



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique



Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Informatique

Département d'intelligence Artificielle et Sciences des Données

Mémoire de Master

Spécialité :

Systemes Informatiques Intelligents

Thème:

**Un système intelligent d'aide à l'audition
au profit de la sécurité publique**

Présenté par:

YOUSFI Zakaria

Encadré par:

Soutenu le:

01/07/2024

Devant le jury:

Projet: SII-19 / 2024

Table des matières

Introduction Générale	1
1 Mise en contexte et problématique	3
1.1 Introduction	3
2 État de l’art	4
2.1 Introduction	4
2.2 Traitement automatique du langage naturel	4
2.3 Représentation du texte	5
2.3.1 Représentation textuelle discrète	6
2.3.2 Représentation textuelle continue	11
2.4 Modèles de réseaux de neurones pour le TALN	13
2.4.1 Réseaux de neurones récurrents	13
2.4.2 LSTM	14
2.4.3 Transformateurs	15
2.4.4 BERT	17
2.4.5 Transformateurs de phrases	19
2.5 Les différentes tâches du TALN	20
2.5.1 Extraction d’informations	21
2.5.2 Reconnaissance d’entités nommées	21
2.5.3 Extraction de relations	22
2.5.4 Classification de texte	25
2.5.5 Similarité des phrases	25
2.5.6 Détection de contradictions	25
2.6 Systèmes de recommandation	26
2.6.1 Introduction	26
2.6.2 Les types des systèmes de recommandation	26
2.6.3 Recommandation des questions	27
2.7 Travaux Connexes	27
2.8 Conclusion	28

3	Conception	29
3.1	Introduction	29
3.2	Diagramme de cas d'utilisation	29
3.3	Diagramme de classes	30
3.4	Approche de détection de la contradiction	31
3.4.1	Introduction	31
3.4.2	Description de l'approche	31
3.4.3	Relations entre personnes	32
3.4.4	Relations entre une personne et un lieu	32
3.4.5	Détection de Contradiction	33
3.4.6	Pourquoi notre approche ?	33
3.5	Approche de recommandation des questions	34
3.5.1	Introduction	34
3.5.2	Description de l'approche	34
3.5.3	Filtrage	35
3.5.4	Similarité de phrases	36
3.5.5	Pourquoi notre approche ?	37
3.6	Conclusion	37
4	Implémentation	38
4.1	Introduction	38
4.2	Présentation de l'environnement de développement	38
4.2.1	Interface	38
4.2.2	Partie serveur	40
4.2.3	Logiciels	42
4.3	Les modèles utilisés	43
4.3.1	Word2vec CBOW	43
4.3.2	Transformateur de phrase	44
4.4	Présentation de l'application	44
4.5	Conclusion	54
	Conclusion Générale& travaux futures	55
	Bibliography	

Table des figures

2.1	Représentation de texte	7
2.2	structure d'un réseau de neurones récurrent standard	13
2.3	Structure d'une cellule de LSTM	14
2.4	L'architecture du transformer	16
2.5	Fenêtre de référence hypothétique de Attention, RNN et LSTM	17
2.6	L'Architecture de BERT pour le Pré-entraînement et le Réglage Fin .	18
3.1	Diagramme de cas d'utilisation de l'acteur « Agent »	30
3.2	Diagramme de cas d'utilisation de l'acteur « Administrateur »	30
3.3	Diagramme de Classe	31
3.4	Schéma global de la recommandation des questions	35
4.1	Interface d'authentification pour l'agent	45
4.2	Interface d'accueil pour l'agent	45
4.3	Interface pour la création des affaires	46
4.4	Interface d'accueil d'une affaire	46
4.5	Interface pour l'ajoute des agents a l'affaire	47
4.6	Interface des Auditions déjà fait pour l'affaire	47
4.7	Interface pour la création d'une nouvelle audition	48
4.8	Interface de saisie des questions et réponses	48
4.9	saisie des question non prédéfinie	49
4.10	saisie des questions prédéfinie, cas relation personne personne	49
4.11	saisie des questions prédéfinie, cas relation personne lieu	50
4.12	Fin de saisie, avant recommandation de questions	50
4.13	recommandation de questions	51
4.14	Après Clôture de l'audition	51
4.15	Saisie d'une deuxième audition	52
4.16	Après Clôture de la deuxième audition, détection de contradiction . .	52
4.17	Interface d'authentification pour l'admin	53
4.18	Interface de la gestion pour l'admin	53

Liste des tableaux

2.1	exemple de l'encodage one-hot	8
2.2	exemple de BOW	8
2.3	Représentation TF-IDF des documents d1 et d2	9
2.4	Matrice de co-occurrence pour le corpus donné	10

Liste d'abréviations

<i>BERT</i>	Bidirectional Encoder Representations from Transformers
<i>BOW</i>	Bag of Words
<i>GPT</i>	Generative Pre-training Transformer
<i>IA</i>	Intelligence Artificielle
<i>KB</i>	Knowledge Base
<i>NER</i>	Named Entity Recognition
<i>RE</i>	Relation Extraction
<i>TALN</i>	Traitement automatique du langage naturelle

Introduction Générale

Contexte

Dans les services de maintien de l'ordre, la collecte et l'analyse précises des déclarations des suspects, des témoins et des victimes sont cruciales pour mener à bien les différentes enquêtes judiciaires ou administratives. Les entretiens et les interrogatoires jouent un rôle central dans ces enquêtes, fournissant des informations essentielles qui peuvent déterminer la direction de l'enquête et influencer les décisions judiciaires. Cependant, la gestion de ces informations présente de nombreux défis, notamment la détection des contradictions dans les déclarations et la formulation de questions pertinentes pour obtenir des informations supplémentaires.

Traditionnellement, les enquêteurs prennent des notes manuelles ou enregistrent des conversations, ce qui nécessite un effort considérable pour analyser et comparer les informations recueillies. Cette approche peut être sujette à des erreurs humaines et à des pertes d'informations importantes. De plus, les enquêteurs peuvent rencontrer des difficultés lorsqu'ils sont confrontés à des déclarations ambiguës ou incohérentes, et lorsqu'ils doivent formuler des questions de suivi pour clarifier ou approfondir les réponses obtenues.

Avec l'avènement des technologies d'intelligence artificielle (IA) et du traitement automatique du langage naturel (TALN), il est désormais possible de développer des systèmes automatisés capables d'analyser les conversations de manière plus efficace et précise. Ces technologies peuvent aider à détecter les contradictions dans les déclarations, en comparant les nouvelles informations avec les données existantes dans le système. Elles peuvent également fournir des recommandations de questions basées sur le contexte de l'entretien, aidant ainsi les enquêteurs à obtenir des informations plus complètes et cohérentes.

Des systèmes similaires ont été développés dans divers domaines de la sécurité publique, tels que la détection de la fraude, la reconnaissance faciale [1] et l'analyse prédictive du crime [2]. Ces systèmes montrent le potentiel de l'IA à transformer les pratiques d'enquête en automatisant des tâches complexes et en fournissant des analyses avancées. Cependant, un système spécifiquement conçu pour enregistrer les conversations d'enquête, détecter les contradictions et recommander des questions en temps réel représente une avancée significative dans ce domaine.

Contribution

Ce projet vise à combler cette lacune en fournissant un outil intégré et automatisé pour les enquêteurs, qui exploite les capacités des technologies d'IA et de TALN. Le système proposé améliorera la précision et l'efficacité des enquêtes, tout en réduisant la charge cognitive des enquêteurs. Cette initiative s'inscrit dans une tendance plus large d'adoption de technologies avancées par les services de sécurité publique pour améliorer leurs capacités d'investigation.

