitement du langage naturel (NLP) se sont révélées être des solutions prometteuses pour automatiser et optimiser ces processus. Les chatbots, en particulier, offrent la possibilité de mener des entretiens, de répondre aux questions des candidats et d'évaluer leurs réponses de manière autonome.

Dans ce contexte, notre projet vise à développer un assistant d'entretien RH basé sur l'IA, capable d'interagir avec les candidats et d'évaluer leurs réponses de manière automatisée et objective.

1.4 Présentation de la société

MintHR[31] est une entreprise innovante spécialisée dans les solutions logicielles de gestion des ressources humaines (RH) au Maroc. Son objectif principal est de simplifier et d'optimiser les processus RH pour les petites et moyennes entreprises (PME) ainsi que les grandes entreprises. MintHR offre une plateforme complète et intégrée qui comprend plusieurs fonctionnalités essentielles :

- **Gestion des données RH** : Centralisation et sécurisation des informations des employés.
- **Suivi des absences** : Automatisation du suivi des congés et absences.
- **Formation des employés** : Planification et suivi des programmes de formation.
- **Recrutement et intégration** : Simplification des processus de recrutement et d'intégration des nouveaux employés. En intégrant ces fonctionnalités, MintHR vise à réduire les tâches administratives, améliorer l'engagement des employés et fournir aux gestionnaires des outils efficaces pour une prise de décision éclairée.



FIGURE 1 – Logo de l'entreprise

1.5 Problématique

Le recrutement est souvent un processus complexe et gourmand en temps, où l'intervention humaine peut parfois introduire des biais et des erreurs, affectant la qualité et l'efficacité de l'évaluation des candidats. Nous nous trouvons donc face à un défi majeur : **Comment pouvons-nous transformer le processus d'entretien et d'évaluation des candidats pour qu'il soit non seulement plus rapide et économique, mais aussi équitable et précis ?**

Notre objectif est de développer une solution qui automatise les entretiens d'embauche tout en préservant, voire en améliorant, la qualité de l'évaluation des compétences des candidats. Pour ce faire, nous nous posons les questions suivantes :

- 1. Comment un chatbot peut-il mener un entretien d'embauche de manière autonome tout en capturant efficacement les nuances des réponses des candidats ?**
- 2. Quels mécanismes peuvent être mis en œuvre pour évaluer automatiquement la pertinence et la qualité des réponses, assurant ainsi une analyse approfondie et objective ?**
- 3. Comment pouvons-nous réduire les biais dans l'évaluation des candidats pour garantir une évaluation équitable et impartiale, et ainsi promouvoir la diversité et l'inclusion ?**

En abordant ces questions, nous visons à créer un système qui non seulement optimise le processus de recrutement, mais qui offre également une expérience de candidature juste et transparente.

1.6 Cibles du projet

Ce projet a pour but de transformer le processus de recrutement en répondant aux défis identifiés avec des solutions innovantes et pratiques. Nos objectifs sont les suivants :

- **Automatiser les entretiens** : Nous souhaitons créer un chatbot intelligent capable de mener des entretiens de manière autonome, posant des questions pertinentes et adaptées à chaque candidat. L'idée est de libérer du temps précieux pour les recruteurs tout en assurant des interactions cohérentes et efficaces.
- **Évaluer les réponses automatiquement** : En utilisant des techniques avancées de traitement du langage naturel (NLP), notre objectif est d'évaluer la qualité et la pertinence des réponses des candidats de manière précise et objective. Nous voulons nous assurer que chaque réponse est analysée en profondeur pour fournir une évaluation juste.
- **Réduire les biais** : Nous mettrons en place des algorithmes conçus pour minimiser les biais dans l'évaluation des candidats. L'objectif est de garantir que les décisions sont basées sur des critères objectifs et impartiaux, promouvant ainsi une évaluation équitable pour tous.
- **Améliorer l'efficacité** : En automatisant les tâches répétitives et en fournissant des évaluations instantanées, nous visons à rendre le processus de recrutement plus rapide et plus efficace. Cela permettra aux recruteurs de se concentrer sur des tâches plus stratégiques et importantes.
- **Faciliter l'utilisation** : Nous voulons offrir une expérience utilisateur fluide et agréable, tant pour les recruteurs que pour les candidats. En concevant une interface intuitive, nous nous assurons que l'interaction avec le chatbot est simple et accessible pour tout le monde.

Chapitre 1

État de l'Art

1.1 Recrutement automatisé



FIGURE 2 – Recrutement Automatisé

Le recrutement automatisé marque une étape importante dans l'évolution des ressources humaines, en utilisant les technologies modernes pour transformer et améliorer le processus de recrutement. Contrairement aux méthodes traditionnelles qui reposent principalement sur l'intervention humaine pour évaluer et sélectionner les candidats, le recrutement automatisé adopte des outils technologiques innovants pour rendre le processus plus efficace et équitable. En intégrant ces technologies, nous pouvons non seulement réduire les biais et accélérer la sélection des candidats, mais aussi offrir une expérience plus fluide et transparente pour tous les participants. Cette approche vise à rationaliser les opérations de recrutement, tout en permettant aux équipes RH de se concentrer sur des tâches plus stratégiques et créatives.

1.1.1 Technologies et outils

Au cœur du recrutement automatisé se trouvent les Systèmes de Gestion des Candidatures (ATS), qui automatisent des tâches clés comme le tri des CV, la gestion des entretiens et la communication avec les candidats. Les ATS permettent de filtrer les candidatures en fonction de mots-clés spécifiques, ce qui réduit considérablement le nombre de candidatures à examiner manuellement et améliore la gestion du flux de travail. En parallèle, des algorithmes d'intelligence artificielle (IA) sont utilisés pour analyser les CV et faire correspondre les compétences des candidats aux exigences des postes, rendant le processus de sélection plus rapide et plus précis. Les chatbots de recrutement, quant à eux, interagissent avec les candidats pour répondre à leurs questions et mener des pré-interviews, facilitant ainsi la présélection.

– **Applicant tracking systems (ATS) :**



Figure 3 – ATS

Les ATS [33] sont des logiciels conçus pour automatiser le processus de recrutement en gérant les candidatures. Ils filtrent les CV en fonction de mots-clés et de critères définis, ce qui permet aux recruteurs de se concentrer sur les candidats les plus qualifiés.

– **Chatbots :**



Figure 4 – Les chatbots dans les domaines de recrutement

Les chatbots automatisent les interactions initiales avec les candidats, en répondant à leurs questions, en menant des entretiens préliminaires, et en programmant des entretiens. Exemples de chatbots populaires : Mya, Olivia, et XOR.

– **Algorithmes d'IA :**



FIGURE 5 – l’AI dans le recrutement

Les algorithmes d’intelligence artificielle [33] analysent les CV, évaluent les compétences des candidats et les font correspondre aux exigences des postes. Ils peuvent également prédire la réussite des candidats dans leurs futurs rôles.

1.1.2 Avantages du recrutement automatisé

Le principal atout du recrutement automatisé est indéniablement le gain de temps. En automatisant les tâches répétitives et souvent chronophages, comme le tri des CV, les entreprises peuvent se libérer des tâches administratives pour se concentrer sur des aspects plus stratégiques et créatifs du recrutement. Cette méthode aide également à atténuer les biais inconscients en se basant sur des critères objectifs plutôt que sur des jugements subjectifs, favorisant ainsi une sélection plus équitable des candidats. En outre, l’automatisation permet de gérer efficacement le volume élevé de candidatures, en identifiant rapidement les candidats les plus qualifiés et en simplifiant le processus de sélection. Cela se traduit par une plus grande précision et une meilleure gestion du recrutement, tout en offrant une expérience plus fluide et objective pour tous les candidats.

- **Gain de Temps** : L’automatisation des tâches répétitives comme le tri des CV permet de libérer du temps pour les recruteurs, leur permettant de se concentrer sur des tâches plus stratégiques et créatives.
- **Réduction des Biais** : En se basant sur des critères objectifs plutôt que sur des jugements subjectifs, les systèmes automatisés aident à atténuer les biais inconscients, favorisant ainsi une sélection plus équitable des candidats.
- **Efficacité Améliorée** : Les outils automatisés gèrent efficacement le volume élevé de candidatures, identifiant rapidement les candidats les plus qualifiés et simplifiant le processus de sélection.

1.1.3 Défis et limitations

Bien que le recrutement automatisé offre de nombreux avantages, il n’est pas sans défis. L’un des principaux risques est celui des biais algorithmiques. Les systèmes automatisés peuvent parfois reproduire ou même amplifier les biais présents dans les données avec lesquelles ils ont été formés, ce qui peut nuire à l’équité du processus. De plus, une dépendance accrue à la technologie peut affecter négativement l’expérience des candidats si les résultats générés ne sont pas correctement interprétés ou si l’interaction humaine est insuffisante. Il est également crucial d’adapter les outils automatisés aux besoins spécifiques de chaque entreprise pour éviter les erreurs de sélection et garantir que le processus reste pertinent et efficace. Ces défis nécessitent une attention constante pour s’assurer que le recrutement automatisé reste une solution bénéfique et équitable.

- **Biais algorithmiques** : Les systèmes automatisés peuvent parfois reproduire ou amplifier les biais présents dans les données avec lesquelles ils ont été formés, ce qui peut nuire à l’équité du processus.
- **Dépendance technologique** : Une dépendance accrue à la technologie peut affecter négativement l’expérience des candidats si les résultats générés ne sont pas correctement interprétés ou si l’interaction humaine est insuffisante.

- **Adaptation aux besoins spécifiques** : Il est crucial d'adapter les outils automatisés aux besoins spécifiques de chaque entreprise pour éviter les erreurs de sélection et garantir que le processus reste pertinent et efficace.

1.1.4 Études de cas et exemples

Des entreprises pionnières comme Google et IBM ont réussi à intégrer des systèmes de recrutement automatisés avec succès. Google, par exemple, a mis en place des outils sophistiqués pour analyser les CV, ce qui a considérablement amélioré la précision et l'efficacité de son processus de recrutement. De son côté, IBM utilise des chatbots pour mener des entretiens préliminaires, simplifiant ainsi la première étape du recrutement. Ces approches ont non seulement réduit de manière significative les délais de recrutement, mais ont également permis d'attirer des candidats de meilleure qualité. Les expériences de ces géants de la technologie offrent des leçons précieuses pour d'autres organisations cherchant à optimiser leur processus de recrutement et à tirer parti des technologies modernes.

- **Google** : Google utilise des outils sophistiqués pour analyser les CV, ce qui a considérablement amélioré la précision et l'efficacité de son processus de recrutement.
- **IBM** : IBM utilise des chatbots pour mener des entretiens préliminaires, simplifiant ainsi la première étape du recrutement. Ces approches ont réduit les délais de recrutement et permis d'attirer des candidats de meilleure qualité.

1.1.5 Tendances futures

Les innovations technologiques continuent de révolutionner le recrutement automatisé. Les récentes avancées en intelligence artificielle et en apprentissage automatique permettent de créer des outils de plus en plus sophistiqués, capables de prédire avec une grande précision la réussite des candidats dans leurs futurs rôles. Parmi les nouvelles tendances, l'utilisation de l'IA pour des évaluations psychométriques plus détaillées permet de mieux comprendre les compétences et les traits de personnalité des candidats. En parallèle, l'intégration de la réalité virtuelle pour des simulations d'entretiens immersifs offre une approche novatrice et engageante. Ces développements promettent de rendre le processus de recrutement non seulement plus efficace, mais aussi plus personnalisé, offrant aux entreprises de nouvelles façons d'attirer et de sélectionner les meilleurs talents, tout en enrichissant l'expérience des candidats.

- **Analyse prédictive** : Utilisation de l'IA pour prévoir les besoins en recrutement, les taux de rétention des employés et identifier les employés à haut potentiel.
- **Assistants virtuels avancés** : Développement de chatbots capables de mener des conversations plus naturelles et contextuelles grâce à l'apprentissage profond et aux avancées en NLP.
- **Réalité virtuelle pour les simulations d'entretiens** : Intégration de la réalité virtuelle pour offrir des simulations d'entretiens immersifs, enrichissant l'expérience des candidats et facilitant une évaluation plus précise.

1.1.6 Chatbot RH populaires

De nombreuses entreprises ont développé des chatbots RH pour améliorer leurs processus de gestion des talents. Voici quelques exemples notables :

- **Mya** :



FIGURE 6 – Mya chatbot

Mya est un chatbot [34] alimenté par l'intelligence artificielle qui aide à automatiser les tâches de recrutement. Il peut mener des entretiens préliminaires, répondre aux questions des candidats et programmer des entretiens. Mya utilise des algorithmes avancés de traitement du langage naturel (NLP) pour comprendre et répondre aux questions des candidats de manière pertinente.

— **Olivia** :



FIGURE 7 – Olivia chatbot

Développé par Paradox[35], Olivia est un assistant virtuel qui automatise le processus de recrutement en engageant les candidats par chat, en répondant à leurs questions et en facilitant la planification des entretiens. Olivia est utilisé par de nombreuses grandes entreprises pour améliorer l'expérience des candidats et réduire le temps de recrutement.

— **XOR** :



FIGURE 8 – Xor chatbot

XOR est une plateforme de chatbot RH [36] qui utilise l'IA pour automatiser le recrutement, l'onboarding et la gestion des employés. Il aide les recruteurs à trouver, engager et suivre les candidats à travers des conversations automatisées.

1.1.7 Tendances et innovations dans les technologies RH

Les avancées technologiques continuent de transformer les RH. Voici quelques tendances émergentes :

- **Analyse prédictive** : Utilisation de l'IA pour prévoir les besoins en recrutement, les taux de rétention des employés et identifier les employés à haut potentiel.
- **Chatbots conversationnels avancés** : Développement de chatbots capables de mener des conversations plus naturelles et contextuelles grâce à l'apprentissage profond et aux avancées en NLP.
- **Intégration avec des systèmes existants** : Solutions permettant une intégration transparente des chatbots RH avec les systèmes de gestion des ressources humaines (HRIS) et d'autres outils d'entreprise.

1.2 Chatbots et IA dans les RH

Les chatbots et l'intelligence artificielle (IA) occupent une place de plus en plus importante dans la gestion des ressources humaines (RH), redéfinissant les méthodes de recrutement, de gestion et d'interaction avec les employés. Cette section examine comment ces technologies innovantes sont intégrées dans les processus RH pour optimiser l'efficacité, réduire les coûts et offrir une expérience enrichie tant pour les candidats que pour les employés. En automatisant les tâches routinières et en fournissant des analyses approfondies, l'IA et les chatbots apportent une nouvelle dimension à la gestion des talents, facilitant des interactions plus fluides et personnalisées et contribuant à une gestion des ressources humaines plus agile et réactive.

1.2.1 Recrutement et sélection

Les chatbots sont désormais des alliés incontournables dans le processus de recrutement, offrant des solutions innovantes pour automatiser les interactions initiales avec les candidats. Ces outils peuvent gérer des tâches variées telles que répondre aux questions fréquemment posées, recueillir des informations de base et pré-sélectionner les candidatures. Grâce à l'intelligence artificielle, les chatbots vont encore plus loin en analysant les CV et les lettres de motivation, en évaluant les compétences et les expériences des candidats, et

en les comparant aux exigences des postes à pourvoir. Des chatbots comme Mya et Olivia illustrent parfaitement cette avancée, en facilitant la gestion des candidatures grâce à leurs capacités à mener des entretiens préliminaires et à organiser des rendez-vous pour des entretiens humains. Cette automatisation non seulement améliore l'efficacité du processus de recrutement mais offre également une expérience plus fluide pour les candidats.