Structure du mémoire

Ce mémoire est structuré en quatre chapitres, abordant tous les aspects liés à notre travail en termes de développement d'un outil intelligent d'aide à améliorer le processus d'audition au sein des services de sécurité. La structure des chapitres est détaillée comme suit :

Chapitre 1 : Dans ce chapitre, nous présenterons brièvement l'organisme d'accueil, en abordant son historique, ses missions, et son organisation. Nous décrirons également le Service Central Informatique du Commandement de la Gendarmerie Nationale et introduirons la problématique de notre projet ainsi que la feuille de route y afférente.

Chapitre 2 : Ce chapitre introduira les notions nécessaires et quelques travaux liés à notre problématique. Dans ce contexte, nous introduirons le Traitement Automatique du Langage Naturel et ses différentes tâches, et nous discuterons des représentations textuelles et des modèles de réseaux de neurones pour le TALN. En plus, Nous aborderons également les systèmes de recommandation et les travaux connexes.

Chapitre 3 : Dans ce chapitre, nous décrirons en détail la conception de notre contribution. Nous présenterons les diagrammes de cas d'utilisation et le diagramme de classes. Nous détaillerons également notre approche pour la détection des contradictions et la recommandation de questions.

Chapitre 4 : Ce chapitre abordera les aspects techniques de la mise en œuvre de notre système, en décrivant en détail l'environnement de développement et les modèles utilisés, ainsi que l'application développée.

Le mémoire se termine par une conclusion résumant les contributions réalisées dans ce travail, des discussions sur les limitations du système développé ainsi que des suggestions et orientations pour d'éventuelles améliorations.

Chapitre 1

Mise en contexte et problématique

1.1 Introduction

intentionally left empty

Chapitre 2

État de l’art

2.1 Introduction

Ce chapitre sera consacré à la présentation des notions de base relatives au domaine de traitement automatique du langage naturel, qui constitue le périmètre de notre étude, ainsi que tous les concepts connexes faisant partie intégrante de son écosystème.

2.2 Traitement automatique du langage naturel

Le traitement automatique du langage naturel (TALN), ou (NLP pour *natural language processing* en anglais), est un domaine essentiel de l’intelligence artificielle qui vise à permettre aux machines de comprendre et d’interpréter le langage humain [3]. Cette discipline repose sur la combinaison de la linguistique informatique, des techniques statistiques, et des algorithmes d’apprentissage automatique et d’apprentissage profond [4].

Les technologies TALN permettent aux ordinateurs de traiter le texte et les données vocales de manière à extraire des informations significatives, comme le contexte, l’intention et les émotions [5]. Grâce à ces capacités, le TALN trouve des applications variées dans de nombreux domaines.

Par exemple, les systèmes de traduction automatique, les assistants virtuels comme **SIRI** d’APPLE¹ et **ALEXA** d’AMAZON², ainsi que les chatbots (agents logiciels qui dialoguent avec les utilisateurs), utilisent des techniques du TALN pour interagir de manière fluide et naturelle.

Le TALN joue un rôle crucial dans l’avancement de diverses technologies contemporaines. Voici quelques exemples illustratifs :

- Modèles de langage comme GPT [6]
- Reconnaissances vocales [7]

1. <https://www.apple.com/fr/siri/>

2. <https://www.alexa.com/>

— Traduction automatique [8], analyse des sentiments [3]

2.3 Représentation du texte

Le domaine du TALN se divise principalement en deux étapes : la première consiste à représenter le texte d'entrée (données brutes) sous format numérique (vecteurs ou matrices), et la deuxième concerne la conception de modèles pour traiter ces données numériques afin d'atteindre un objectif ou une tâche spécifique. Cette section et la section 2.4 se concentre sur la première étape et montre comment, grâce à l'évolution des méthodes de représentation textuelle, le domaine du TALN est passé de la compréhension de fragments isolés à la prise en compte de tous les aspects du texte.

Le contexte est essentiel pour comprendre la signification d'un mot, d'une phrase ou d'un texte. Par exemple, considérez le mot anglais "*bank*". Sans contexte, ce mot est ambigu. Cependant, dans la phrase "*I sat on the bank in the park*" en français "Je me suis assis sur le banc au parc", le contexte lexical indique clairement qu'il s'agit d'un meuble pour s'asseoir. En revanche, dans "*I opened an account at the bank*" en français "J'ai ouvert un compte à la banque", le mot "bank" est une institution financière. Le contexte syntaxique, c'est-à-dire la structure grammaticale et la relation entre les mots dans une phrase, est aussi crucial. Dans "Le chat chasse la souris", l'ordre des mots et leur relation syntaxique indiquent que le chat est l'agent qui effectue l'action de chasser. Le contexte situationnel, qui inclut les informations sur la situation ou le cadre dans lequel la communication a lieu, est également déterminant. Si quelqu'un dit "*I am going to the bank*" en français "Je vais à la banque" en se tenant près d'un lac avec une canne à pêche, le contexte situationnel suggère qu'il s'agit d'un lieu pour s'asseoir et non d'une institution financière. Enfin, le contexte pragmatique, comprenant les intentions et les objectifs de communication des locuteurs ainsi que les conventions sociales et culturelles, aide à interpréter les phrases correctement. Par exemple, "Peux-tu me passer le sel?" est une demande polie et non une question sur la capacité physique de passer le sel. Pour une compréhension précise du langage, le contexte est indispensable car il permet de résoudre les ambiguïtés et de saisir les nuances de signification. Cette nécessité du contexte est également cruciale dans le traitement du langage par les machines. Les modèles du TALN, tels que les embeddings statiques (section 2.3.2) et dynamiques (section 2.4.4), utilisent le contexte pour générer des représentations plus riches et plus précises des mots en fonction de leur usage dans chaque phrase.

Pour que les machines puissent comprendre et analyser les modèles linguistiques, il est essentiel de convertir les mots en nombres. Ce processus, appelé représentation textuelle, est fondamental pour la plupart des tâches de TALN. La

manière dont le texte est représenté influence grandement les performances des modèles d'apprentissage automatique. On distingue deux grandes catégories de représentations textuelles [9] :

- Représentation textuelle discrète
- Représentation textuelle continue

2.3.1 Représentation textuelle discrète

Dans ce type de représentation, les mots sont initialement représentés par des indices correspondant à leur position dans un dictionnaire dérivé d'un corpus. Le processus de création de cette représentation se déroule en plusieurs étapes.

Tout d'abord, un corpus est une collection de textes utilisés comme base pour analyser et construire des représentations. Il peut s'agir de documents, d'articles, de livres, ou tout autre ensemble de textes pertinents pour le domaine d'application envisagé.

À partir du corpus, un dictionnaire de mots est construit. Le processus commence par la collecte de toutes les données textuelles qui feront partie du corpus. Ensuite, le texte est prétraité par des tâches comme la conversion du texte en minuscules, la suppression de la ponctuation, et l'élimination des mots vides (*stop words* en anglais). Une fois le texte prétraité, il est divisé en unités significatives appelées tokens. Optionnellement, les tokens peuvent être filtrés pour enlever ceux qui sont trop fréquents ou trop rares dans le corpus. Enfin, un dictionnaire est créé où chaque mot unique du corpus est associé à un indice unique. Ce dictionnaire est souvent représenté sous forme de table de correspondance ou de liste triée [5].

Prenons un exemple simple. Considérons le corpus constitué de deux documents suivants :

d1 = "Il fait beau aujourd'hui. Le soleil brille."

d2 = "Aujourd'hui est une belle journée."

Et la phrase à représenter : "Il fait beau aujourd'hui"

Le processus est résumé dans la figure 2.1.

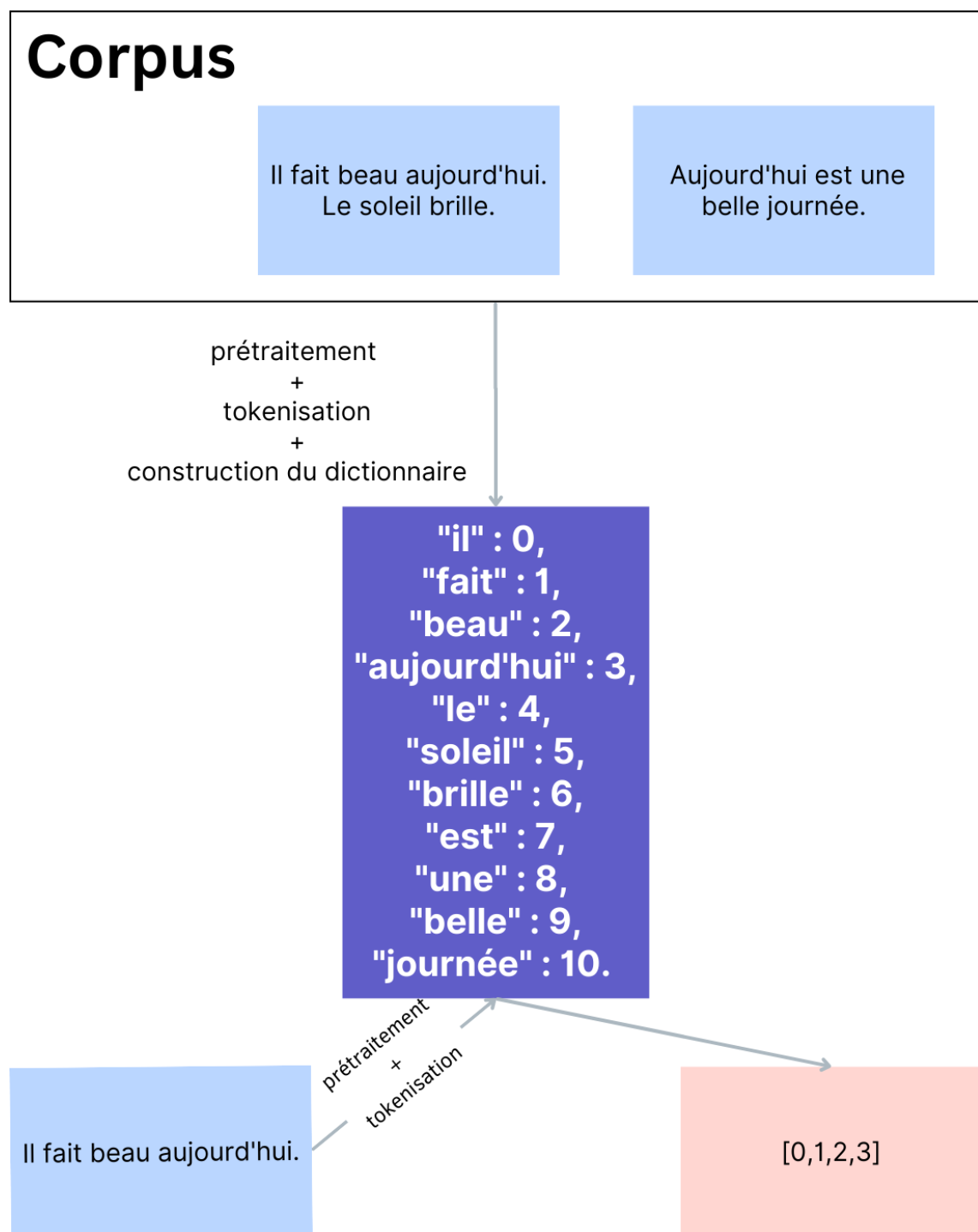


FIGURE 2.1 – Représentation de texte

Les méthodes de représentation textuelle discrètes couramment utilisées sont :

- encodage One-Hot
- Sac de mots
- TF-IDF
- Matrice de co-occurrence

2.3.1.1 encodage One-Hot

Cette méthode [10] représente chaque mot du vocabulaire par un vecteur binaire de la taille du vocabulaire, avec un seul élément à 1 et tous les autres à 0.

Par exemple, la phrase "il fait beau aujourd'hui" est illustrée par la matrice présentée dans le tableau 2.1.

	il	fait	beau	aujourd'hui	le	soleil	brille	est	une	belle	journée
il	1	0	0	0	0	0	0	0	0	0	0
fait	0	1	0	0	0	0	0	0	0	0	0
beau	0	0	1	0	0	0	0	0	0	0	0
aujourd'hui	0	0	0	1	0	0	0	0	0	0	0

TABLEAU 2.1 – exemple de l'encodage one-hot

Pour les vocabulaires de grande taille, cette méthode produit des vecteurs de grande dimension, ce qui peut entraîner des problèmes de stockage et de calcul. De plus, le One-Hot ne capture pas les relations sémantiques entre les mots. Par exemple, les mots "chien" et "loup", qui sont sémantiquement proches, auront des vecteurs orthogonaux (leur produit scalaire est égal à 0) et sans aucune similarité. Il faut noter aussi que parmi les représentations textuelles discrètes, cette méthode est la seule qui conserve l'ordre des mots.

2.3.1.2 Sac de mots

La représentation par sac de mots (BOW pour *bag of words* en anglais), comme son nom l'indique intuitivement, place les mots dans un « sac » et calcule la fréquence d'apparition de chaque mot. Il existe d'autres variantes de cette représentation qui ne considère pas la fréquence mais la présence des mots seulement.

Par exemple, la phrase "il fait beau aujourd'hui" est présentée par la matrice dans le tableau 2.2.

	il	fait	beau	aujourd'hui	le	soleil	brille	est	une	belle	journée
il fait beau aujourd'hui	1	1	1	1	0	0	0	0	0	0	0

TABLEAU 2.2 – exemple de BOW

L'intuition derrière la représentation par sac de mots est que les documents ayant les mêmes mots sont similaires, indépendamment de la position des mots.

Cette méthode est simple à implémenter et efficace pour des tâches comme l'analyse de sentiments [11]. Cependant, comme elle repose sur une correspondance exacte, la similarité des documents n'est correcte que si les mêmes mots exacts sont utilisés. Elle donnera donc un score de similarité inexact pour des documents transmettant la même information mais utilisant des mots différents ou des synonymes.

Le nombre des lignes de la matrice de représentation est significativement plus petite que celle de l'encodage one-hot, mais elle reste grande en dimension pour les vocabulaires de grande taille.

2.3.1.3 TF-IDF

Le TF-IDF (*Term Frequency-Inverse Document Frequency*) est utilisé notamment dans la recherche d'information [12]. Cette approche attribue un poids à chaque mot en fonction de sa fréquence dans le document et de sa rareté dans l'ensemble des documents (corpus). Il combine deux mesures pour évaluer l'importance d'un mot dans un document :

$$\text{TFIDF}(m, d) = \text{TF}(m, d) \times \text{IDF}(m) \quad (2.1)$$

où :

- **TF (Term Frequency)** : la fréquence d'apparition d'un mot m dans un document d .
- **IDF (Inverse Document Frequency)** : Cette valeur, appelée valeur de discrimination [13], mesure la fréquence d'un mot dans l'ensemble des documents et elle est calculée comme suit :

$$\text{IDF}(m) = \log \left(\frac{N}{n_i} \right) \quad (2.2)$$

- N : le nombre total de documents
- n_i : le nombre de documents contenant le mot m

Pour notre exemple, la matrice tf-idf construite est présentée dans le tableau 2.3.