1.2.2 Onboarding et formation

Les chatbots jouent également un rôle crucial dans l'intégration des nouveaux employés (onboarding). Ils fournissent des informations importantes sur les politiques de l'entreprise, les procédures administratives et les ressources disponibles, tout en répondant aux questions des nouveaux venus. Cette approche permet d'améliorer l'efficacité du processus d'intégration et de réduire le stress des nouveaux employés. En outre, les solutions d'IA sont utilisées pour créer des programmes de formation personnalisés en fonction des besoins individuels des employés, en utilisant des plateformes d'apprentissage adaptatif qui ajustent le contenu en fonction des performances et des préférences des utilisateurs.

1.2.3 Gestion des performances et développement

L'intelligence artificielle joue un rôle clé dans la gestion des performances en offrant des analyses détaillées et pertinentes sur les performances des employés. Grâce à des outils d'analyse avancés, l'IA peut examiner divers indicateurs de performance, déceler des tendances et des opportunités de développement, et proposer des recommandations personnalisées pour améliorer les compétences et les résultats. Ces technologies permettent aux gestionnaires de prendre des décisions éclairées basées sur des données objectives plutôt que sur des impressions subjectives. En intégrant ces insights dans leurs stratégies, les entreprises peuvent élaborer des plans de développement personnel plus ciblés et efficaces, contribuant ainsi à l'amélioration continue des performances et à la croissance professionnelle des employés.

1.2.4 Engagement des employés et support

Les chatbots et l'IA sont également utilisés pour améliorer l'engagement des employés en offrant un soutien instantané et accessible. Les chatbots peuvent recueillir des feedbacks anonymes, mesurer la satisfaction des employés et détecter les signes de démotivation ou de stress. Par ailleurs, les systèmes d'IA peuvent analyser ces données pour fournir des recommandations aux gestionnaires sur les actions à entreprendre pour améliorer l'engagement et le bien-être des employés.

1.2.5 Défis et considérations éthiques

L'intégration des chatbots et de l'IA dans les RH soulève plusieurs défis et questions éthiques. Les biais algorithmiques peuvent influencer les décisions de recrutement et de gestion des performances, et il est crucial de garantir que les systèmes d'IA soient conçus de manière à minimiser ces biais. La confidentialité des données est également une préoccupation majeure, car l'utilisation de l'IA implique le traitement de grandes quantités d'informations personnelles. Les entreprises doivent veiller à respecter les réglementations en matière de protection des données et à assurer la transparence dans l'utilisation des technologies d'IA.

1.2.6 Tendances futures

L'avenir des chatbots et de l'intelligence artificielle dans les ressources humaines est très prometteur, avec l'émergence de technologies innovantes telles que les assistants virtuels avancés et les systèmes d'analyse prédictive. Les progrès récents en traitement du langage naturel (NLP) et en apprentissage profond permettront de développer des solutions encore plus intelligentes et interactives. Ces technologies seront capables de comprendre et d'anticiper les besoins des employés avec une précision accrue, offrant ainsi une expérience plus personnalisée et réactive. En facilitant une automatisation plus poussée des tâches opérationnelles, ces avancées permettront aux professionnels des RH de se concentrer davantage sur les aspects stratégiques et humains de leur rôle, tout en maximisant l'efficacité des processus quotidiens.

1.2.7 Solutions d'intégration et de formation des employés

Les chatbots ne se limitent pas au recrutement ; ils sont également utilisés pour l'onboarding et la formation des nouveaux employés. Voici quelques exemples de solutions existantes :

– **Talla** :



FIGURE 9 – Talla—an AI-driven platform

Talla [37] utilise l'IA pour automatiser l'onboarding et la gestion des connaissances. Il fournit des réponses instantanées aux questions des employés, aide à la formation continue et centralise les informations importantes sur l'entreprise.

– **LMS365** :



FIGURE 10 – Lms365

LMS365 est une solution de gestion de l'apprentissage intégrée à Microsoft 365. Elle utilise des chatbots pour fournir des formations personnalisées, suivre les progrès des employés et adapter les programmes de formation en fonction des besoins individuels.

1.2.8 Outils d'évaluation des performances

L'évaluation des performances est un domaine clé où l'IA apporte des innovations significatives. Voici quelques outils populaires :

– **Reflektive** :



FIGURE 11 – Reflektive

Reflektive [38] est une plateforme de gestion des performances qui utilise l'IA pour fournir des feedbacks en temps réel, suivre les objectifs et analyser les performances des employés. Elle aide les gestionnaires à identifier les forces et les faiblesses des employés et à planifier des actions de développement.

– **Betterworks** :

betterworks

FIGURE 12 – betterworks

Betterworks [39] propose une solution d'IA pour la gestion des objectifs et des performances. Elle aide les entreprises à définir et suivre les objectifs des employés, à fournir des feedbacks continus et à analyser les données de performance pour prendre des décisions éclairées.

1.2.9 Outils Tendances et innovations dans les technologies RH

Les avancées technologiques continuent de transformer les RH. Voici quelques tendances émergentes :

- **Analyse prédictive** : Utilisation de l'IA pour prévoir les besoins en recrutement, les taux de rétention des employés et identifier les employés à haut potentiel.
- **Chatbots conversationnels avancés** : Développement de chatbots capables de mener des conversations plus naturelles et contextuelles grâce à l'apprentissage profond et aux avancées en NLP.
- **Intégration avec des systèmes existants** : Solutions permettant une intégration transparente des chatbots RH avec les systèmes de gestion des ressources humaines (HRIS) et d'autres outils d'entreprise.

Chapitre 2

Analyse des besoins

L'analyse des besoins est une étape cruciale dans la conception de tout système, permettant de comprendre et de définir les exigences des utilisateurs finaux ainsi que les contraintes techniques. Pour ce projet, j'ai identifié deux principaux acteurs : le candidat et le responsable RH. Chacun de ces acteurs a des besoins spécifiques que je vais détailler ci-dessous.

1.1 Analyse fonctionnelle

1.1.1 Identification des utilisateurs

1. Candidat :

- **Objectif** : Postuler à une offre d'emploi et passer un test automatisé via un chatbot.
- **Actions possibles** :
 - Créer un compte et se connecter.
 - Parcourir et postuler aux offres d'emploi disponibles.
 - Passer des tests d'évaluation via le chatbot.
 - Recevoir des notifications et des résultats des tests.
 - Consulter son profil et modifier ses informations personnelles.

2. Responsable RH

- Créer un compte et se connecter.
- Visualiser les candidatures et les scores des candidats.
- Filtrer les candidats selon un seuil de score fixé pour chaque job.
- Ajouter une nouvelle offre d'emploi et supprimer ou modifier une autre.
- Consulter son Profil et modifier ses informations personnelles.
- Planification du deuxième entretien .

1.1.2 Scénarios d'utilisation

1. Candidat postule à une offre :

- Candidat parcourt les offres d'emploi disponibles. se connecte à son compte.
- Il choisit une offre et postule.
- On lui demande de se connecter à son compte avant de postuler.
- Il passe un test via le chatbot intégré.
- Il reçoit une notification avec les résultats du test.

2. Responsable RH gère les offres et candidats :

- Le responsable RH se connecte à son compte.
- Il ajoute une nouvelle offre d'emploi en précisant les détails et le seuil de score.
- Il visualise les candidats qui ont postulé.
- Il filtre les candidats selon le score obtenu pour chaque offre.
- Il invite les candidats qui vérifient les critères posés pour ce job à un deuxième entretien.
- Il modifie ou supprime une offre d'emploi.

1.2 Analyse technique

1.2.1 Contraintes techniques

1. **Sécurité des données :**
 - Mise en place de protocoles de sécurité pour protéger les informations personnelles des utilisateurs.
2. **Performance et scalabilité :**
 - Assurance que le système peut gérer un grand nombre de candidatures et de tests simultanément.
 - Optimisation des performances du chatbot pour des interactions rapides et fluides.
3. **Intégration de l'IA :**
 - Extraction des mots-clés à partir de la description du poste pour générer des questions posées par le bot.
 - Génération de réponses idéales pour les utiliser lors de l'attribution des scores aux réponses des candidats.
 - Utilisation de modèles de traitement du langage naturel (NLP) pour évaluer automatiquement les réponses des candidats.
 - Mise à jour régulière des algorithmes pour améliorer la précision des évaluations.

1.2.2 Choix technologiques

Pour la réalisation de ce projet, plusieurs technologies ont été sélectionnées en fonction de leurs capacités à répondre aux besoins fonctionnels et techniques identifiés. Voici un aperçu des choix technologiques effectués :

- **Backend :**
 - **Framework :** Django [41] a été choisi comme framework backend en raison de sa robustesse, de sa sécurité intégrée, et de sa capacité à accélérer le développement grâce à son approche batteries-included.



FIGURE 13 – Django Framework

- **Langage de programmation :** Python, connu pour sa simplicité [42] et sa large adoption dans le domaine de l'intelligence artificielle et du développement web.



FIGURE 14 – Python framework

- **Base de données :** PostgreSQL [43] a été choisi pour ses performances, sa fiabilité et ses fonctionnalités avancées adaptées aux applications de grande envergure.



FIGURE 15 –PostgreSQL Database

— **Frontend :**

- **Framework :** React.js a [44] été choisi pour développer une interface utilisateur réactive et dynamique, permettant une expérience utilisateur fluide et interactive.

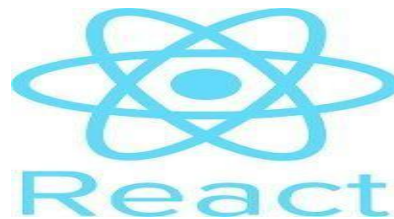


FIGURE 16 –Re a c t

- **Bibliothèque de composants :** Shadcn [45] a été utilisé pour intégrer des composants de qualité et uniformiser l'apparence de l'application.



FIGURE 17 –Sh a d c n

- **Langage de programmation :** JavaScript (ES6+) [46] et JSX pour la création des composants React.



FIGURE 18 –JavaScript

— **Chatbot et IA :**

- **Framework NLP :** Rasa a été utilisé pour le développement du chatbot, en raison de ses capacités avancées de traitement du langage naturel et de son flexibilité pour la création de dialogues complexes.
- **Modèles de machine learning :** Utilisation de modèles NLP pour l'analyse des réponses des candidats et l'attribution des scores basés sur la pertinence des réponses par rapport aux questions posées. En particulier, le modèle T5 a été utilisé pour la génération des questions, ChatGPT-2 pour la génération des réponses idéales, et KeyBERT pour l'extraction des mots clés.

Ces choix technologiques ont été guidés par des critères de performance, de sécurité, de scalabilité et de

compatibilité avec les besoins du projet. L'utilisation conjointe de Django, React.js, Shaden, et Rasa permet de créer une application complète, allant de la gestion des offres d'emploi à l'évaluation automatisée des candidats via un chatbot intelligent.

1.3 Planification du projet

1.4 Diagramme de Gantt



FIGURE 19 – Diagramme de Gant

Au cours de l'élaboration de ce projet, plusieurs tâches majeures ont été réalisées de manière séquentielle. Une documentation approfondie sur les frameworks proposés a été effectuée pour choisir les technologies les plus appropriées. Le prétraitement des données a été suivi de l'extraction des mots-clés des descriptions de poste. Ensuite, les questions ont été générées en utilisant ces mots-clés, et des réponses idéales ont été formulées. Le chatbot a été développé avec Rasa, puis intégré au backend développé en Django et PostgreSQL. Enfin, l'interface utilisateur a été créée avec React.js et Shaden, et un modèle de machine learning a été entraîné pour évaluer automatiquement les réponses des candidats.

Chapitre 3

Développement

1.1 Configuration de l'environnement de développement

Pour la réalisation de ce projet, plusieurs outils et configurations ont été mis en place afin de créer un environnement de développement efficace. Les principales étapes incluent l'installation de Python, Django, PostgreSQL, React.js, Shadcn et Rasa. Chaque technologie a été choisie pour sa capacité à répondre aux besoins spécifiques du projet, et leur installation a été effectuée en respectant les versions recommandées pour assurer la compatibilité et la stabilité de l'application.

1.2 Développement du backend

1.2.1 Base de données

La base de données PostgreSQL a été choisie pour sa robustesse et ses fonctionnalités avancées. Les modèles de données ont été définis pour représenter les différentes entités du projet, telles que les offres d'emploi, les candidats, et leurs réponses. Des relations entre les tables ont été établies pour refléter les interactions entre ces entités et permettre une gestion efficace des données. Voici une capture d'écran du schéma de la base de données PostgreSQL, montrant les relations entre les tables principales comme Jobs, User, JobQuestions :

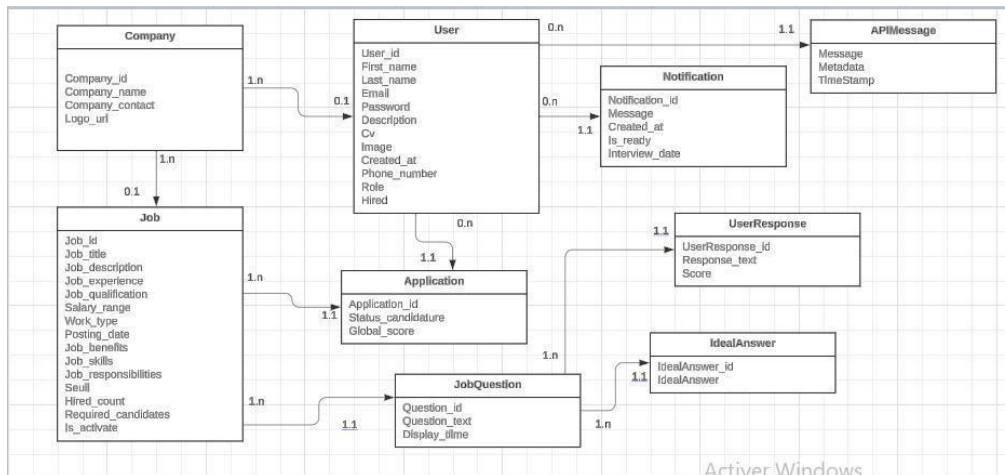


FIGURE 20 – MCD

1.3 Développement du frontend

1.3.1 Frameworks et bibliothèques utilisés

Le frontend a été développé en utilisant React.js pour créer une interface utilisateur réactive et dynamique. Shadcn a été utilisé pour intégrer des composants de haute qualité et uniformiser l'apparence de l'application. Ces outils permettent de créer une expérience utilisateur fluide et interactive, essentielle pour l'engagement des candidats et des gestionnaires RH.

1.4 Interface Utilisateur

1.4.1 Candidats

Page d'accueil



FIGURE 21 – Page d'accueil

La première interface de notre projet est la page d'accueil, qui permet aux utilisateurs de naviguer vers d'autres pages via une barre de navigation. Cette barre de navigation les dirige vers la visualisation des offres d'emploi, la page de connexion, ou la création d'un nouveau compte. Cette page d'accueil est très simple et épurée, ne contenant pas d'informations complexes. Elle offre néanmoins une vue d'ensemble sur le contenu de notre application, permettant aux utilisateurs de comprendre facilement ce que nous proposons.