Term	TF-IDF(d1)	TF-IDF(d2)
il	0.301	0.0
fait	0.301	0.0
beau	0.301	0.0
aujourd'hui	0.0	0.0
le	0.301	0.0
soleil	0.301	0.0
brille	0.301	0.0
est	0.0	0.301
une	0.0	0.301
belle	0.0	0.301
journée	0.0	0.301

TABLEAU 2.3 – Représentation TF-IDF des documents d1 et d2

Cette représentation réduit l'impact des mots courants et met en valeur les mots informatifs (rares) ce qui permet de distinguer entre les documents. Les vecteurs

restent de grande dimension pour un grand vocabulaire et elle ne capture pas les relations contextuelles entre les mots.

2.3.1.4 Matrice de co-occurrence

Cette représentation [14] examine la proximité des entités les unes par rapport aux autres au sein d'un texte. Une entité peut être un mot unique, un bi-gramme (séquence de deux mots), ou même une phrase, bien que l'utilisation d'un seul mot soit la méthode la plus courante pour calculer la matrice. En analysant les cooccurrences dans une fenêtre contextuelle (c'est-à-dire un nombre défini de mots autour d'un mot cible), cette matrice permet de révéler les associations et les relations entre différents mots dans un corpus. Cela aide à comprendre comment les mots sont utilisés ensemble et à identifier les tendances et motifs linguistiques présents dans le texte.

Tableau 2.4 montre la matrice de cooccurrence construite pour l'exemple en considération.

	il	fait	beau	aujourd'hui	le	soleil	brille	est	une	belle	journée
il	0	1	0	0	0	0	0	0	0	0	0
fait	1	0	1	0	0	0	0	0	0	0	0
beau	0	1	0	1	0	0	0	0	0	0	0
aujourd'hui	0	0	1	0	0	0	0	1	0	0	0
le	0	0	0	0	0	1	0	0	0	0	0
soleil	0	0	0	0	1	0	1	0	0	0	0
brille	0	0	0	0	0	1	0	0	0	0	0
est	0	0	0	1	0	0	0	0	1	0	0
une	0	0	0	0	0	0	0	1	0	1	0
belle	0	0	0	0	0	0	0	0	1	0	1
journée	0	0	0	0	0	0	0	0	0	1	0

TABLEAU 2.4 – Matrice de co-occurrence pour le corpus donné

Dans cette représentation, chaque vecteur de mot est formé en fonction des mots qui l'entourent, de sorte que le sens du mot est déduit de son contexte. Cependant, un problème majeur avec cette approche est que les mots fréquemment utilisés influencent de manière disproportionnée la mesure de similarité. Les mots fréquents dans le contexte peuvent ne pas fournir beaucoup d'informations sémantiques sur les mots cibles mais affectent tout de même considérablement leur score de similarité. Par exemple, "épicerie" et "sports" ne sont pas étroitement liés, mais s'ils partagent des mots de contexte similaires, leurs vecteurs seront projetés plus près les uns des autres, ce qui entraînera un score de similarité élevé.

Il existe d'autres méthodes de représentation discrète qui ne sont pas mentionnées ici. Pour une discussion plus détaillée et une couverture complète de ces techniques, le lecteur est invité à consulter la référence suivante : [9].

2.3.2 Représentation textuelle continue

La représentation continue, qui est basée sur les réseaux de neurones, permet de capturer des structures complexes dans les données. Les représentations de ce type sont souvent appelées des *embeddings*, qui sont des vecteurs denses contenant des nombres réels. Cela diffère des représentations discrètes où il n'y a que des nombres naturels et où les vecteurs sont clairsemés. Dans ce cadre, la signification d'un mot dépend du contexte fourni par les autres mots, et n'est donc pas indépendante. Les configurations des mots reflètent diverses métriques et concepts présents dans les données. Ainsi, les informations relatives à un mot sont réparties le long du vecteur qui le représente. À l'opposé, dans la représentation discrète, chaque mot est considéré comme unique et indépendant des autres. Ces nouveaux vecteurs de mots sont sensibles au contexte, peuvent identifier des synonymes et des antonymes, et construire des analogies et des catégories de mots, ce qui était impossible avec les approches vues précédemment. Les vecteurs de mots capturent le sens des mots (littéral et implicite) et les représentent en utilisant des valeurs flottantes denses.

Ils représentent à la fois les aspects sémantiques et syntaxiques des mots. Leur longueur se situe généralement entre 100 et 500 dimensions.

Il existe deux types de représentations continues : les *embeddings* de mots statiques, qui attribuent une seule représentation vectorielle fixe à chaque mot indépendamment de son contexte (Word2Vec, GloVe, etc.), et les *embeddings* de mots dynamiques, qui génèrent des représentations contextuelles variant en fonction de l'usage du mot dans chaque phrase (générés par des modèles de langage comme BERT (section 2.4.4) [9]).

Les méthodes courantes de représentation continue incluent :

- Word2Vec
- GloVe
- Transformateurs (voir section 2.4.3)

2.3.2.1 Word2Vec

Word2Vec est un algorithme de word embedding statique, qui représente les mots ou phrases d'un texte sous forme de vecteurs de nombres réels dans un modèle vectoriel. Développé par une équipe de recherche de Google sous la direction de Tomas Mikolov [15], cet algorithme est devenu un outil essentiel dans le TALN [16].

Word2Vec propose deux architectures neuronales principales : CBOW (Continuous Bag of Words) et Skip-Gram.

- **CBOW (Continuous Bag of Words)** : Cette méthode tente de prédire un mot en se basant sur son contexte, c'est-à-dire les termes qui l'entourent dans une phrase. Plus précisément, le modèle prend une fenêtre de mots fixe avant et après un mot cible et utilise ces mots contextuels pour prédire le mot

cible. Par exemple, dans la phrase "Il fait beau aujourd'hui", pour prédire le mot "beau", le modèle utiliserait les mots "Il", "fait", et "aujourd'hui" comme contexte.

CBOW est particulièrement efficace avec des ensembles de données plus petits et offre un temps d'entraînement rapide comparé à Skip-Gram. Cette rapidité est due au fait que CBOW apprend à partir de plusieurs mots contextuels pour chaque mot cible simultanément, ce qui permet une convergence plus rapide. De plus, CBOW tend à bien capturer les relations de sens général et les similarités entre les mots dans des corpus de taille modérée.

- **Skip-Gram** : Cette méthode, à l'inverse de CBOW, prédit le contexte à partir du mot cible. Elle tend à mieux fonctionner avec des ensembles de données plus larges, bien qu'elle nécessite un temps d'entraînement plus long.

2.3.2.2 GloVe

GloVe [17] (Global Vectors for Word Representation) est un algorithme d'embedding statique qui combine les avantages des méthodes basées sur le comptage (comme les matrices de cooccurrence) et des méthodes prédictives (comme Word2Vec), le qualifiant ainsi de méthode hybride. L'objectif principal de GloVe est de trouver des vecteurs de mots qui capturent les relations statistiques globales des cooccurrences de mots. L'algorithme s'appuie sur la fonction d'objectif suivante :

$$J = \sum_{i,j=1}^V f(X_{ij})(w_i^T w_j + b_i + b_j - \log(X_{ij}))^2$$

où :

- X_{ij} représente le nombre de cooccurrences entre le mot i et le mot j
- w_i w_j sont les vecteurs de mots,
- b_i et b_j sont les biais associés à chaque mot,
- f est une fonction de pondération.

Ainsi, GloVe construit des vecteurs de mots w_i et w_j qui respectent les cooccurrences observées X_{ij} , une statistique calculée globalement à partir de la matrice de cooccurrence.

2.4 Modèles de réseaux de neurones pour le TALN

2.4.1 Réseaux de neurones récurrents

2.4.1.1 Introduction

Les réseaux de neurones récurrents (RNN pour *recurrent neural network* en anglais) sont une architecture de réseau de neurones spécialement conçue pour traiter des données séquentielles telles que le texte, l'audio, et plus encore. Contrairement aux réseaux de neurones classiques, qui traitent chaque donnée indépendamment, les RNN conservent une "mémoire" des états précédents. À chaque nouvelle entrée, les RNN concatènent cette entrée avec l'état précédent, ce qui permet au réseau d'apprendre et de comprendre le contexte global, comme le contexte d'une phrase ou d'un extrait audio, et ainsi de prédire le mot suivant [18].

En pratique, un réseau de neurones récurrent se présente comme dans la figure suivante [19] :

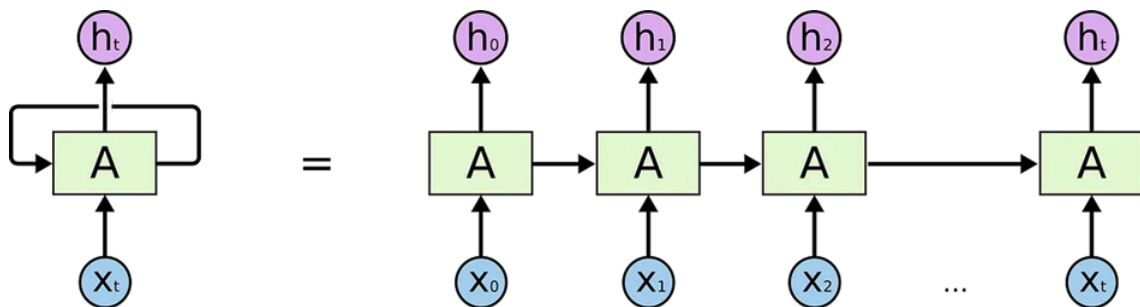


FIGURE 2.2 – structure d'un réseau de neurones récurrent standard

2.4.1.2 Limites des RNN Classiques

Problème de Mémoire à Long Terme

Un des principaux défis des RNN est la gestion de la mémoire à long terme. Prenons l'exemple d'une phrase complexe :

"Le football est un sport très populaire, il est joué dans presque tous les pays du monde. Cristiano Ronaldo et Lionel Messi, deux des plus grands joueurs de tous les temps, sont de véritables légendes dans ce domaine. Le basketball, un autre sport apprécié par des millions de personnes à travers le monde, est basé sur des règles similaires à"

Pour prédire "le football" à la fin, le modèle doit se souvenir que "Le football est un sport" mentionné au début, une tâche difficile pour un RNN classique. Cela est principalement dû au problème de gradients qui disparaissent (*vanishing gradients* en anglais) [18].

Problème de Parallélisation

Les RNN traitant les données de manière séquentielle, leur entraînement ne peut pas tirer pleinement parti des optimisations matérielles et logicielles pour la parallélisation, ce qui ralentit considérablement leur processus d'apprentissage.

Pour surmonter ces limitations, des unités récurrentes plus complexes, telles que les cellules "gated" comme les LSTM (*Long Short-Term Memory*) et les GRU (*Gated Recurrent Units*) [20], ont été développées. Ces cellules améliorent la gestion des flux de données et filtrent les informations pour ne conserver que les plus pertinentes.

2.4.2 LSTM

2.4.2.1 Introduction

Les LSTM (*Long Short-Term Memory* en anglais ou Mémoire à Long Court Terme en français) ont été créés pour résoudre le problème de la mémoire à long terme rencontré par les RNN classiques [18]. Ces réseaux utilisent des cellules mémoire plus complexes, appelées "cellules à portes" (*gated cells*), qui contrôlent le flux de données. Une cellule LSTM se compose de trois états et trois portes internes, comme illustré dans la figure 2.3 [21].

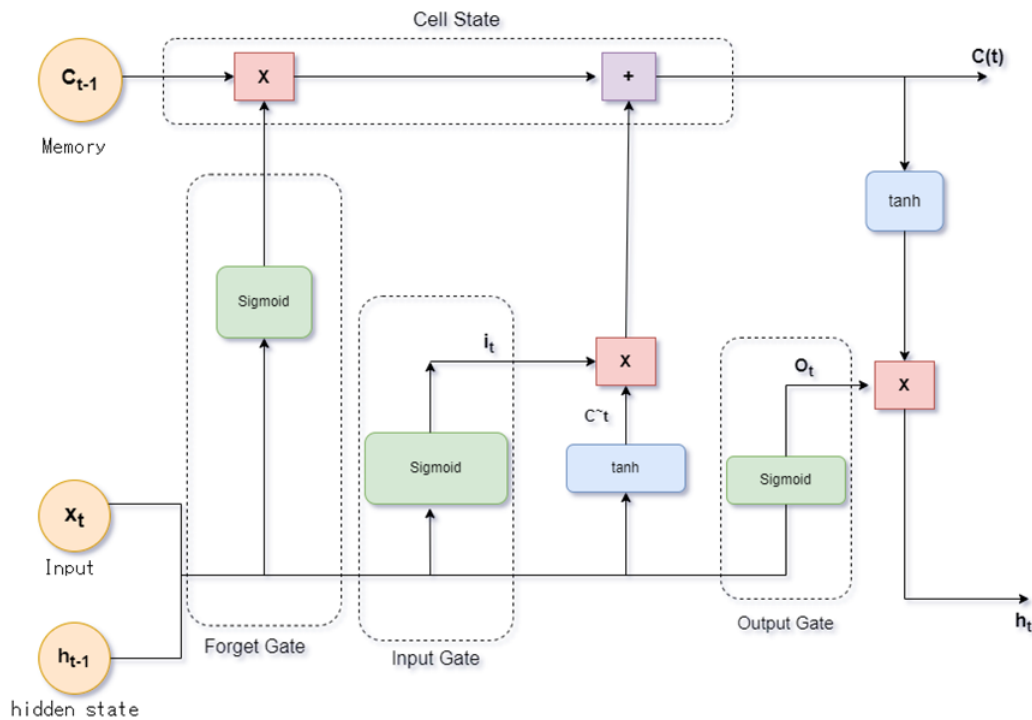


FIGURE 2.3 – Structure d'une cellule de LSTM

2.4.2.2 Limites des LSTM

Complexité et ressources computationnelles

Les LSTM sont plus complexes que les RNN traditionnels en raison de leur architecture avec des portes d'entrée, de sortie et d'oubli. Cette complexité accrue entraîne une augmentation des ressources computationnelles nécessaires pour l'entraînement et l'inférence. De plus, la présence de nombreuses hyperparamètres à régler peut compliquer le processus de formation et nécessiter des efforts supplémentaires pour trouver les configurations optimales.

Problèmes de parallélisation

Les LSTM, comme tous les réseaux de neurones récurrents (RNN), traitent les données de manière séquentielle. Cette caractéristique séquentielle limite leur capacité à tirer parti des optimisations matérielles et logicielles pour la parallélisation, ce qui ralentit considérablement le processus d'apprentissage et d'inférence. Cette limitation rend les LSTM moins efficaces pour les applications nécessitant une grande vitesse de traitement.

2.4.3 Transformateurs

2.4.3.1 Introduction

Le transformateur (*transformer* en anglais), une innovation majeure dans le domaine de l'apprentissage profond, a été dévoilé en 2017 dans l'essai "Attention is all you need" [22]. Conçu par une équipe de chercheurs de Google Brain et Google Research, ce modèle révolutionnaire a été conçu pour pallier les lacunes observées dans les architectures précédentes, telles que les RNN et les LSTM.

Au cœur du Transformer réside son mécanisme d'attention, qui permet au modèle de prendre en compte le contexte global de l'entrée. Ce mécanisme permet aux transformateurs d'avoir une mémoire à très long terme. Un modèle de transformateur peut « assister » ou « se concentrer » sur tous les tokens précédents qui ont été générés. Contrairement aux RNN, qui traitent les données de manière séquentielle et donc lente, le Transformer peut exploiter la parallélisation pour accélérer considérablement le processus d'entraînement. La figure 2.4 montre l'architecture du transformer [22].