Interface d'Inscription

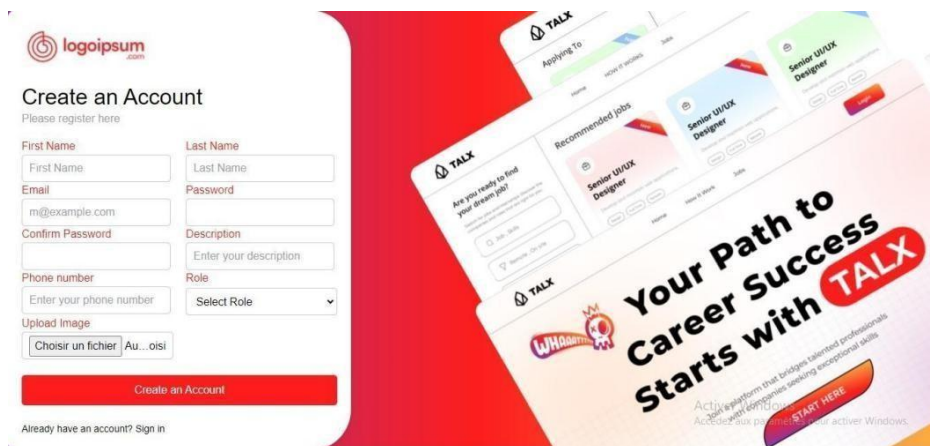


FIGURE 22 – Page d'inscription

L'interface suivante est celle de la création d'un compte. Dans cette section, l'utilisateur est invité à entrer ses informations personnelles, telles que son nom, son adresse e-mail, et son mot de passe. Il doit également préciser son rôle, soit en tant que candidat (une personne postulant à un emploi et passant un entretien), soit en tant que professionnel, c'est-à-dire un administrateur ou un membre de l'équipe RH.

Si l'utilisateur choisit le rôle de candidat, un champ supplémentaire apparaîtra pour lui permettre de télécharger son CV. Cette fonctionnalité garantit que les candidats puissent facilement fournir leurs informations professionnelles complètes, facilitant ainsi le processus de recrutement. Pour les professionnels, les informations requises se concentreront davantage sur leur rôle et leurs responsabilités au sein de l'organisation.

Interface de connexion

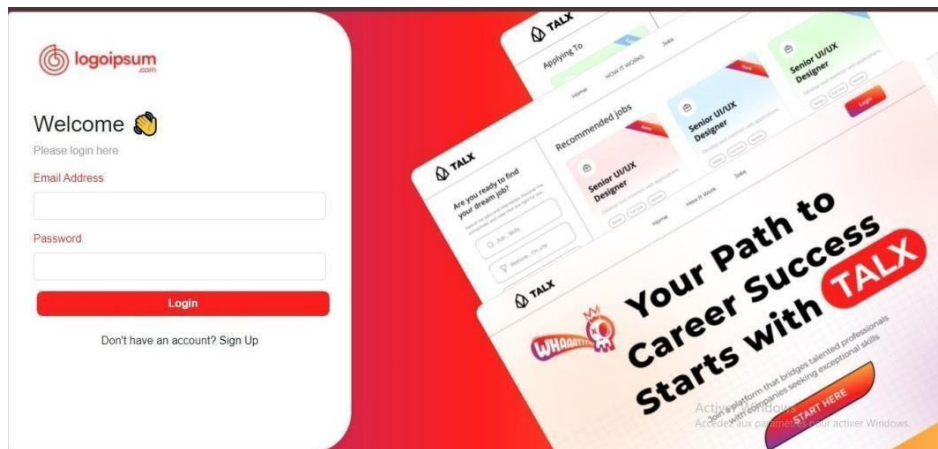


FIGURE 23 – Page de login

L'interface suivante est celle de la connexion. Ici, l'utilisateur accède à son compte en utilisant les informations fournies lors de la création de son compte. Une fois connecté, l'utilisateur est redirigé vers une interface dédiée où il peut postuler à des offres d'emploi et passer des entretiens. Cette page est conçue pour offrir une navigation fluide et permettre aux utilisateurs de gérer leurs candidatures et leurs entretiens de manière efficace.

L'interface des offres

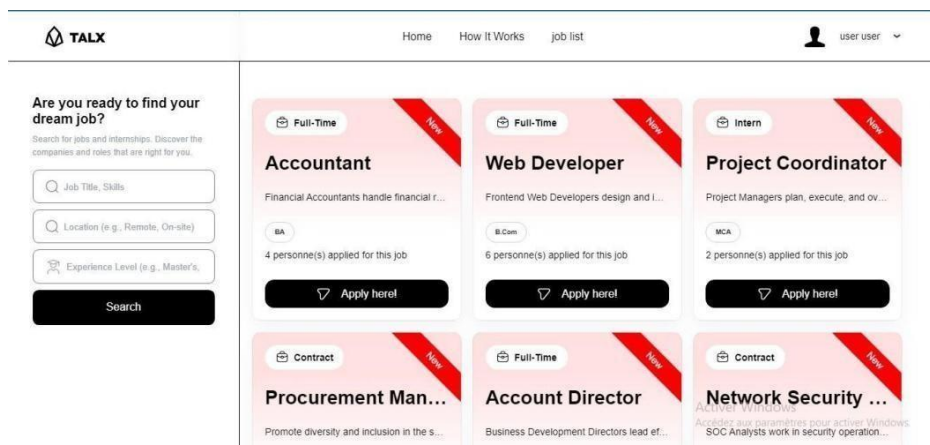
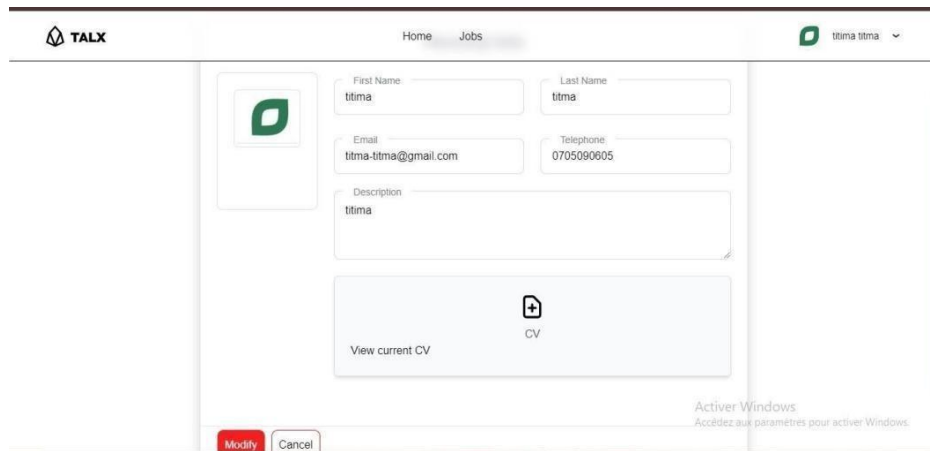


FIGURE 24 – Interface des offres

L'interface de liste des offres d'emploi est accessible sans connexion, permettant aux utilisateurs de consulter les opportunités disponibles. Cependant, si un utilisateur souhaite postuler à une offre, il sera redirigé vers la page de connexion, car le processus de candidature nécessite une authentification. Cette interface permet également aux utilisateurs de rechercher rapidement des offres en utilisant des critères tels que le nom de

l'offre, la localisation, les compétences requises, ou les expériences nécessaires, afin de trouver les opportunités qui correspondent le mieux à leur profil sans avoir à parcourir toutes les offres.

L'interface du profil Utilisateur



The screenshot displays the user profile page on the TALX website. The header includes the TALX logo, navigation links for 'Home' and 'Jobs', and a user profile dropdown menu showing 'tlima tlima'. The main content area contains a profile card with the following fields: First Name (tlima), Last Name (tlima), Email (tlima-tlima@gmail.com), Telephone (0705090005), and Description (tlima). Below these fields is a 'View current CV' button with a plus icon. At the bottom of the profile card are 'Modify' and 'Cancel' buttons. A Windows watermark is visible in the bottom right corner of the browser window.

FIGURE 25 – Profil utilisateur

Dans cette interface, l'utilisateur peut visualiser ses informations personnelles et a également la possibilité de les modifier. Cela lui permet de mettre à jour ses données, telles que son adresse e-mail, son numéro de téléphone, ou toute autre information pertinente, afin de maintenir ses informations à jour et précises.

1.4.2 Intégration du chatbot

Le but de notre projet vise à rendre les entretiens plus efficaces en automatisant le processus d'interaction entre les candidats et les recruteurs. Nous avons développé un bot qui initie la conversation avec le candidat, en lui posant des questions spécifiques au poste pour lequel il a postulé et en enregistrant ses réponses pour évaluation. Pour assurer une gestion optimale du temps, nous avons mis en place un système de rappel. Le bot pose une nouvelle question toutes les deux minutes et attend la réponse du candidat. Si le candidat ne répond pas dans ce délai, le bot lui rappelle gentiment que le temps imparti est écoulé et passe à la question suivante. Ainsi, notre solution garantit que chaque candidat a une chance égale de répondre tout en respectant les contraintes de temps, rendant le processus d'entretien à la fois équitable et efficace.

Utilisation de Rasa

Rasa a été utilisé pour développer le chatbot en raison de ses capacités avancées de traitement du langage naturel et de sa flexibilité pour la création de dialogues complexes. La configuration de Rasa a inclus la définition des intents et des entities, ainsi que la création des actions personnalisées pour interagir avec la base de données.

Interaction utilisateur-chatbot

L'interface utilisateur permet aux candidats d'interagir facilement avec le chatbot. Les réponses des candidats sont capturées et traitées par le chatbot, et les résultats de l'évaluation sont affichés en temps réel. Cette interaction est facilitée par des composants React qui gèrent la logique de conversation et la mise à jour dynamique de l'interface.

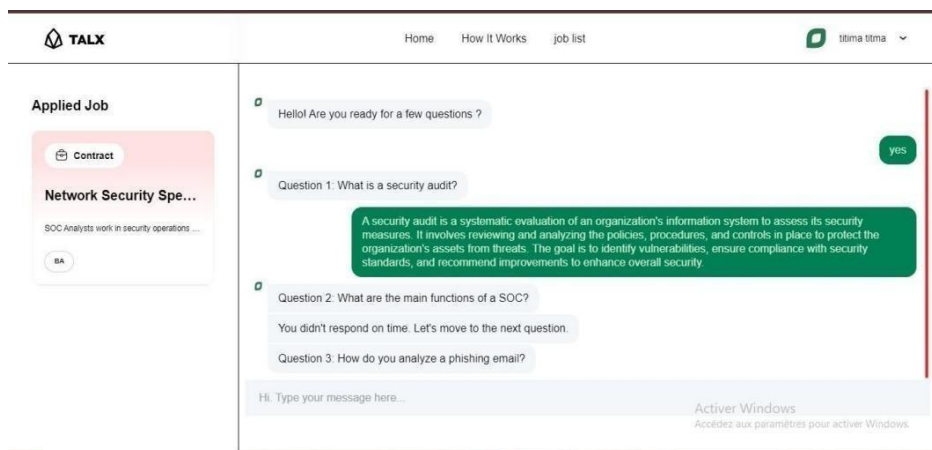


FIGURE 26 – Interaction avec le chatbot

Scénarios de dialogue et flux conversationnels

Des scénarios de dialogue ont été créés pour guider les candidats à travers le processus de candidature et d'évaluation. Les flux conversationnels ont été conçus pour assurer une interaction naturelle et efficace entre le chatbot et les utilisateurs, en couvrant toutes les étapes de l'entretien virtuel.

1.4.3 Feedback de la candidature

Une fois l'entretien soumis, un email personnalisé ou un feedback immédiat est généré. Cet email est envoyé immédiatement après la soumission de l'entretien uniquement aux candidats dont le score général est inférieur au seuil requis pour le poste. Dans cet email, nous informons courtoisement le candidat qu'il n'a malheureusement pas atteint le score nécessaire et qu'il n'est pas retenu pour le poste. L'email est rédigé de manière courtoise et encourageante, expliquant les raisons de cette décision et invitant le candidat à postuler à d'autres opportunités futures.

Dear ghali el haoumi,

Thank you for participating in the interview for the Account Director position at L&T Finance Holdings. We regret to inform you that we will not be proceeding with your application at this time. However, we appreciate your time and effort.

We wish you all the best in your future endeavors.

Sincerely,
L&T Finance Holdings

FIGURE 27 – feedback candidature

1.5 HR Team

Dans une autre partie de notre application, le groupe RH gère les candidatures. Ce groupe partage les mêmes interfaces qu'un candidat, telles que le login, l'inscription, et le profil. Cependant, il se distingue par sa capacité à visualiser les candidats qui ont réussi à obtenir un score supérieur au seuil requis. Le groupe RH peut ainsi préciser une date pour un deuxième entretien et envoyer une invitation par email aux candidats sélectionnés. De plus, les membres du groupe RH ont le droit d'ajouter ou de supprimer une offre d'emploi.

1.5.1 Gestion des Offres d'Emploi

L'interface de gestion des offres d'emploi est la première qui s'affiche après la connexion pour les membres de l'équipe RH. Cette interface est similaire à celle des candidats, mais avec des fonctionnalités supplémentaires spécifiques aux responsables RH.

1. **Liste des Offres d'Emploi** : Après la connexion, l'utilisateur accède à la liste des offres d'emploi. Pour les membres de l'équipe RH, cette interface permet de visualiser les candidatures ayant obtenu un score supérieur au seuil requis pour chaque poste.

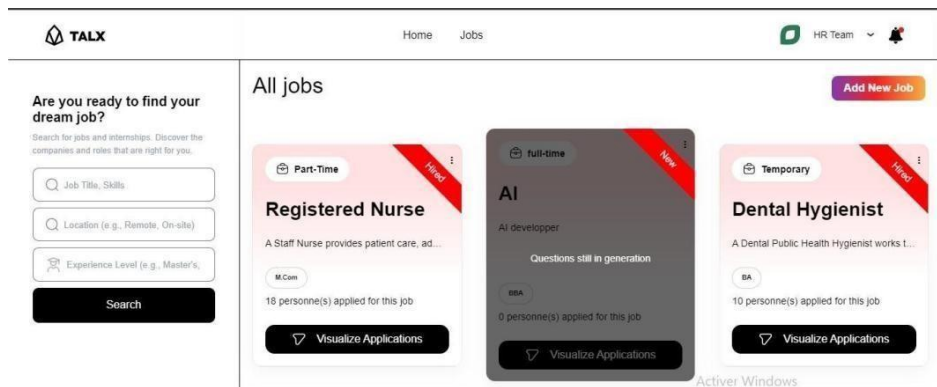


FIGURE 28 – Gestion des offres d'emploi

2. **Visualisation des Candidatures** : Les membres du groupe RH peuvent cliquer sur le bouton "Visualize Applications" pour voir les détails des candidats ayant postulé et leurs scores.

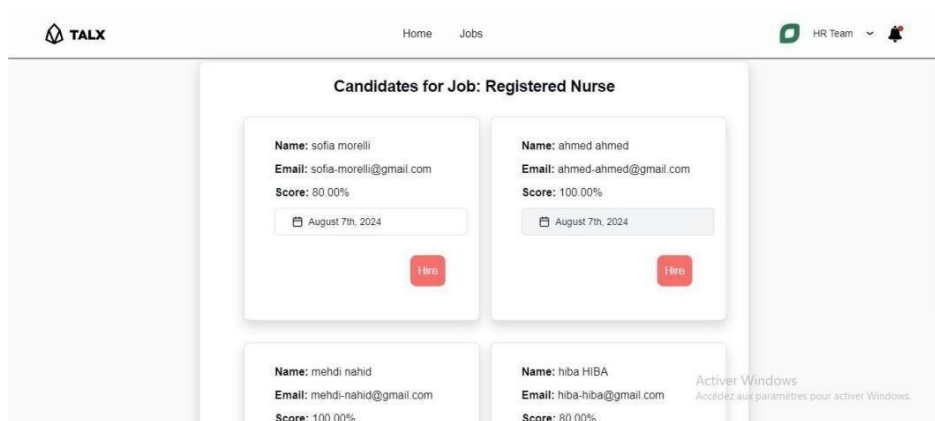


Figure 29 – candidature pour chaque job

Le groupe RH peut visualiser les candidatures pour chaque offre, fixer les dates des prochains entretiens pour les candidats sélectionnés, et envoyer des invitations par email. La date de l'entretien doit être postérieure à la date actuelle et ne doit pas tomber un week-end. En cas de non-respect de ces conditions, un message d'erreur (toast) s'affiche. Une fois la date fixée et le bouton cliqué, celui-ci devient non cliquable pour éviter d'inviter un candidat plusieurs fois.