2.4.3.2 Mécanisme d'attention

Le concept fondamental du mécanisme d'attention, connu sous le nom de *self-attention*, est au cœur de l'architecture Transformer. Ce mécanisme analyse les relations entre les différents mots d'une séquence afin de produire une représentation contextuelle pertinente, enrichissant ainsi les informations présentes dans notre entrée. Au fur et à mesure que le modèle génère le texte mot par mot, il peut « assister » ou « se concentrer » sur les mots pertinents pour le mot généré. La capacité de savoir à quels mots prêter attention s'apprend au cours de l'entraînement par backpropagation.

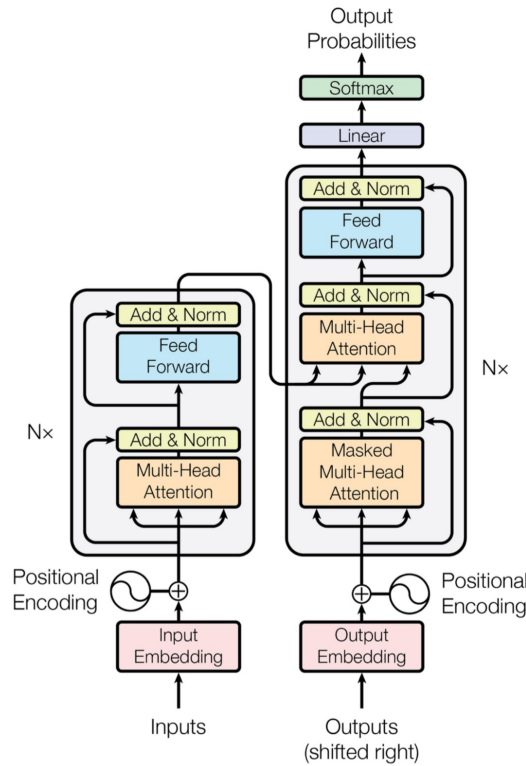


FIGURE 2.4 – L'architecture du transformer

La figure 2.5 montre la différence (hypothétique) de la fenêtre de référence entre ce mécanisme et les méthodes vues précédemment. Contrairement aux autres, le mécanisme d'attention possède une fenêtre de référence infinie.

Le processus de self-attention repose sur trois vecteurs clés :

- Vecteur de requête (q).
- Vecteur de clé (k).
- Vecteur de valeur (v).

Chacun de ces vecteurs est indexé en fonction de la position du mot dans la séquence (par exemple, q_1 , k_1 et v_1 pour le premier mot). Ces vecteurs sont calculés en multipliant l'embedding de la séquence d'entrée par trois matrices, qui sont ajustées pendant le processus d'entraînement du Transformer.

2.4.3.3 Encodeur et Décodeur

À un niveau élevé, l'encodeur mappe une séquence d'entrée en une représentation continue abstraite qui contient toutes les informations apprises de cette entrée. Le décodeur prend ensuite cette représentation continue et génère étape par étape une seule sortie tout en recevant également la sortie précédente.

En ce qui concerne l'architecture spécifique du Transformer, l'encodeur et le décodeur jouent des rôles essentiels. L'encodeur est constitué d'un empilement de 6 couches identiques, chaque couche comprenant deux sous-couches : un mécanisme de multi-têtes self-attention et un réseau de feed-forward. L'entrée de chaque encodeur

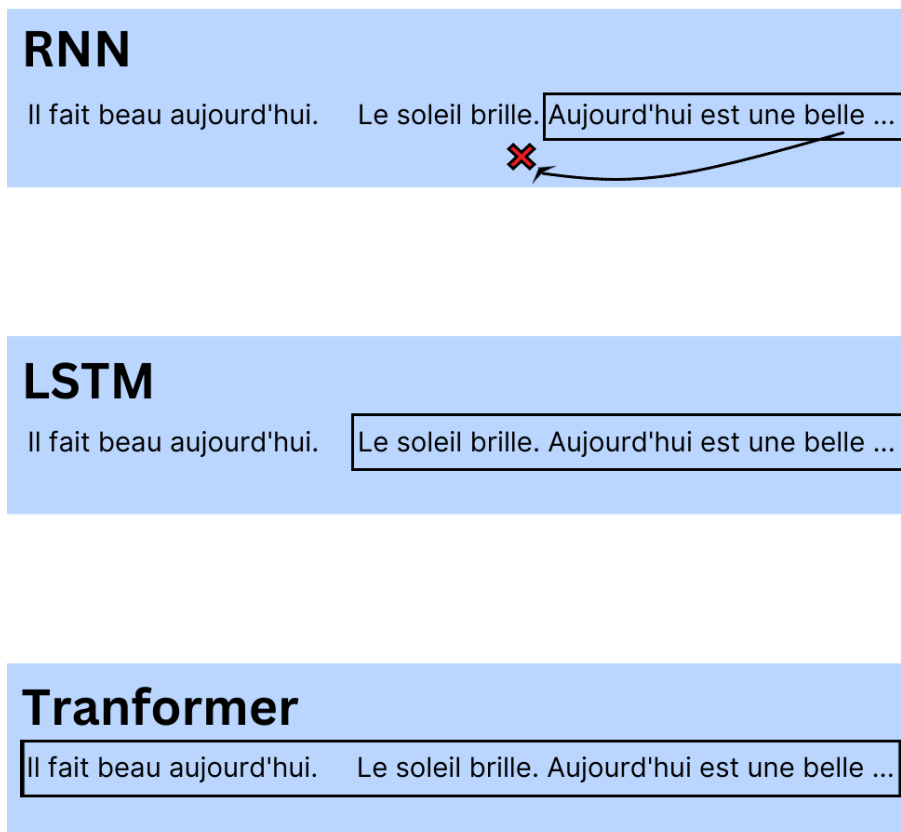


FIGURE 2.5 – Fenêtre de référence hypothétique de Attention, RNN et LSTM

est la sortie du précédent, avec le premier encodeur recevant un vecteur d’embedding et la sortie du dernier encodeur étant utilisée dans le décodeur.

Le décodeur, similaire à l’encodeur, est également composé de 6 couches identiques, intégrant une sous-couche de self-attention et un réseau feed-forward. De plus, le décodeur comprend une couche d’attention encodeur-décodeur, qui permet au décodeur de se concentrer sur les parties pertinentes de la séquence d’entrée pendant le processus de décodage. Finalement, le dernier décodeur est connecté à un bloc de réseau neuronal linéaire plus Soft-max, qui identifie les correspondances dans le vocabulaire pour les sorties du dernier encodeur.

2.4.4 BERT

2.4.4.1 Introduction

BERT (*Bidirectional Encoder Representations from Transformers*) se présente comme un modèle transformateur révolutionnaire conçu par une équipe de Google en 2018 [23]. Il réinvente la pré-entraînement de représentations profondes bidirectionnelles à partir de texte non étiqueté en conditionnant conjointement les contextes gauche et droit, dans le but d’acquérir une compréhension approfondie du contexte linguistique.