3. **Ajout, Modification et Suppression d'Offres**
 – **Ajout d'une offre**

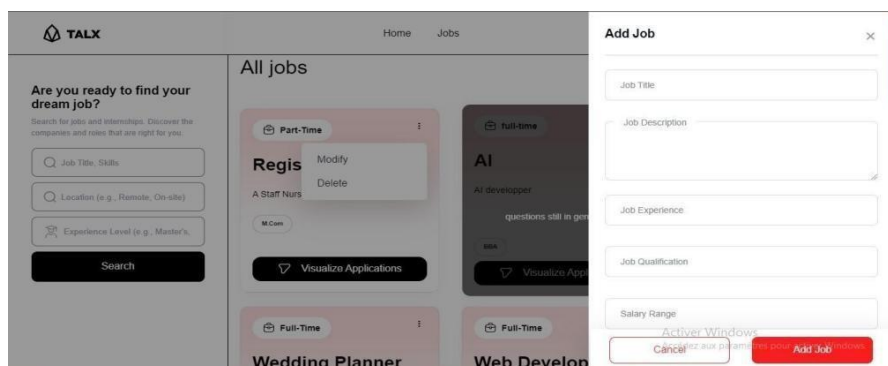


FIGURE 30 – Ajout d'une offre

Lorsqu'un utilisateur clique sur le bouton "Add Job", une barre latérale (sidebar) s'ouvre permettant de saisir les informations nécessaires pour l'offre, telles que le titre, la description, les compétences requises et d'autres champs pertinents. Une fois que l'utilisateur clique sur le bouton "Add Job", l'offre est ajoutée et s'affiche dans l'interface des offres côté RH avec un indicateur "overflow". Simultanément, les questions pour l'entretien commencent à être générées en arrière-plan dans le backend. Une fois que les questions sont générées et sauvegardées dans la base de données, l'indicateur "overflow" disparaît, permettant au RH de manipuler l'offre. À ce stade, les candidats peuvent également passer le test pour cette offre.

— Modification d'une offre

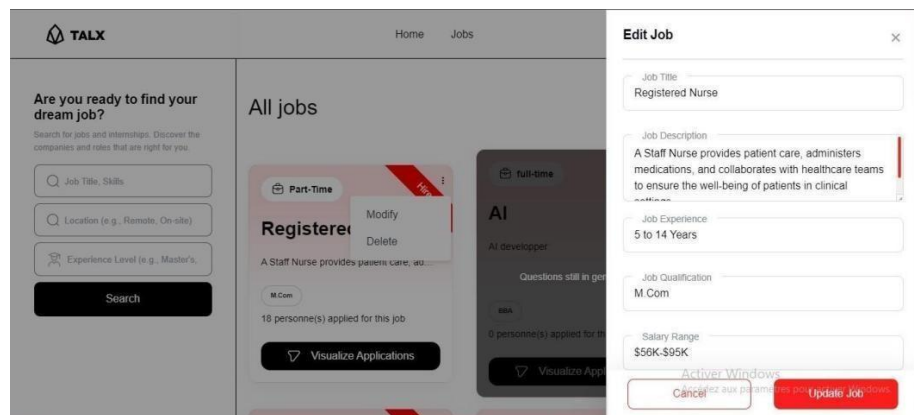


FIGURE 31 – modification d'une offre

Lorsqu'un utilisateur clique sur "Modify", une barre latérale (sidebar) remplie avec les informations de l'offre sélectionnée s'affiche, permettant à l'utilisateur de modifier n'importe quelle information pour cette offre.

— Suppression d'une offre

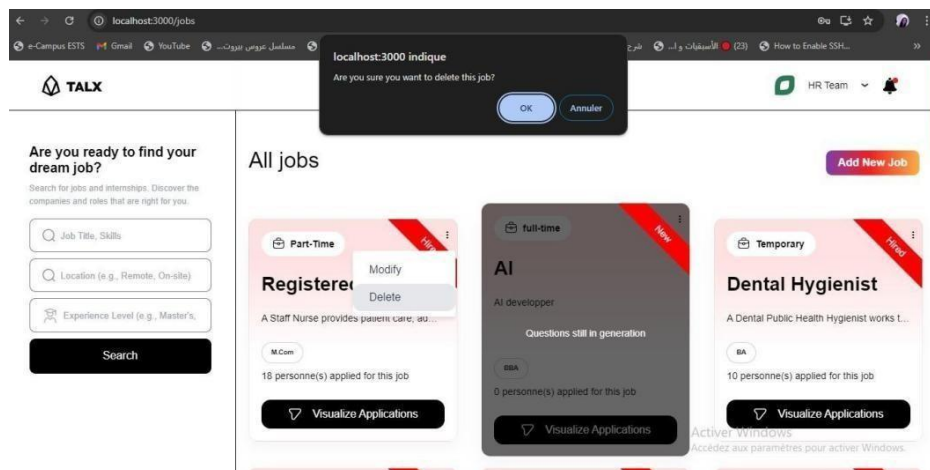


FIGURE 32 – suppression d'une offre

Lorsqu'un utilisateur clique sur "Delete", une alerte s'affiche pour confirmer la suppression. Si l'utilisateur confirme, un message de confirmation (toast) s'affiche et l'offre est supprimée.

4. **État des Offres d'Emploi** Les offres d'emploi peuvent avoir différents états :
 - **New** : Indique que l'offre vient d'être ajoutée.
 - **Questions still in generation** : Indique que les questions pour l'entretien sont en cours de génération. Cette offre n'est visible que par le groupe RH et ne sera accessible aux candidats qu'une fois les questions prêtes.
 - **Hired** : Indique que les dates des prochains entretiens ont été fixées et que le nombre de places requis pour cette offre a été atteint. L'offre n'est alors plus affichée pour les candidats.
5. **Gestion des Invitations pour le Deuxième Entretien** :



FIGURE 33 – Invitation pour le deuxième entretien

Les membres du groupe RH peuvent fixer une date pour le prochain entretien pour les candidats sélectionnés. La date doit être postérieure à la date actuelle et ne doit pas tomber un week-end, sinon un message d'erreur (toast) sera affiché. Une fois qu'un utilisateur fixe une date et clique sur le bouton, celui-ci devient non cliquable pour éviter d'inviter un candidat plusieurs fois. Des invitations personnalisées sont ensuite envoyées aux candidats par email.

6. Suivi des Entretiens via Notifications

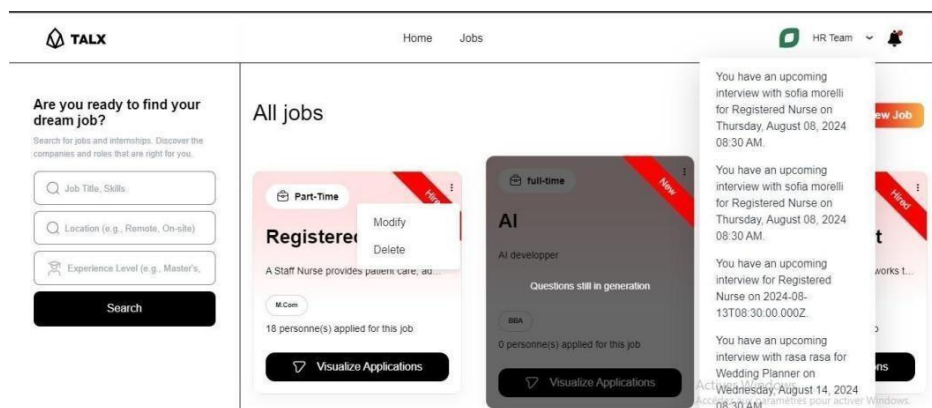


FIGURE 34 – HR Notifications

La section des notifications permet aux membres du groupe RH de suivre l'ensemble des entretiens planifiés, d'avoir une vue d'ensemble des candidats invités, et de gérer efficacement le processus de recrutement.

Chapitre 4

Modèles et frameworks utilisés

1.1 Prétraitement des données

Dans la phase de prétraitement des données du projet, une approche globale a été adoptée pour nettoyer et préparer l'ensemble des données pour l'analyse ultérieure et l'apprentissage du modèle. Ce processus est crucial pour garantir la qualité et la fiabilité des données qui alimentent notre assistant d'entretien RH basé sur l'IA.

Normalisation du texte :

L'étape initiale consiste à convertir toutes les données textuelles en minuscules afin de normaliser l'ensemble des données et d'éliminer ainsi les divergences dues à la sensibilité à la casse. Cette uniformité est essentielle pour un traitement cohérent des informations textuelles.

Suppression des balises HTML [10] :

Étant donné que les données brutes peuvent souvent contenir des balises HTML intégrées dans les champs de texte, celles-ci ont été supprimées pour s'assurer que seul le contenu textuel pertinent était conservé. Cette étape est essentielle pour nettoyer les descriptions d'emploi et autres champs de texte, qui ne doivent contenir que du texte brut pour l'analyse.

Caractères spéciaux et suppression de chiffres [10] :

Les caractères superflus, tels que les ponctuation et les chiffres, ont été supprimés du texte. Ces éléments ne contribuent pas à la compréhension du contexte ou du contenu des descriptions d'emploi et des domaines connexes. leur suppression permet de concentrer l'analyse sur les seuls mots significatifs.

Réduction des espaces blancs [10] :

Les espaces supplémentaires, y compris les tabulations et les caractères de retour à la ligne, ont été éliminés, et les espaces multiples ont été réduits à un seul espace. Cette action permet d'éviter les erreurs d'interprétation des données dues à des incohérences de formatage.

Tokenisation [10] :

La tokenisation des phrases est la procédure qui consiste à diviser le texte en unités plus petites ou tokens. Par conséquent, nous devons diviser le corpus de texte en un ensemble de phrases. Après quoi, les phrases sont encore tokenisées pour les représenter par une collection de mots. Cette opération est réalisée à l'aide des fonctions `sent_tokenize(text)` et `word_tokenize(text)` du paquetage du paquet NLTK.

Suppression des mots vides [10] :

Les mots vides sont un terme utilisé pour définir les mots et les phrases qui sont utilisés pour joindre des mots significatifs dans une phrase. Ainsi, ces mots non pertinents peuvent être facilement supprimés du

texte. Par conséquent, après avoir divisé le corpus en phrases et en mots, nous devons supprimer tous les mots non pertinents de toutes les phrases.

Stemming [10] :

Le stemming en traitement du langage naturel (NLP) consiste à réduire un mot à sa racine ou à sa forme de base. Cette méthode simplifie les mots en supprimant leurs préfixes et suffixes, tout en conservant leur sens général. Contrairement à la lemmatisation, le stemming ne garantit pas toujours que le résultat soit un mot valide, mais il vise principalement à enlever les affixes pour regrouper des mots similaires sous une même racine. Ce procédé contribue à la normalisation des textes et améliore les processus de recherche et d'analyse des données.

Lemmatisation [10] :

Dans une phrase, un mot peut contenir un préfixe ou un suffixe en plus du mot de base ou racine. Le mot racine porte l'information la plus significative sur toute entité de mot présente dans une phrase. Par conséquent, identifier le mot racine est une tâche majeure dans le prétraitement des données textuelles. En NLP, la lemmatisation est le processus d'identification du mot racine d'un mot particulier dans une phrase. Ce processus est soutenu par le POS tagging (étiquetage des parties du discours), qui indique le type et le comportement d'un mot, aidant ainsi à identifier les suffixes et préfixes possibles qu'il peut contenir, en plus de la base. Les mots racines présents dans toutes les N phrases sont découverts, ce qui signifie que toutes les N phrases sont lemmatisées. Pour lemmatiser toutes les phrases, la fonction `WordNetLemmatizer()` du package NLTK a été utilisée.

1.2 Exploration de modèles pour l'extraction de mots-clés

L'extraction de mots-clés [11](également connue sous le nom de détection de mots-clés ou analyse de mots-clés) est une technique d'analyse de texte qui extrait automatiquement les mots et expressions les plus utilisés et les plus importants d'un texte. Elle aide à résumer le contenu des textes et à identifier les principaux sujets abordés.

L'extraction de mots-clés utilise l'intelligence artificielle (IA) et le traitement du langage naturel (NLP) pour décomposer le langage humain de manière à ce qu'il puisse être compris et analysé par les machines. Cette technique est utilisée pour identifier des mots-clés dans divers types de textes : documents réguliers et rapports d'entreprise, commentaires sur les réseaux sociaux, forums et avis en ligne, articles de presse, et plus encore.

Dans cette section, j'explore le domaine des modèles d'intelligence artificielle (IA) conçus pour l'extraction de texte, un processus essentiel en traitement du langage naturel (NLP) qui consiste à identifier et extraire des informations structurées à partir de textes non structurés. J'examinerai divers modèles d'IA de pointe développés à cet effet, discuterai de leurs méthodologies, de leurs points forts et de leurs applications. Mon attention se portera sur la compréhension de la façon dont ces modèles fonctionnent dans le contexte de l'extraction d'informations pertinentes à partir des descriptions de poste, contribuant ainsi à l'automatisation et à l'optimisation des processus RH et à la prise de décision.

1.2.1 TF-IDF (Fréquence de terme-Fréquence inverse de document)

$$w_{x,y} = \text{tf}_{x,y} \times \log \left(\frac{N}{df_x} \right)$$

TF-IDF
Term x within document y

$\text{tf}_{x,y}$ = frequency of x in y
 df_x = number of documents containing x
 N = total number of documents

FIGURE 35 – TF-IDF

La fréquence de terme-fréquence inverse de document (TF-IDF) [7] est une méthode statistique largement utilisée dans le traitement du langage naturel et la recherche d'informations. Elle mesure l'importance d'un terme au sein d'un document par rapport à une collection de documents.

Fréquence de terme :

La fréquence de terme (TF) d'un terme ou mot est le nombre de fois que ce terme apparaît dans un document par rapport au nombre total de mots dans ce document.

Fréquence inverse de document :

La fréquence inverse de document (IDF) d'un terme reflète la proportion de documents dans le corpus qui contiennent ce terme. Les mots uniques à un petit pourcentage de documents (par exemple, les termes de jargon technique) reçoivent des valeurs d'importance plus élevées que les mots communs à tous les documents (par exemple, a, the, and).

1.2.2 TextRank

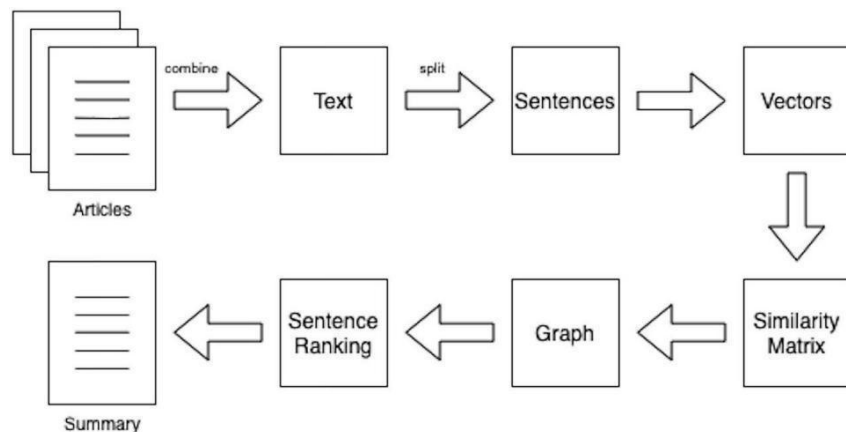


FIGURE 36 – TextRank

TextRank [4] est un modèle de classement basé sur les graphes pour le traitement du texte, qui peut être utilisé pour trouver les phrases les plus pertinentes dans un texte et également pour identifier les mots-clés. Pour trouver les mots-clés pertinents, l'algorithme TextRank construit un réseau de mots. Ce réseau est créé en examinant quels mots suivent les uns après les autres. Un lien est établi entre deux mots s'ils se suivent,

et ce lien obtient un poids plus élevé si ces deux mots apparaissent plus fréquemment l'un à côté de l'autre dans le texte.