Grâce au réglage fin (*fine-tuning* en anglais), réalisé en ajoutant une seule couche de sortie supplémentaire, BERT peut produire des résultats de pointe. Cette avancée

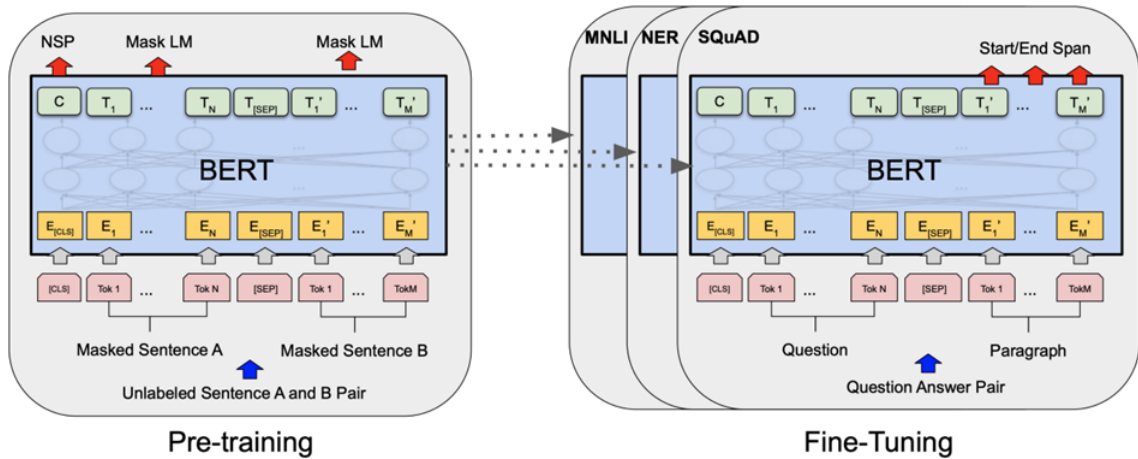


FIGURE 2.6 – L'Architecture de BERT pour le Pré-entraînement et le Réglage Fin

est rendue possible grâce à la technique de Masked LM, permettant un entraînement bidirectionnel dans des modèles jusque-là inatteignables.

La figure 2.6 montre les deux grande aspect du bert, le pré-entraînement et le réglage fin [23].

2.4.4.2 Aperçu Architectural de BERT

BERT se compose d'un encodeur de transformateur multicouche bidirectionnel, avec deux variantes : le modèle de base et le modèle large. La distinction réside dans le nombre de couches, la taille cachée et le nombre de têtes d'auto-attention. La représentation des données d'entrée peut englober à la fois des phrases simples et des paires de phrases au sein d'une seule séquence de tokens, facilitant ainsi la mise en œuvre de tâches de TALN sur des données conversationnelles.

2.4.4.3 Pré-entraînement

Deux concepts clés sous-tendent l'ascension de BERT en tant que modèle standard pour l'apprentissage par transfert :

- **Masked LM** : Pour chaque séquence, un pourcentage de mots est remplacé par le token [MASK], incitant le modèle à les prédire en fonction du contexte fourni par les autres mots non masqués de la séquence.
- **Prédiction de la Phrase suivante (NSP)** : Au cours de l'entraînement de BERT, le modèle reçoit des paires de phrases en entrée et apprend à prédire si la deuxième phrase de la paire est la phrase suivante dans le document d'origine. Tout au long de l'entraînement, 50% des entrées sont une paire dans laquelle la deuxième phrase est la phrase suivante dans le document d'origine, tandis que dans les 50% restants, une phrase aléatoire du corpus est choisie comme deuxième phrase. Ce concept est essentiel pour comprendre la relation entre deux phrases, cruciale pour des tâches de TALN telles que la réponse aux questions et l'inférence en langage naturel.

2.4.4.4 Réglage Fin

Le réglage fin permet à BERT de modéliser de nombreuses tâches de TALN, à un coût relativement faible, en ajoutant simplement une couche au modèle de base. Par exemple :

- Pour affiner BERT pour la tâche de **l'extraction de relations** (voir section 2.5.3), vous pouvez ajouter une couche de classification au-dessus du modèle BERT afin de prédire la relation entre deux entités dans une phrase. Le processus consiste à prétraiter les données d'entrée pour inclure des marqueurs indiquant les entités, à passer ces entrées à travers BERT, puis à utiliser les états cachés finaux des marqueurs d'entités comme entrées pour la couche de classification. L'entraînement est effectué à l'aide d'un ensemble de données annotées où les relations entre les entités sont étiquetées, et le modèle est affiné en optimisant une fonction de perte de classification, telle que la perte d'entropie croisée [24].
- pour **la reconnaissance d'entités nommées (NER)** (voir section 2.5.2), il faut ajuster le modèle pré-entraîné de BERT à la tâche NER en ajoutant une couche de classification au-dessus de celui-ci. Le processus commence par la tokenisation du texte d'entrée à l'aide du tokenizer de BERT, en veillant à ce que l'entrée soit compatible avec l'architecture du modèle. Chaque mot tokenisé est ensuite introduit dans le modèle BERT, qui génère des embeddings de mots contextualisés. Une couche linéaire est ajoutée au-dessus des sorties de BERT, et l'ensemble du modèle est ajusté sur des données étiquetées pour la NER en utilisant une fonction de perte appropriée, telle que la perte par entropie croisée. Pendant l'entraînement, le modèle apprend à classer chaque token dans des catégories d'entités prédéfinies (par exemple, personne, organisation, lieu) en fonction du contexte fourni par les mots environnants [23].

2.4.5 Transformateurs de phrases

2.4.5.1 Introduction

Les transformateurs de phrases (*sentence transformers* en anglais) sont un type de modèle transformateur conçu pour générer des embeddings de phrases de haute qualité, qui sont des vecteurs de taille fixe capturant la signification sémantique des phrases. Contrairement aux modèles transformers traditionnels comme BERT, qui génèrent des embeddings pour des tokens individuels, les transformers de phrases produisent des embeddings pour des phrases entières. Cela les rend particulièrement utiles pour des tâches impliquant la comparaison ou la compréhension de phrases complètes, telles que la similarité textuelle sémantique et le regroupement de phrases [25].

2.4.5.2 Développement et Évolution des Transformateurs de Phrases

Le développement des transformateurs de phrases a été marqué par plusieurs innovations clés. L'un des travaux fondamentaux est le Sentence-BERT (SBERT) [25], qui adapte l'architecture BERT pour des tâches au niveau des phrases. SBERT a introduit une structure de réseau siamois qui ajuste BERT sur une combinaison de tâches de classification et de régression pour produire des embeddings de phrases significatifs.

BERT [23] et RoBERTa [26] ont atteint des performances de pointe dans les tâches de régression de paires de phrases telles que la similarité sémantique des textes (STS pour *semantic textual similarity* en anglais) [27]. Cependant, ces modèles nécessitent que les deux phrases soient traitées simultanément, ce qui entraîne une surcharge computationnelle significative. Par exemple, identifier la paire la plus similaire dans une collection de 10 000 phrases avec BERT nécessite environ 50 millions de calculs d'inférence, ce qui peut prendre environ 65 heures. Cette exigence computationnelle rend BERT impraticable pour des tâches telles que la recherche de similarité sémantique et le clustering non supervisé. Les modèles ultérieurs ont construit sur cette approche, améliorant l'efficacité et la qualité des embeddings [28].

2.4.5.3 Modèle All-MiniLM-L6-v2

Le modèle all-MiniLM-L6-v2 [29] est une version compacte et efficace du MiniLM (Miniature Language Model) [28], conçu par Hugging Face. Il a été développé pour créer des embeddings de phrases en utilisant des ensembles de données volumineux au niveau des phrases grâce à un objectif d'apprentissage contrastif auto-supervisé. Le modèle pré-entraîné, MiniLM-L6-H384-uncased [30], a été affiné sur un ensemble de données contenant 1 milliard de paires de phrases. Cette approche a employé un objectif d'apprentissage contrastif (*contrastive learning objective* en anglais) où le modèle, étant donné une phrase d'une paire, devait prédire laquelle parmi un ensemble de phrases échantillonnées aléatoirement était réellement appariée avec elle dans l'ensemble de données. Il utilise une architecture plus petite (6 couches) par rapport aux 12 ou 24 couches de BERT, le rendant ainsi beaucoup plus rapide tout en maintenant des performances compétitives.