Ensuite, l'algorithme PageRank est appliqué au réseau résultant pour déterminer l'importance de chaque mot. Le tiers supérieur de tous ces mots est conservé et considéré comme pertinent. Après cela, une table de mots-clés est construite en combinant les mots pertinents ensemble s'ils apparaissent l'un après l'autre dans le texte.

1.2.3 Bert

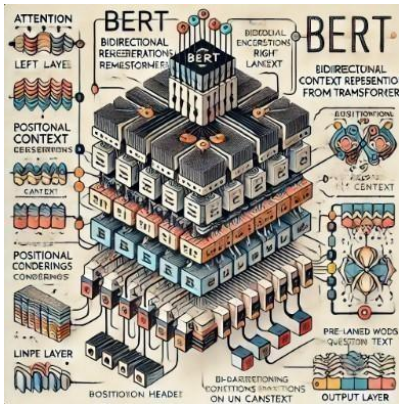


FIGURE 37 – Bert

BERT [12] est conçu pour pré-entraîner des représentations bidirectionnelles profondes à partir de texte non étiqueté en conditionnant conjointement sur le contexte gauche et droit dans toutes les couches. En conséquence, le modèle BERT pré-entraîné peut être affiné avec une seule couche de sortie supplémentaire pour créer des modèles à la pointe de la technologie pour une large gamme de tâches, telles que la réponse à des questions et l'inférence de langage, sans modifications substantielles de l'architecture spécifique aux tâches. BERT est conceptuellement simple et empiriquement puissant.

1.2.4 RAKE

Abréviation de Rapid Automatic Keyword Extraction algorithm [8], est un algorithme d'extraction de mots-clés indépendant du domaine qui tente de déterminer les phrases clés dans un texte en analysant la fréquence d'apparition des mots et leur co-occurrence avec d'autres mots dans le texte. Un des principes clés de RAKE est que les mots-clés se composent souvent de plusieurs mots mais incluent rarement des signes de ponctuation, des mots vides ou d'autres termes ayant une signification lexicale minimale. L'algorithme met l'accent sur la collocation et la co-occurrence des mots, ce qui est évident lors de l'analyse des données de retour d'expérience de sources telles que les sites de commerce électronique, où des expressions comme "Bonne Caméra" ou "Service Client" apparaissent fréquemment ensemble, démontrant une proximité sémantique. À l'inverse, des termes comme "Mauvaise" et "Pire" dans des expressions comme "Mauvaise Caméra" ou "Pire Caméra" partagent des significations similaires, soulignant davantage leur probabilité de co-occurrence, en particulier dans des contextes spécifiques tels que les téléphones mobiles ou les appareils photo reflex numériques.

Après avoir obtenu le corpus de texte, RAKE segmente le texte en une liste de mots, en filtrant les mots vides pour obtenir ce que l'on appelle la liste des mots de contenu. Les praticiens du traitement du langage naturel comprennent que les mots vides sont des termes comme "sont", "ne", "il y a", ou "est", qui apportent une signification minimale à une phrase et peuvent être ignorés en toute sécurité. En supprimant ces mots vides, le corpus principal est simplifié, le rendant plus concis et pertinent.

1.2.5 KeyBERT



FIGURE 38 – keybert

KeyBERT [2] est une technique d'extraction de mots-clés minimale et facile à utiliser qui tire parti des embeddings BERT pour créer des mots-clés et des expressions clés les plus similaires à un document.

1.2.6 YAKE

Yet Another Keyword Extractor [1] est une méthode d'extraction de mots-clés automatique non supervisée qui identifie les mots-clés les plus pertinents dans un texte en utilisant des données statistiques textuelles à partir de textes uniques. Cette technique ne repose pas sur des dictionnaires, des corpus externes, la taille du texte, la langue ou le domaine, et ne nécessite pas d'entraînement sur un ensemble spécifique de documents.

1.3 Application d'extraction de mots-clés

1.3.1 KeyBert Methodology

Il y a quatre étapes que KeyBERT accomplit [5] pour produire des prédictions [3]. À un niveau élevé, le package calcule un vecteur qui représente le sens de chaque mot/expression possible et un vecteur pour le sens global du document, puis détermine les mots/expressions les plus similaires au document global en utilisant les vecteurs calculés.

Étape 1 : Extraction de mots/phrases[5] :

La phase initiale consiste à identifier les termes et expressions potentiels en utilisant le CountVectorizer de scikit-learn. Cet outil segmente le texte en tokens et permet de spécifier une gamme pour les n-grammes à évaluer, en suivant le compte des occurrences de chaque n-gramme spécifié. À l'inverse, KeyBERT évalue les n-grammes uniquement en fonction de leur présence, sans tenir compte de leur fréquence dans le texte.

Étape 2 : Vecteurs [5] :

À cette étape, la bibliothèque Sentence Transformers est utilisée pour créer des vecteurs capturant l'essence des mots/phrases extraits ainsi que de l'ensemble du document. Contrairement à des outils comme Word2vec, qui génèrent des vecteurs pour des mots individuels, Sentence Transformers produit efficacement des vecteurs pour des textes complets. Cette méthode rapproche des phrases similaires dans l'espace vectoriel, facilitant la compréhension des sémantiques du document en comparant leurs représentations vectorielles. Pour une compréhension complète, référez-vous à l'article complet sur Sentence Transformers.

Étape 3 : Comparaison des vecteurs [5] :

À cette phase, KeyBERT évalue la similarité entre le vecteur global du document et les vecteurs des mots/phrases extraits en utilisant la similarité cosinus. Cette mesure aide à discerner la proximité des mots-clés avec le contexte du document. Les mots-clés avec des vecteurs les plus proches du vecteur du document sont jugés les plus pertinents et sont ainsi sélectionnés comme les meilleurs mots-clés.

Étape 4 : Diversification [5] :

À cette étape, des stratégies de diversification sont mises en œuvre pour garantir une plus grande variété de mots-clés, évitant la sélection de termes très similaires. Cette approche empêche le modèle de choisir des phrases répétitives, comme différentes variations de "pomme", lorsqu'un texte couvre divers fruits. L'objectif est que le modèle mette en avant une gamme diversifiée de sujets clés du document plutôt que de se concentrer sur un seul sujet.

1.3.2 YAKE methodology

L'algorithme YAKE! fonctionne selon les étapes principales suivantes :

Prétraitement du texte et identification des termes candidats :

Nettoie le texte et repère les candidats potentiels pour les mots-clés en transformant le document en un format que la machine peut analyser efficacement.

Extraction de caractéristiques [1] :

Analyse chaque terme en fonction de caractéristiques spécifiques telles que la fréquence, la position dans le texte et le contexte, afin de comprendre sa pertinence potentielle.

Calcul du score de chaque terme [1] :

Attribue un score de signification à chaque terme en utilisant les caractéristiques extraites, les scores les plus bas indiquant une importance plus élevée.

Génération de n-grammes et calcul du score des mots-clés candidats [1] :

Combine les mots individuels en n-grammes et calcule des scores pour ces phrases, identifiant ainsi les meilleurs candidats pour les mots-clés.

Déduplication et classement des données [1] :

Élimine les termes redondants et organise les mots-clés restants par leurs scores, mettant en évidence les termes les plus pertinents dans le texte.

1.3.3 TextRank Methodology

L'algorithme utilise l'algorithme [4] PageRank pour classer les termes les plus significatifs dans le texte. TextRank s'intègre parfaitement dans le pipeline de Spacy et exécute les étapes clés suivantes pour l'extraction de mots-clés :

Étape 1 [4] :

TextRank construit un réseau de mots (graphe de mots) pour identifier les termes pertinents en se basant sur la co-occurrence des mots. Les mots fréquemment trouvés ensemble dans le texte sont liés, avec des connexions plus fortes pour les paires à fréquence élevée.

Étape 2 [4] :

L'algorithme PageRank évalue la pertinence des mots dans le réseau. Le tiers supérieur des termes est conservé comme significatif. Lorsque des termes pertinents apparaissent consécutivement dans le texte, ils sont regroupés ensemble dans une table de mots-clés.

1.3.4 TF-IDF methodology

Calcul de la Fréquence de Terme (TF) [7] :

Comptez le nombre de fois que chaque mot apparaît dans un document et divisez par le nombre total de mots dans ce document.

Calcul de la Fréquence Inverse de Document (IDF) [7] :

Échelle logarithmique de la fraction inverse des documents contenant le mot, pour diminuer le poids des termes qui apparaissent très fréquemment dans les documents.

Calcul du TF-IDF [7] :

Multipliez TF par IDF pour chaque terme dans chaque document afin de déterminer son importance dans le document.

Extraction de mots-clés [7] :

Classez les mots en fonction de leurs scores TF-IDF pour identifier les termes les plus pertinents dans chaque document.

1.3.5 RAKE methodology

Analyse de texte [9] :

Divisez le texte en un tableau de mots et de phrases, en utilisant les mots vides et la ponctuation comme délimiteurs.

Identification des candidats [9] :

Identifiez les mots-clés potentiels, en considérant les phrases délimitées par des mots vides ou la ponctuation comme des mots-clés candidats.

Calcul des scores [9] :

Calculez les scores pour chaque candidat en tenant compte de la fréquence des mots et de leur degré (nombre total de candidats dans lesquels un mot apparaît).

Extraction de mots-clés [9] :

Classez les phrases en fonction de leurs scores et sélectionnez les phrases ayant les scores les plus élevés comme mots-clés extraits.

1.4 Méthodologie d'extraction et de classification des mots-clés

Dans la phase initiale de ce projet, j'ai adopté une approche multi-facettes pour l'identification et la classification des mots-clés à partir des descriptions de poste. Cette stratégie reposait principalement sur l'exploitation de modèles d'extraction de mots-clés non supervisés tels que **KeyBert**, **RAKE**, **YAKE** et **TextRank**. J'ai déployé ces modèles, développés from scratch, pour isoler ce que j'ai identifié comme des mots-clés potentiels dans le corpus des descriptions de poste.

Étape 1 : Extraction des mots-clés potentiels

J'ai appliqué les modèles KeyBert, RAKE, YAKE et TextRank au jeu de données pour extraire une liste initiale de mots-clés. Ces mots-clés potentiels ont servi de base pour la phase de classification suivante. Cette approche m'a donné une vue d'ensemble des thèmes et sujets prédominants dans les descriptions de poste, tout en maintenant la diversité des méthodes d'extraction.

Étape 2 : Classification et référencement des mots

Avec la liste de mots-clés obtenue, j'ai procédé à la classification de chaque terme présent dans les descriptions de poste. J'ai utilisé la technique TF-IDF pour évaluer l'importance de chaque mot dans le contexte spécifique des descriptions, en utilisant les mots-clés extraits comme références. Cette méthode a aidé à affiner la pertinence et le contexte des mots-clés potentiels.

Étape 3 : Entraînement et évaluation du modèle de régression logistique

Par la suite, j'ai entraîné un modèle de régression logistique en lui fournissant des données étiquetées, où chaque mot était marqué comme étant un mot-clé ("isKey") ou non. Ce modèle était conçu pour estimer la probabilité que chaque mot examiné soit un mot-clé valide, en se basant sur les critères appris lors de l'entraînement.

Étape 4 : Sélection des mots-clés basée sur la probabilité

J'ai fixé un seuil de probabilité pour déterminer quels mots devaient être considérés comme des mots-clés finaux. Seuls les termes avec une probabilité supérieure à ce seuil ont été retenus comme mots-clés à haute probabilité. Cette approche a permis d'isoler les termes les plus significatifs et pertinents par rapport au contenu général des descriptions de poste.

Étape 5 : Création et enregistrement des résultats

Enfin, j'ai consolidé les résultats en créant une nouvelle colonne dans le jeu de données, appelée "**Skills**", pour stocker les mots-clés sélectionnés. Cette colonne sert non seulement de résumé des termes les plus pertinents identifiés dans chaque description de poste, mais aussi de base pour des analyses ultérieures et des processus d'amélioration du modèle.

Cette méthodologie détaillée reflète l'approche systématique et itérative adoptée pour maximiser l'efficacité de l'extraction et de la classification des mots-clés dans ce projet. En identifiant et en affinant constamment les mots-clés les plus pertinents, je vise à améliorer la précision et la fiabilité de notre assistant d'interview basé sur l'IA.

1.5 Modèles entraînés

1.5.1 Régression logistique

La régression logistique [18] teste un modèle de régression dont la variable dépendante est dichotomique (codée 0-1) et dont les variables indépendantes peuvent être continues ou catégorielles. La régression logistique binomiale est très similaire à la régression linéaire. Le poids de chaque variable indépendante est représenté par un coefficient de régression, et il est possible de calculer la taille de l'effet du modèle avec un indice similaire au coefficient de détermination. Cependant, elle ne nécessite pas la présence d'une relation linéaire entre les variables, puisque la variable dépendante est dichotomique.

Un modèle de régression logistique peut également prédire la probabilité qu'un événement se produise (valeur de 1) ou non (valeur de 0) en fonction de l'optimisation des coefficients de régression. Ce résultat varie toujours entre 0 et 1. Lorsque la valeur prédite est supérieure à 0,5, l'événement est susceptible de se produire, tandis que lorsque cette valeur est inférieure à 0,5, il est peu probable qu'il se produise.

1.5.2 Forêt aléatoire

La forêt aléatoire [16] est un algorithme de machine learning couramment utilisé, inventé par Leo Breiman et Adele Cutler, qui combine les résultats de plusieurs arbres de décision pour parvenir à un seul résultat. Sa facilité d'utilisation et sa flexibilité ont favorisé son adoption, car il gère à la fois les problèmes de classification et de régression.

L'algorithme de la forêt aléatoire est une extension de la méthode de bagging, car il utilise à la fois le bagging et la randomisation des caractéristiques pour créer une forêt non corrélée d'arbres de décision. La randomisation des caractéristiques, également connue sous le nom de feature bagging ou "méthode des sous-espaces aléatoires", génère un sous-ensemble aléatoire de caractéristiques, ce qui assure une faible corrélation entre les arbres de décision. C'est une différence clé entre les arbres de décision et les forêts aléatoires. Alors que les arbres de décision considèrent toutes les divisions possibles des caractéristiques, les forêts aléatoires ne sélectionnent qu'un sous-ensemble de ces caractéristiques.

1.6 Méthodes d'optimisation

L'optimisation [17] est le processus par lequel nous entraînons le modèle de manière itérative, ce qui aboutit à une évaluation maximale et minimale de la fonction. C'est l'un des phénomènes les plus importants en apprentissage automatique pour obtenir de meilleurs résultats.

Pourquoi optimisons-nous nos modèles d'apprentissage automatique ? Nous comparons les résultats à chaque itération en changeant les hyperparamètres à chaque étape jusqu'à atteindre les résultats optimaux. Nous créons un modèle précis avec un taux d'erreur réduit. Il existe différentes manières d'optimiser un modèle.

1.6.1 GridSearch

GridSearchCV [15] est le processus de réglage des hyperparamètres afin de déterminer les valeurs optimales pour un modèle donné. Comme mentionné précédemment, les performances d'un modèle dépendent considérablement de la valeur des hyperparamètres. Il n'y a pas de moyen de connaître à l'avance les meilleures valeurs pour les hyperparamètres, donc idéalement, nous devons essayer toutes les valeurs possibles pour connaître les valeurs optimales. Faire cela manuellement pourrait prendre beaucoup de temps et de ressources, c'est pourquoi nous utilisons GridSearchCV pour automatiser le réglage des hyperparamètres. GridSearchCV est une fonction qui se trouve dans le package 'model_selection' de Scikit-learn (ou SK-learn). Il est donc important de noter que nous devons avoir la bibliothèque Scikit-learn installée sur l'ordinateur. Cette fonction permet de parcourir les hyperparamètres prédéfinis et d'ajuster votre estimateur (modèle) sur votre ensemble d'entraînement. Ainsi, à la fin, nous pouvons sélectionner les meilleurs paramètres parmi les hyperparamètres listés.

Comme mentionné précédemment, nous passons des valeurs prédéfinies pour les hyperparamètres à la fonction GridSearchCV. Nous le faisons en définissant un dictionnaire dans lequel nous mentionnons un hyperparamètre particulier avec les valeurs qu'il peut prendre. GridSearchCV essaie toutes les combinaisons des valeurs passées dans le dictionnaire et évalue le modèle pour chaque combinaison en utilisant la méthode de validation croisée. Ainsi, après avoir utilisé cette fonction, nous obtenons la précision/perte pour chaque combinaison d'hyperparamètres et nous pouvons choisir celle avec les meilleures performances.

1.6.2 RandomSearch

Alors que la recherche par grille (grid search)[14] examine toutes les combinaisons possibles d'hyperparamètres pour trouver le meilleur modèle, la recherche aléatoire (random search) sélectionne et teste uniquement une combinaison aléatoire d'hyperparamètres.

Cette technique échantillonne aléatoirement à partir d'une grille d'hyperparamètres au lieu de mener une recherche exhaustive.

Nous pouvons spécifier le nombre total d'exécutions que la recherche aléatoire doit essayer avant de retourner le meilleur modèle.

1.7 Les modèles de génération des Questions

Les systèmes de génération automatique de questions (AQG) [19] sont ceux dans lesquels les questions sont générées en fonction d'un sujet, d'une idée ou d'un contexte en langage naturel à partir d'un paragraphe de texte ou d'images. Ces systèmes deviennent de plus en plus populaires, en particulier dans les applications de compréhension de lecture par machine, les systèmes conversationnels et même les applications éducatives.

L'objectif final des systèmes AQG est la capacité de générer des questions qui sont correctes syntaxiquement et sémantiquement ainsi que significatives dans le contexte de leur utilisation.

1.7.1 Modèle T5

Ce modèle [20] est un générateur de questions de type séquence-à-séquence basé sur le modèle pré-entraîné T5-base. Conçu spécifiquement pour créer des questions de compréhension de lecture, il gère efficacement les réponses complètes ainsi que les mots ou phrases courts. En utilisant le contexte pour former une séquence d'entrée, et après encodage, cette séquence est utilisée par la méthode generate() pour produire une question pertinente. Cet outil est idéal pour générer du matériel pédagogique, automatiser les FAQ et créer des bases de données pour entraîner des systèmes avancés de compréhension de texte. Son intégration dans les applications

de traitement du langage naturel facilite la génération automatique de questions et améliore la compréhension textuelle des algorithmes, contribuant ainsi aux avancées de l'intelligence artificielle conversationnelle et des systèmes de support décisionnel basés sur le texte.

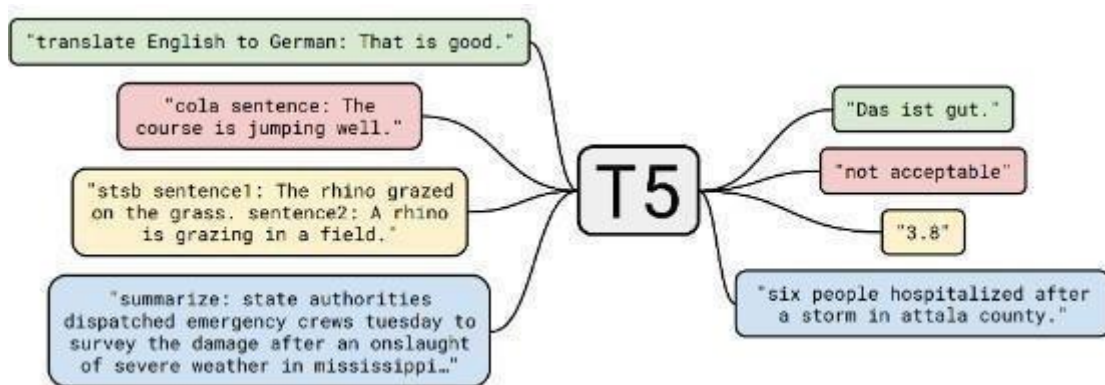


Figure 39 – Modèle T5[20]

1.7.2 Modèle BART

Ce modèle [21] est un générateur de questions basé sur BART-base et entraîné sur le corpus EQG-RACE. Il prend un contexte comme entrée et génère une question en sortie, avec une longueur de séquence maximale de 1024 tokens. Les séquences d'entrée encodées sont utilisées par la méthode `generate()` pour créer des questions pertinentes. Ce modèle est idéal pour automatiser la création de questions éducatives et améliorer les systèmes de compréhension de texte.

1.7.3 Modèle `lmqg/t5-large-squad-qg`

Notre modèle de génération automatique de questions (AQG) [22] se divise en trois catégories : Pipeline, Multitâche et End2end. Le modèle Pipeline génère des questions indépendamment des réponses fournies dans le contexte. Le modèle Multitâche, similaire au modèle Pipeline, partage et affine conjointement les modules pour la génération de questions. L'approche End2end, quant à elle, génère des questions de manière intégrée, sans se concentrer séparément sur les réponses.

Les modèles Pipeline et Multitâche ont tendance à générer plus de questions, tandis que le modèle End2end offre une génération de questions plus rapide. Tous les types sont disponibles en 8 langues différentes via `lmqg` et sont partagés sur HuggingFace.

1.7.4 The Stanford Question Answering Dataset

Stanford Question Answering Dataset (SQuAD) [40] is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable.

1.8 Les modèles d'évaluation des réponses

Dans ce projet, j'ai utilisé un ensemble de techniques de traitement du langage naturel (NLP) et d'apprentissage automatique pour évaluer la pertinence des réponses générées par un modèle de génération automatique de questions.

Tout d'abord, j'ai installé le package `'datasets'` et chargé le dataset SQuAD (Stanford Question Answering Dataset). Ensuite, j'ai créé une fonction `'generate_incorrect_response'` qui génère des réponses incorrectes en mélangeant les mots du contexte de la question. Cette fonction m'a permis de créer des exemples de réponses incorrectes pour l'entraînement du modèle.

Pour chaque entrée du dataset SQuAD, j'ai extrait le contexte, la question et la réponse correcte. J'ai ensuite généré une réponse incorrecte en utilisant la fonction précédemment définie. J'ai structuré ces données

sous forme de liste de dictionnaires, où chaque dictionnaire contient une question, une réponse (correcte ou incorrecte) et un label indiquant si la réponse est correcte (1) ou incorrecte (0).

Ensuite, j'ai converti cette liste en un DataFrame pandas et sauvegardé le DataFrame sous forme de fichier CSV pour une utilisation ultérieure.

Afin de préparer les données pour l'entraînement, j'ai effectué un prétraitement des textes en utilisant les bibliothèques **'nlTK'** et **'spacy'**. J'ai nettoyé les textes en supprimant les caractères non alphabétiques et les espaces supplémentaires, en convertissant les textes en minuscules et en lemmatisant les mots tout en supprimant les mots vides (stop words). J'ai combiné les questions et les réponses prétraitées en une seule colonne de texte pour la vectorisation.

Pour la vectorisation, j'ai utilisé le **'TfidfVectorizer'** de **'scikit-learn'** pour convertir les textes en vecteurs TF-IDF. J'ai ensuite entraîné un modèle de régression logistique en utilisant ces vecteurs comme caractéristiques et les labels comme cible. Après l'entraînement, j'ai sauvegardé le modèle et le vectoriseur dans des fichiers pour une utilisation ultérieure.

Enfin, j'ai chargé le modèle et le vectoriseur sauvegardés et créé une fonction **'evalue_reponse'** pour évaluer de nouvelles réponses. Cette fonction prétraite la question et la réponse, les combine, les transforme en vecteur TF-IDF et utilise le modèle de régression logistique pour prédire si la réponse est correcte ou incorrecte. J'ai testé cette fonction avec un exemple de question et de réponse, et affiché la prédiction obtenue.

Cette approche m'a permis de créer un système automatisé pour évaluer la pertinence des réponses générées, en utilisant des techniques avancées de NLP et d'apprentissage automatique.

1.9 Chatbot Frameworks

1.9.1 Rasa



FIGURE 40 – rasa framework

Rasa [23] se distingue par sa plateforme open-source dédiée à l'intelligence artificielle conversationnelle, soutenue par une communauté de développeurs en pleine expansion, avec plus de 3 millions de téléchargements. Ce qui rend Rasa particulièrement unique, c'est sa capacité à apprendre des dialogues grâce à l'apprentissage automatique à partir d'exemples concrets. De plus, son architecture flexible pour la compréhension du langage naturel (NLU) lui permet de s'adapter à divers besoins.

L'une des caractéristiques notables de Rasa est sa flexibilité en matière de déploiement, offrant des options à la fois sur site (on-premise) et dans le cloud. Cela en fait l'une des rares plateformes capables de former des modèles de dialogue via l'apprentissage automatique tout en permettant une personnalisation poussée de son pipeline de compréhension du langage naturel. Cette personnalisation inclut l'intégration de différents composants pour le traitement du langage, permettant aux utilisateurs d'ajuster précisément le comportement du système en fonction de leurs besoins spécifiques.

En résumé, Rasa combine la puissance de l'apprentissage automatique pour l'apprentissage des dialogues avec une architecture modulable et une flexibilité de déploiement, répondant ainsi à une grande variété d'applications en matière d'IA conversationnelle.

1.9.2 DialogFlow



FIGURE 41 – DialogFlow

Dialogflow [24] est une plateforme issue initialement de API.AI, conçue pour permettre une communication naturelle et orientée dialogue avec les utilisateurs. Aujourd’hui, Dialogflow est un service de traitement du langage naturel (NLP) qui fait partie intégrante de la Google Cloud Platform. Cette plateforme offre à ses utilisateurs une suite de développement complète, comprenant un éditeur de code, une bibliothèque et de nombreux outils.

Grâce à cette suite intégrée, la création d’interfaces conversationnelles devient beaucoup plus accessible. Les développeurs peuvent ainsi concevoir des agents conversationnels sophistiqués qui interprètent les intentions des utilisateurs, gèrent les dialogues et répondent de manière appropriée. Dialogflow simplifie le processus en fournissant des outils pour tester et déployer les agents, tout en permettant une intégration fluide avec d’autres services Google Cloud.

En somme, Dialogflow facilite la création et le déploiement d’interfaces conversationnelles grâce à sa riche palette d’outils, tout en profitant de la robustesse et de l’évolutivité de la Google Cloud Platform. Cela en fait une solution idéale pour les développeurs cherchant à concevoir des expériences utilisateurs interactives et naturelles.

1.9.3 Microsoft Bot Framework

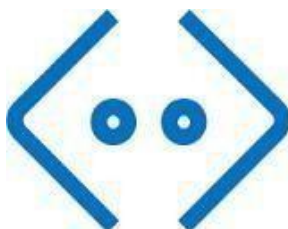


FIGURE 42 – Microsoft Bot Framework

Le Microsoft Bot Framework [25] est un ensemble d’outils et de services conçus pour aider les développeurs à créer et connecter des bots intelligents. Ce framework permet aux développeurs de concevoir des bots

capables d'interagir avec les utilisateurs de manière naturelle, en utilisant des dialogues conversationnels. Il offre également une série d'outils et de services qui facilitent la connexion de votre bot aux canaux populaires tels que Facebook, Skype et Twitter.

Avec le Microsoft Bot Framework, les développeurs bénéficient d'une plateforme robuste pour construire des bots capables de comprendre et de répondre aux utilisateurs de façon fluide et intuitive. En outre, la plateforme permet une intégration aisée avec diverses applications et services, rendant la diffusion des bots sur plusieurs canaux simple et efficace. Les outils fournis incluent des fonctionnalités pour tester, déployer et gérer les bots, tout en assurant une compatibilité avec les dernières technologies en matière de traitement du langage naturel.

En résumé, le Microsoft Bot Framework se distingue par sa capacité à simplifier le développement de bots intelligents tout en offrant une flexibilité pour les connecter à un large éventail de canaux de communication, facilitant ainsi la création de solutions conversationnelles interactives et adaptatives.

1.9.4 Botpress



FIGURE 43 – Botpress

Botpress [32] est un framework que nous avons conçu en pensant spécifiquement aux développeurs de chatbots professionnels, y compris les équipes de contenu. Nos clients cibles sont les entreprises, les agences et les start-ups qui ont besoin de créer des bots de qualité avec des ressources limitées.

Botpress se distingue par sa capacité à fournir un environnement complet pour le développement de chatbots, tout en étant accessible aux équipes ayant des ressources techniques limitées. Le framework permet aux utilisateurs de concevoir des bots sophistiqués sans nécessiter une expertise approfondie en programmation. Il offre des outils intuitifs pour la création de flux conversationnels, l'intégration de contenu, et la gestion des interactions utilisateur.

L'objectif de Botpress est de permettre à ses utilisateurs de déployer rapidement et efficacement des solutions de chatbot qui répondent aux besoins de leurs entreprises, tout en minimisant les coûts de développement et en maximisant l'impact des bots créés. En somme, Botpress offre une solution pratique et puissante pour les organisations cherchant à exploiter les capacités des chatbots avec des moyens réduits, tout en garantissant une haute qualité dans les résultats obtenus.

1.9.5 IBM Watson



FIGURE 44 – IBM Watson

watsonx Assistant [27] est une plateforme conviviale qui fournit aux utilisateurs non techniques, notamment ceux des lignes de métier, tous les outils nécessaires pour créer des chatbots personnalisés basés sur l'IA, et cela sans avoir à écrire de code.

Cette solution développée par IBM Watson est spécialement conçue pour permettre à des utilisateurs ayant peu ou pas de compétences techniques de concevoir et déployer des chatbots intelligents. Grâce à une interface intuitive, watsonx Assistant offre la possibilité de configurer des conversations, d'intégrer des réponses adaptées et de personnaliser l'expérience utilisateur sans nécessiter l'intervention d'un développeur.

Les utilisateurs peuvent ainsi créer des solutions sur mesure, alignées sur les besoins spécifiques de leur entreprise, tout en tirant parti des capacités avancées de l'intelligence artificielle.

En résumé, watsonx Assistant d'IBM Watson démocratise la création de chatbots en la rendant accessible à un public plus large, notamment les professionnels de métier, tout en offrant une personnalisation poussée et une simplicité d'utilisation qui permet de maximiser l'efficacité des interactions clients.

1.9.6 Wit.ai



FIGURE 45 – Wit.ai

Wit.ai [28] est un framework open-source pour la création de chatbots, développé par Facebook. Il permet aux utilisateurs de contrôler des enceintes intelligentes, des appareils électroménagers, des systèmes d'éclairage, et bien plus encore, simplement en utilisant leur voix.

Wit.ai se distingue par sa capacité à transformer les commandes vocales en actions concrètes, facilitant ainsi l'interaction avec divers dispositifs intelligents. Grâce à cette technologie, les développeurs peuvent intégrer des fonctionnalités de reconnaissance vocale dans leurs applications, rendant ainsi les interactions plus naturelles et intuitives. Le framework est conçu pour être flexible et accessible, permettant à une large communauté de développeurs de créer des solutions vocales personnalisées pour une variété d'applications.

En résumé, Wit.ai offre une plateforme puissante pour le développement de chatbots et d'assistants vocaux, facilitant l'intégration de commandes vocales dans une multitude de dispositifs connectés, tout en étant soutenu par l'expertise de Facebook dans le domaine de l'intelligence artificielle.

1.9.7 ChatterBot



FIGURE 46 – chatterBot

ChatterBot [29] est une bibliothèque Python qui simplifie la génération de réponses automatisées aux entrées des utilisateurs. ChatterBot utilise une sélection d'algorithmes d'apprentissage automatique pour produire différents types de réponses, ce qui facilite grandement la création de chatbots et l'automatisation des conversations avec les utilisateurs.

Grâce à ChatterBot, les développeurs peuvent rapidement mettre en place des bots conversationnels capables de s'adapter et d'apprendre à partir des interactions. La bibliothèque est conçue pour être flexible et extensible, permettant aux développeurs de personnaliser les réponses du bot en fonction des besoins spécifiques de leur application. Les algorithmes intégrés permettent au bot d'améliorer ses réponses au fil du temps, rendant les interactions plus pertinentes et naturelles.

En résumé, ChatterBot est un outil puissant pour les développeurs souhaitant créer des chatbots avec des capacités d'apprentissage automatique, tout en offrant une solution simple et efficace pour automatiser les échanges avec les utilisateurs.

1.9.8 PandoraBot



FIGURE 47 – PandoraBot

Pandorabots [30] est un framework open-source de développement de chatbots. Il est basé sur le langage de script AIML (Artificial Intelligence Markup Language), que les développeurs peuvent utiliser pour créer des bots conversationnels. Pandorabots est conçu principalement pour les développeurs et les concepteurs d'expérience client (CX Designers).

Contrairement à d'autres plateformes, Pandorabots ne propose pas d'outils d'apprentissage automatique préconfigurés. Cela signifie que les développeurs ont un contrôle total sur la manière dont le bot répond aux utilisateurs, en se basant uniquement sur des règles définies par l'AIML. Ce framework est idéal pour ceux qui cherchent à créer des bots hautement personnalisés sans dépendre de modèles d'apprentissage automatique. Il offre une grande flexibilité pour concevoir des dialogues structurés et des interactions spécifiques selon les besoins de l'application.

En résumé, Pandorabots est un outil puissant pour les développeurs et les designers CX qui souhaitent créer des chatbots conversationnels en utilisant AIML, tout en ayant un contrôle complet sur le comportement du bot sans recourir à l'apprentissage automatique.

1.9.9 Comparaison des frameworks du chatbot

Framework	Open Source	Deployment	Machine Learning	Community Support
Rasa	Yes	Both	Yes	High
Dialogflow	No	Cloud	Yes	High
Microsoft Bot Framework	Yes	Both	Yes	High
Botpress	Yes	Both	Yes	Medium
Amazon Lex	No	Cloud	Yes	High
IBM Watson	No	Cloud	Yes	High
Wit.ai	Yes	Cloud	Yes	Medium
ChatterBot	Yes	On-premise	No	Low
Pandorabots	Yes	Cloud	No	Medium

TABLE 4.1 – Comparaison des frameworks de chatbot

Tout d'abord, j'ai commencé par explorer le framework Pandorabots pour la construction de mon chatbot. Cependant, j'ai rapidement rencontré des limitations à plusieurs niveaux. Par exemple, il était difficile de programmer des rappels pour afficher la question suivante si le candidat ne répondait pas dans un délai de 2 minutes. De plus, il n'était pas possible de spécifier le nombre de questions à afficher pour chaque offre d'emploi. D'autres aspects m'ont également posé problème, notamment le fait que Pandorabots n'utilise pas un langage de programmation performant pour répondre à mes besoins spécifiques.

Face à ces contraintes, je me suis tourné vers l'utilisation de Rasa. Bien que j'aie rencontré des difficultés initiales, notamment dans la configuration des différents fichiers et l'utilisation du framework, j'ai pu m'appuyer sur la documentation pour surmonter ces obstacles. Au final, Rasa s'est révélé répondre à tous les critères que je devais établir pour mon chatbot, ce qui m'a permis de poursuivre efficacement le développement de mon projet.

Chapitre 5

Évaluation comparative des modèles : étude comparative (benchmarking)

1.1 Modèles d'extraction des mots-clés

MODELS	ACCURACY	TEMPS D'EXECUTION	TRESHOLD	CLASSIFICATION
RF+RAKE	89%	13.43S	-	CLASS 0: 16720 CLASS 1: 4622
RF+YAKE	98%	24.61S	-	CLASS 0: 21342 CLASS 1: 9087
RF+KEYBERT	99%	18.35S	-	CLASS 0: 21342 CLASS 1: 4926
RF+TEXTRANK	86%	15.63S	-	CLASS 0: 21342 CLASS 1: 5000
LR+RAKE	87%	0.31S	0.5	CLASS 0: 16720 CLASS 1: 4622
LR+YAKE	95%	0.77S	0.2	CLASS 0: 21342 CLASS 1: 9087
LR+KEYBERT	97%	0.40S	0.1	CLASS 0: 21342 CLASS 1: 4926
LR+TEXTRANK	94%	0.40S	0.1	CLASS 0: 21342 CLASS 1: 5000

FIGURE 48 – comparaison des performances des modèles d'extraction des mots clés

Le tableau présente une comparaison détaillée de plusieurs combinaisons de modèles, mettant en évidence leurs performances en termes de précision (accuracy), de temps d'exécution, et de classification. Parmi les modèles évalués, RF+KEYBERT se distingue par sa précision élevée de 99% et un temps d'exécution relativement court de 18,355 secondes, ce qui en fait un candidat robuste pour des applications où la précision est primordiale. Cependant, certains modèles, tels que RF+YAKE, bien que très précis (98%), nécessitent

un temps de traitement plus long, indiquant un compromis potentiel entre précision et efficacité temporelle.

L'intégration d'un seuil (threshold) dans les modèles basés sur la régression logistique (LR+RAKE, LR+YAKE, etc.) permet une meilleure gestion des classifications, en particulier lorsque la distribution des classes est déséquilibrée. Par exemple, le modèle LR+RAKE présente une répartition équilibrée entre les classes, ce qui pourrait être bénéfique pour des applications où la juste représentation des classes est critique.

En somme, ce tableau offre une vue d'ensemble des forces et des faiblesses de chaque combinaison de modèles, aidant à orienter le choix en fonction des priorités spécifiques du projet, qu'il s'agisse de maximiser la précision ou de minimiser le temps d'exécution.

1.2 Modèles de génération de questions

MODÈLES COMBINÉ	TEMPS D'EXECUTION	NB DE QUESTION GÉNÉRER	DIVERSITÉ DES QUESTIONS	COMPRÉHENSION CONTEXTUELLE
T5+RAKE+LR	25MIN	200	OUI	NON(LE MODÈLE UTILISE JUSTE LES MOTS CLÉS)
T5+KEYBERT+RF	35 MIN	200	OUI	NON(LE MODÈLE UTILISE JUSTE LES MOTS CLÉS)
BART+RAKE+LR	16.05 MIN	200	OUI	NON(LE MODÈLE UTILISE JUSTE LES MOTS CLÉS)
BART+KEYBERT+RF	32.35 MIN	200	OUI	NON(LE MODÈLE UTILISE JUSTE LES MOTS CLÉS)
T5 LARGE FOR SQUAD QG+RAKE+LR	40MIN	200	OUI	OUI (LE MODÈLE SE BASE SUR LE CONTEXTE)
T5 LARGE FOR SQUAD QG+KEYBERT+RF	16 MIN	200	OUI	OUI (LE MODÈLE SE BASE SUR LE CONTEXTE)

FIGURE 49– Comparaison des performances des modèles de génération des questions

Dans le cadre de mon évaluation comparative des modèles de génération de questions, j'ai examiné plusieurs combinaisons de techniques et de modèles, y compris l'association de méthodes d'extraction de mots-clés avec des modèles de traitement du langage naturel. Les critères évalués comprenaient le temps d'exécution, le nombre de questions générées, la diversité des questions et la compréhension contextuelle. Parmi les combinaisons testées, T5-RAKE-LR, T5-KeyBERT-RF, BART-RAKE-LR, BART-KeyBERT-RF, ainsi que deux configurations utilisant T5 Large for Squad avec différentes approches de prétraitement des mots-clés, ont toutes généré 200 questions. Cependant, leurs performances variaient en termes de temps d'exécution et de qualité des questions en fonction de la compréhension contextuelle.

Après une analyse minutieuse, j'ai choisi d'adopter la combinaison T5 Large for Squad QG+KeyBERT+RF pour sa capacité à générer des questions qui étaient non seulement pertinentes mais aussi diverses, reflétant une solide compréhension contextuelle. Cette configuration s'est distinguée par son efficacité et sa rapidité, avec un temps d'exécution de 16 minutes, tout en maintenant une haute qualité dans la génération de questions. Elle s'est avérée être le meilleur équilibre entre performance, pertinence et diversité des questions générées, idéale pour répondre aux besoins de mon projet de génération de questions. Ce tableau compare les performances de divers modèles combinés en termes de temps d'exécution, de nombre de questions générées,

de diversité des questions, et de compréhension contextuelle. Il est notable que le modèle BART+RAKE+LR se distingue par son efficacité en termes de temps, nécessitant seulement 16,05 minutes pour générer 200 questions. Cependant, cette rapidité est atteinte au prix d'une compréhension limitée du contexte, car le modèle repose uniquement sur les mots-clés.

D'un autre côté, les modèles comme T5 Large for SQuAD QG+RAKE+LR, bien que plus lents (40 minutes), intègrent une compréhension contextuelle approfondie, ce qui permet de générer des questions non seulement diversifiées mais également pertinentes en fonction du contexte. Cela est particulièrement utile dans des scénarios où la pertinence des questions, par rapport aux réponses précédentes, est essentielle, comme dans les entretiens automatisés.

Le tableau souligne ainsi l'importance de choisir un modèle en fonction des besoins spécifiques de l'application, qu'il s'agisse d'une exécution rapide ou d'une interprétation contextuelle plus fine .

1.3 Model d'évaluation des réponses des utilisateurs

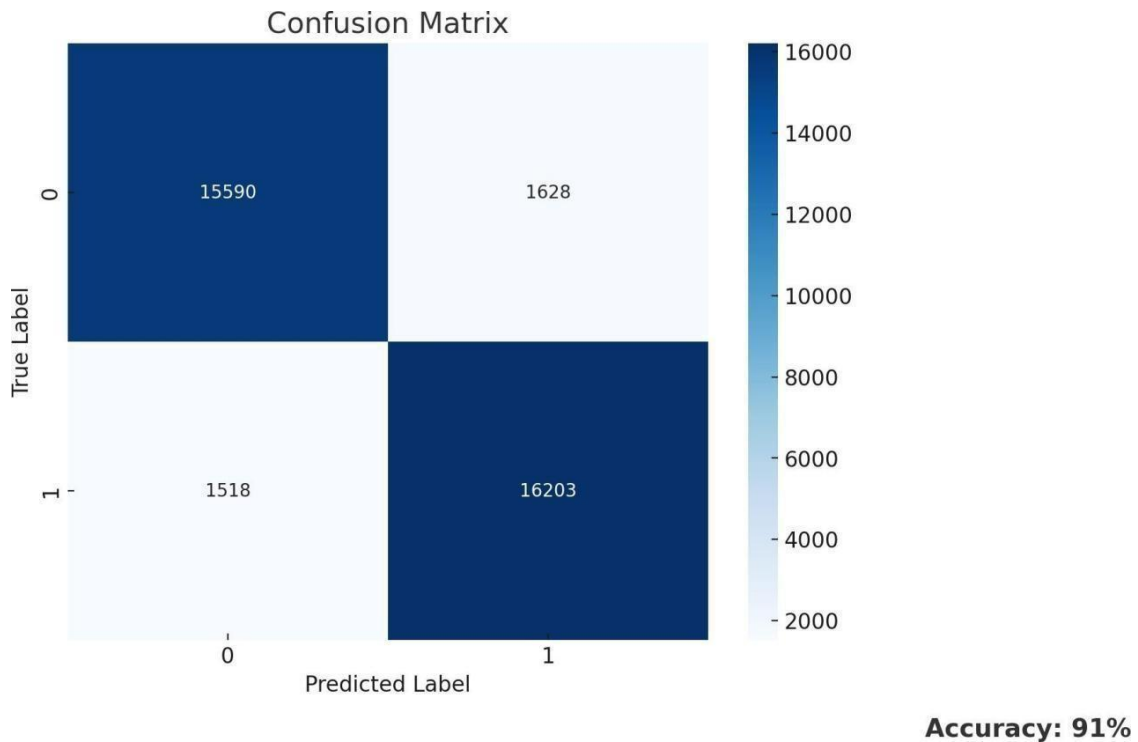


FIGURE 50 – Performance du model de l'évaluation des réponses utilisateurs

Ce modèle est conçu pour évaluer les réponses des utilisateurs dans le cadre d'une candidature à une offre d'emploi. Le processus d'évaluation commence par la présentation de plusieurs questions à l'utilisateur. Chaque réponse donnée par l'utilisateur est ensuite analysée et classée comme correcte ou incorrecte. Si la réponse est correcte, elle est étiquetée avec un score de 1. Si la réponse est incorrecte, elle reçoit un score de 0.

Une fois que toutes les réponses ont été collectées, on additionne les scores attribués à chaque réponse pour obtenir un total. Ce total est ensuite divisé par le nombre de questions qui ont été visualisées par l'utilisateur, ce qui donne un score moyen. Ce score moyen représente la performance globale de l'utilisateur sur l'ensemble des questions.

Le score moyen est ensuite transformé en une densité de probabilité, ce qui permet de normaliser les résultats et de les interpréter sur une échelle plus significative. Cette densité de probabilité reflète le score global de la candidature de l'utilisateur pour l'offre d'emploi en question.

Enfin, tous les candidats sont classés en fonction de leur score global. Cette classification permet d'identifier les candidats les plus qualifiés pour l'offre d'emploi, en se basant sur leur performance lors de l'évaluation. Ce processus garantit une sélection objective et basée sur les compétences démontrées par chaque candidat.

Chapitre 6

Challenges rencontrés

Au cours du développement du projet, plusieurs défis techniques majeurs ont été identifiés et surmontés pour assurer le bon fonctionnement du chatbot et la fluidité des interactions utilisateur. Voici les principaux challenges rencontrés :

1.1 Challenges techniques

Problème d'Extraction des Informations Envoyées du Backend au Serveur Rasa

L'un des défis les plus complexes a été la gestion de la transmission des informations critiques telles que le `user_id` et le `job_id` entre le backend (Django) et le serveur Rasa. Ce problème s'est manifesté par des difficultés dans l'envoi et la récupération de ces identifiants, ce qui a perturbé la logique d'application. Des ajustements ont été nécessaires au niveau de la vue Django ainsi que dans la logique d'envoi des données, pour garantir que les informations soient correctement extraites et traitées par le serveur Rasa.

Séparation des Sessions Connectées en Parallèle

La gestion des sessions utilisateur lorsque plusieurs utilisateurs étaient connectés en parallèle a posé un défi important. Il s'agissait de s'assurer que chaque session soit isolée, pour éviter toute confusion ou mélange d'informations entre utilisateurs différents. Ce problème a nécessité une révision approfondie de la manière dont les sessions étaient gérées et la mise en œuvre de mécanismes pour assurer que chaque utilisateur interagisse avec le bot dans un environnement totalement séparé.

Activation des Reminders et Affichage des Prochaines Questions

L'intégration et l'activation des reminders ont également représenté un défi crucial. L'objectif était de faire en sorte que le chatbot affiche automatiquement la question suivante si l'utilisateur ne répondait pas dans un délai de deux minutes. La mise en place de cette fonctionnalité a nécessité une compréhension approfondie des mécanismes de temporisation dans Rasa, ainsi qu'une manipulation précise des délais et des logiques d'affichage des questions. Ce challenge a été résolu en programmant des rappels (reminders) et en s'assurant que le flux de conversation reste fluide et engageant.

Problèmes dans la Phase de Prédiction et de Gestion des Données

Malgré une précision élevée du modèle (96,97%), la phase de prédiction a révélé des incohérences significatives, notamment dues à un étiquetage initial incorrect et à un déséquilibre des classes dans le jeu de données. Ce défi a exigé des ajustements dans le traitement des données et des améliorations dans l'équilibrage des classes pour permettre au modèle de mieux généraliser lors des prédictions sur de nouveaux cas.

Automatisation de l'Édition des Scénarios (Stories)

L'automatisation de l'ajout des titres de poste et des questions correspondantes dans les scénarios du chatbot s'est avérée complexe. Des erreurs de syntaxe et la nécessité de structurer correctement les interactions entre l'utilisateur et le bot ont compliqué ce processus. Des ajustements ont été nécessaires pour que les scénarios se déroulent sans interruption et que les interactions avec le chatbot restent cohérentes.

Problèmes de Performance et Optimisation des Modèles

L'optimisation des hyperparamètres des modèles, notamment via des méthodes de recherche aléatoire (*random search*) et de recherche en grille (*grid search*), a posé des défis, en particulier en termes de temps de calcul et de complexité. La gestion des tenseurs parcimonieux (*sparse tensors*) lors de l'entraînement du modèle a également nécessité des ajustements pour améliorer les performances globales.

Difficultés rencontrées dans l'évaluation des réponses des utilisateurs

Au cours du développement du système d'évaluation des réponses des candidats, j'ai rencontré plusieurs défis. Initialement, l'approche que j'avais choisie consistait à générer des réponses idéales (*idealAnswers*) pour chaque question posée par le chatbot. L'idée était ensuite de comparer la liaison sémantique entre ces réponses générées et les réponses fournies par les candidats.

Cependant, cette méthode s'est avérée problématique pour plusieurs raisons :

- **Algorithme de correspondance sémantique** : L'algorithme utilisé pour évaluer la similitude sémantique n'a pas toujours fonctionné comme prévu. Dans certains cas, il attribuait un bon score à des réponses incorrectes simplement parce qu'elles étaient formulées de manière similaire à la réponse idéale ou suivaient la même structure que la question. Cela a conduit à des évaluations biaisées et peu fiables.
- **Complexité du langage naturel** : Le langage naturel est complexe et varie énormément d'une personne à l'autre. Une simple correspondance basée sur la structure ou la formulation n'est pas suffisante pour évaluer correctement l'exactitude ou la pertinence d'une réponse. Par exemple, une réponse pourrait utiliser des synonymes ou une reformulation qui, bien que correcte, pourrait ne pas être reconnue comme telle par l'algorithme.

Chapitre 7

Conclusion & perspectives

1.1 Perspectives d'amélioration technologique du chatbot

Pour l'avenir, plusieurs axes de développement pourraient enrichir et améliorer encore davantage le chatbot :

- **Support Multilingue** : Afin d'élargir l'accessibilité du chatbot à un public international, il serait pertinent de développer un support multilingue. Cela impliquerait l'intégration de modèles de traitement du langage naturel (NLP) capables de gérer plusieurs langues, permettant ainsi au chatbot de comprendre et de répondre dans la langue choisie par le candidat.
- **Génération Contextuelle des Questions** : Une évolution intéressante serait de rendre le chatbot capable de générer des questions en fonction des informations spécifiques au poste à pourvoir et aux données fournies par le candidat lors de son enregistrement (par exemple, via son CV ou une description personnelle). Cette personnalisation pourrait être réalisée en utilisant des modèles d'apprentissage supervisé ou non supervisé, qui ajusteraient les questions en fonction des compétences recherchées et du profil du candidat.
- **Intégration avec les Plateformes RH** : Afin d'optimiser l'efficacité du processus de recrutement, l'intégration du chatbot avec des systèmes de gestion des ressources humaines (SIRH) existants, tels que Workday ou SAP SuccessFactors, serait un atout majeur. Cette intégration permettrait une synchronisation fluide des données, une gestion centralisée des candidatures, et une analyse plus approfondie des performances des candidats.
- **Personnalisation Avancée des Entrevues** : Le développement d'une fonctionnalité permettant une personnalisation plus poussée des entretiens en fonction des besoins spécifiques des entreprises serait également bénéfique. Par exemple, certaines entreprises pourraient souhaiter se concentrer sur des compétences techniques particulières, tandis que d'autres pourraient privilégier les compétences comportementales. Le chatbot pourrait être configuré pour s'adapter à ces préférences, offrant ainsi une expérience plus ciblée et pertinente.
- **Analyse des Réponses avec Intelligence Artificielle** : Enfin, l'intégration de techniques d'intelligence artificielle avancées, telles que l'analyse du sentiment ou la reconnaissance d'émotions, permettrait d'évaluer non seulement le contenu des réponses fournies par les candidats, mais aussi la manière dont ces réponses sont formulées. Cela offrirait une évaluation plus complète et nuancée, aidant ainsi à mieux distinguer les candidats en fonction de leur aisance et de leur attitude pendant l'entretien.

Ces perspectives offrent des pistes prometteuses pour l'amélioration continue du chatbot, rendant l'outil non seulement plus performant, mais aussi plus adaptable aux divers besoins du marché du travail moderne.

1. 2 Critiques sociologiques et limites de l'automatisation

il est important de souligner les implications sociales et humaines des technologies avancées, notamment dans le domaine du recrutement et des interactions homme-machine. Voici comment on pourrait enrichir la critique en incluant cette perspective :

– **Impact sur les Interactions Humaines et la Déshumanisation** : Du point de vue sociologique, certains experts pourraient exprimer des inquiétudes quant à la déshumanisation progressive du processus de recrutement. En déléguant une grande partie de la communication à un chatbot, les entreprises risquent de minimiser l'importance des interactions humaines directes, qui sont souvent perçues comme essentielles pour évaluer des qualités subtiles telles que l'empathie, la capacité à collaborer, ou même l'intuition dans certaines décisions.

– **Répercussions Sociales du Multilinguisme Automatisé** : Le support multilingue, bien qu'utile pour l'accessibilité, pourrait également être perçu comme une manière de standardiser les réponses et de négliger les particularités culturelles qui émergent souvent dans les entretiens en face-à-face. Les sociologues pourraient soulever des préoccupations sur la façon dont cette uniformisation linguistique pourrait entraîner une perte de nuances dans les échanges interculturels, impactant la diversité des réponses et des expériences exprimées par les candidats venant de différents horizons.

– **Renforcement des Inégalités Sociales par l'Automatisation** : Les sociologues pourraient également critiquer l'automatisation du recrutement comme une source potentielle de renforcement des inégalités. En effet, l'IA et les algorithmes, s'ils ne sont pas soigneusement conçus et audités, peuvent perpétuer des biais existants en se basant sur des données historiques, renforçant ainsi des stéréotypes ou des discriminations. Par exemple, un chatbot basé sur des modèles existants pourrait involontairement privilégier des candidats issus de milieux privilégiés ou ayant des parcours "classiques" au détriment de ceux avec des parcours non conventionnels ou marginalisés.

– **Transformation du Travail et Perception du Candidat** : Du point de vue de la sociologie du travail, une autre critique pourrait concerner l'impact de ces technologies sur la perception du rôle du candidat dans le processus de recrutement. La standardisation des questions et l'utilisation d'un chatbot pour interagir avec les candidats peuvent contribuer à une vision plus transactionnelle et instrumentale de l'embauche, où le candidat est perçu davantage comme une "ressource" à optimiser que comme un individu avec des aspirations et des valeurs uniques. Cela pourrait engendrer une aliénation du candidat vis-à-vis du processus et réduire son engagement envers l'entreprise à long terme.

– **Contrôle Social par l'IA et Surveillance des Comportements** : L'analyse des réponses via des techniques avancées telles que l'analyse des émotions pourrait être perçue comme une forme de contrôle social par l'IA. Les sociologues pourraient s'inquiéter du fait que les candidats, conscients qu'ils sont évalués non seulement sur leur contenu mais aussi sur leur façon de parler, leur langage corporel ou leurs émotions, pourraient modifier leur comportement de manière non naturelle, créant un environnement où l'authenticité est compromise au profit de l'optimisation des résultats algorithmiques.

En intégrant ces perspectives sociologiques, on comprend mieux les implications sociales et humaines de l'usage croissant des technologies dans le recrutement. Il devient donc essentiel de considérer ces technologies non seulement sous l'angle de l'efficacité, mais aussi sous celui de leur impact sur les relations humaines, la diversité culturelle, et l'équité sociale.

1.3 Conclusion

Ce projet de développement d'un chatbot intelligent pour les entretiens automatisés a permis d'explorer et d'intégrer différentes technologies et modèles de machine learning, démontrant leur efficacité dans un contexte de recrutement. Nous avons pu établir un cadre robuste, en choisissant des frameworks adaptés, notamment Rasa, pour la gestion des interactions, et en testant plusieurs modèles d'extraction de mots-clés et de génération de questions. Les résultats montrent que le chatbot développé est non seulement capable de mener des entretiens structurés, mais aussi de s'adapter aux réponses des candidats, offrant ainsi une expérience utilisateur plus fluide et pertinente.

En termes de performance, les différents modèles ont été rigoureusement comparés, ce qui a permis de sélectionner ceux qui offrent le meilleur compromis entre précision, rapidité et compréhension contextuelle. Le système a été conçu pour être flexible, permettant des ajustements en fonction des besoins spécifiques des utilisateurs finaux, qu'il s'agisse de recruteurs ou de candidats. Le projet a donc atteint ses objectifs principaux, en développant un outil capable de simplifier et d'automatiser une partie significative du processus de recrutement, tout en assurant une qualité de sélection élevée.

webliographie

- [1] YAKE : https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/Certification_Trainings/Public/8.Keyword_Extraction_YAKE.ipynb
- [2] KeyBERT : <https://maartengr.github.io/KeyBERT/#usage>
- [3] KeyBERT methodology : https://www.vennify.ai/keybert-keyword-extraction/#:~:text=Now%2C%20we%20just%20need%20to,to%20the%20%22top_n%22%20parameter.
- [4] TextRank : <https://www.mdpi.com/2079-9292/12/2/372#:~:text=The%20TextRank%20algorithm%20is%20a,a%20similarity%20matrix%20%5B10%5D.>
- [5] Keywords extraction models : https://www.analyticsvidhya.com/blog/2022/03/keyword-extraction-methods-from-documents-in-nlp/#:~:text=Frequently%20Asked%20Questions,ake_NLTK,that%20is%20supported%20by%20NLTK.
- [6] Keywords extraction online using YAKE : <http://yake.inesctec.pt/>
- [7] TF-IDF : <https://www.kaggle.com/code/rowhitsuwami/keywords-extraction-using-tf-idf-method>
- [8] RAKE : <https://www.analyticsvidhya.com/blog/2021/10/rapid-keyword-extraction-rake-algorithm-in-natur>
- [9] RAKE methodology : <https://pypi.org/project/rake-nltk/>
- [10] data preprocessing : <https://www.mdpi.com/2079-9292/12/2/372#:~:text=The%20TextRank%20algorithm%20is%20a,a%20similarity%20matrix%20%5B10%5D.>
- [11] KeyWords Extraction : [https://monkeylearn.com/keyword-extraction/#:~:text=Keyword%20extraction%20\(also%20known%20as,recognize%20the%20main%20topics%20discussed.](https://monkeylearn.com/keyword-extraction/#:~:text=Keyword%20extraction%20(also%20known%20as,recognize%20the%20main%20topics%20discussed.)
- [12] Bert : <https://medium.com/@prudhviraju.srivatsavaya/bert-bidirectional-embedding-representations-from>
- [13] Optimisation bayésienne Lien : <https://management-datascience.org/articles/24309/#:~:text=L'optimisation%20bay%C3%A9sienne%20est%20une,en%20utilisant%20des%20techniques%20probabilistes.>
- [14] RandomSearch : <https://www.kdnuggets.com/2022/10/hyperparameter-tuning-grid-search-random-search-pyt.html>
- [15] GridSearch : <https://www.mygreatlearning.com/blog/gridsearchcv/>
- [16] RandomForest : <https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.>
- [17] Optimisation methods : <https://towardsdatascience.com/understanding-optimization-algorithms-in-machine>
- [18] Logistique regression : <https://spss.espaceweb.usherbrooke.ca/regression-logistique/>
- [19] Questions Generation : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9886210/>
- [20] T5 model for questions generation : <https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap>
- [21] Bart model for questions generation : <https://pypi.org/project/lmqg/>
- [22] distilbert-base-uncased-distilled-squad model for questions generation : <https://huggingface.co/distilbert/distilbert-base-uncased-distilled-squad>
- [23] Rasa framework : <https://rasa.com/docs/rasa/>
- [24] DialogFlow : <https://cloud.google.com/dialogflow?hl=fr>

- [25] Microsoft bot framework : <https://dev.botframework.com/>
- [26] Botpress framework : <https://botpress.com/fr>
- [27] IBM Watson : <https://www.ibm.com/docs/en/cloud-paks/cp-data/4.8.x?topic=models-watson-machine-learning>
- [28] Wit.ai : <https://wit.ai/>
- [29] chatterBot : <https://chatterbot.readthedocs.io/en/stable/>
- [30] Pandorabots : <https://pandorabots.com/docs/>
- [31] MintHR : <https://minthr.com/fr/>
- [32] ATS : <https://www.manatal.com/features/applicant-tracking-system>
- [33] Algorithmes d'IA : <https://taleez.com/w/blog/intelligence-artificielle-et-recrutement-usages-enjeux-et>
- [34] MYA : <https://www.thestepstonegroup.com/english/>
- [35] Olivia : <https://www.paradox.ai/>
- [36] XOR : <https://xor.ai/>
- [37] Talla : <https://bloomfire.com/>
- [38] Reflektive : <https://www.reflektive.com/>
- [39] Betterworks : <https://www.betterworks.com/>
- [40] The Stanford Question Answering Dataset : <https://rajpurkar.github.io/SQuAD-explorer/>
- [41] Django Framework : <https://www.djangoproject.com>
- [42] Python Framework : <https://www.python.org/>
- [43] PostgreSQL : <https://www.postgresql.org/>
- [44] React : <https://react.dev/>
- [45] Shadcn : <https://ui.shadcn.com/>
- [46] JavaScript : https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- [47] LinkedIn Etude : <https://business.linkedin.com/fr-fr/talent-solutions/ressources/future-of-recruiting/archival/future-of-recruiting-2023>

Résumé

Ce rapport décrit les étapes et outils utilisés pour réaliser le projet d'assistant d'entretien RH basé sur l'IA. Cet assistant facilite les entretiens de pré-sélection, permettant aux RH de gagner du temps et d'améliorer l'efficacité dans le choix des meilleurs profils. Le processus débute par la préparation des données, incluant la normalisation et le nettoyage. Ensuite, les mots clés sont extraits des descriptions de poste à l'aide d'une combinaison du modèle KeyBert et de la méthode TF-IDF, garantissant une extraction optimale. Ces mots clés sont ensuite formatés pour générer des questions via le modèle T5.

La construction du chatbot, chargé de mener les entretiens, est réalisée avec le framework Rasa. Pour évaluer les réponses des utilisateurs, un modèle Random Forest a été entraîné sur la base de données SQUAD, attribuant 1 à une réponse correcte et 0 sinon. Une interface conviviale a été développée en React et Shadcn, tandis que Django a été utilisé pour la partie backend avec une base de données relationnelle PostgreSQL.