2.5 Les différentes tâches du TALN

Les subtilités du langage humain rendent extrêmement complexe la tâche de développer des logiciels capables de comprendre avec précision le sens voulu des textes ou d'autres données. Des phénomènes tels que les homonymes, les homophones, le sarcasme, les métaphores et les variations de la structure des phrases ne sont que quelques exemples des défis que présente le langage humain.

Ces subtilités, qui peuvent prendre des années à être maîtrisées par les humains, doivent être apprises par les programmeurs dès le début pour que les applications basées sur le langage naturel puissent reconnaître et comprendre correctement ces nuances, et ainsi être véritablement utiles.

Plusieurs tâches sont définies pour aider les ordinateurs à interpréter les données textuelles. Voici quelques exemples de ces tâches qui sont pertinentes pour la réalisation de notre projet :

2.5.1 Extraction d'informations

L'extraction d'informations est la tâche du TALN qui extrait des informations sémantiques structurées à partir de texte. Ces informations incluent des relations binaires - par exemple, des interactions biochimiques entre deux protéines [31] ou des événements n-aires - c'est-à-dire des événements avec plus de deux arguments tels que des attaques terroristes, où chaque attaque est associée à plusieurs arguments, y compris l'emplacement de l'attaque, l'identité de l'attaquant, le nombre de victimes, le montant des dommages matériels, et ainsi de suite [32]. L'extraction d'informations permet de nombreuses applications réelles importantes telles que la découverte de traitements potentiels pour les maladies ou la surveillance des attaques terroristes à partir de documents de presse.

2.5.2 Reconnaissance d'entités nommées

La reconnaissance d'entités nommées (NER pour *named entity recognition* en anglais) est une sous-tâche de l'extraction d'informations visant à identifier et classer les mots clés, appelés entités, présents dans un document. Cette technologie permet de regrouper ces entités en catégories prédéfinies. Par exemple, dans un texte, la NER peut détecter et distinguer des mentions de personnes et de lieux, qui appartiennent à des catégories distinctes.

Les chercheurs ont développé de nombreuses techniques pour la NER au cours des 30 dernières années, en commençant par des efforts initiaux qui intégraient la NER de manière limitée dans l'extraction d'informations. La recherche initiale sur la NER utilisait des modèles basés sur des règles conçues manuellement, adaptés à des corpus de textes spécifiques. Les efforts de standardisation ont conduit à la création de jeux de données de référence et de conférences pour faciliter les progrès. Les méthodes d'apprentissage supervisé ont produit des résultats à la pointe de la technologie en se formant sur de grands corpus annotés, tout en abordant diverses limitations comme le problème du biais des étiquettes. Des méthodes semi-supervisées et non supervisées ont émergé pour tirer parti des textes non structurés, en extrayant des informations contextuelles à partir de vastes quantités de données. Les avancées récentes incluent des approches de

réseaux neuronaux, telles que les modèles d'apprentissage profond et les embeddings de mots, qui ont considérablement amélioré les performances de la NER en capturant des dépendances complexes et des structures hiérarchiques. Ces développements ont collectivement amélioré la précision et l'applicabilité de la NER dans différents domaines [33].

2.5.3 Extraction de relations

2.5.3.1 Introduction

L'extraction de relations (RE pour *relation extraction* en anglais) est une sous-tâche de l'extraction d'informations qui vise à identifier et à extraire les liens sémantiques entre différents éléments dans un texte. Ces éléments peuvent être des entités telles que des personnes, des lieux ou des événements, et les relations entre eux peuvent être diverses, allant des simples associations binaires aux structures plus complexes impliquant plusieurs entités. L'objectif principal de l'extraction de relations est de transformer le texte non structuré en données exploitables, facilitant ainsi la compréhension automatique des informations contenues dans les documents textuels. Cette tâche est cruciale dans de nombreux domaines, notamment la réponse aux questions et l'extraction de biotexte [34].

2.5.3.2 Les approches du RE

Plusieurs approches existent pour l'extraction de relations :

- **Approche basé sur les règles**

Ces approches sont également connues sous le nom de méthodes fondées sur des motifs (*patterns* en anglais) définis manuellement. Elles établissent un ensemble de motifs d'extraction pour un certain nombre de relations prédéfinies, motifs qui sont ensuite appliqués au texte. Lorsqu'un motif est trouvé dans le texte, une relation correspondante est identifiée.

Par exemple, un motif pour les hyponymes comme 'tels X que Y' appliqué au texte 'tels acteurs que Angelina' permet d'identifier la relation hyponyme(acteur, angelina).

Les méthodes basées sur des règles nécessitent une expertise spécialisée et une connaissance approfondie de la langue pour formuler des modèles d'extraction. Ces approches sont spécifiques à un domaine particulier, s'appuyant sur une structure de document stable et des relations cibles définies à l'avance. Ainsi, lorsqu'on change de domaine, il faut redéfinir les relations et les motifs d'extraction, ce qui nécessite un effort manuel considérable et rend ces méthodes inadaptées à des corpus diversifiés [35].

- **Approche basé sur l'apprentissage supervisé**

Les méthodes supervisées nécessitent une grande quantité de données d'entraînement, annotées avec un ensemble d'entités et de relations. Elles utilisent ces données d'entraînement pour former un classificateur, qui extraira ensuite les relations des données de test [35].

— Approche basé sur l'apprentissage non supervisé

Les méthodes non supervisées n'exigent pas de données annotées. La plupart des techniques non supervisées d'extraction de relations reposent sur une approche de clustering. L'une des premières stratégies non supervisées d'extraction de relations utilisant le clustering s'est appuyée sur un étiqueteur d'entités nommées pour extraire ces entités, se concentrant ainsi uniquement sur les relations entre entités nommées. Les étapes principales de cette méthode d'apprentissage non supervisée sont les suivantes [35] :

1. Identification des entités nommées dans le corpus de texte
2. Identification des entités nommées co-occurentes et de leur contexte
3. Regroupement des paires d'entités en fonction de la similarité de leur contexte
4. Attribution d'un nom de relation sémantique à chaque groupe.

— Approche basé sur l'apprentissage semi-supervisé

Créer des données pour les méthodes supervisées d'extraction de relations est coûteux et chronophage. Cependant, les méthodes supervisées peuvent utiliser des algorithmes de bootstrap pour automatiser la génération de données étiquetées. Cette méthode présente deux avantages principaux :

- Réduction de l'effort nécessaire pour créer des données étiquetées
- Utilisation des données non étiquetées disponibles gratuitement

L'algorithme de bootstrap s'appuie sur une grande quantité de données non étiquetées et un petit ensemble d'exemples initiaux illustrant la relation souhaitée. Par exemple, pour la relation "CapitaleDe", des exemples initiaux comme (New Delhi, Inde), (Canberra, Australie), et (Londres, Angleterre) peuvent être utilisés pour former un modèle d'extraction. Ce modèle pourra ensuite identifier des relations similaires, comme (Paris, France) [35].

— Approche de supervision distante

Ces méthodes, également appelées méthodes faiblement supervisées ou basées sur la connaissance, génèrent automatiquement des données d'entraînement en alignant le texte avec une base de connaissances (KB pour *knowledge base* en anglais), éliminant ainsi le besoin d'étiquetage manuel. La supervision distante fonctionne sur le principe que si deux entités partagent une relation dans une KB, toutes les phrases mentionnant ces entités peuvent exprimer cette relation.

Cette approche utilise des bases de connaissances comme **FREEBASE**³ pour extraire les relations entre entités. Lorsqu'une paire d'entités apparaît dans une phrase et dans la KB, la phrase est heuristiquement liée à la relation correspondante de la KB. Par exemple, dans la phrase "Bill Gates est le fondateur de Microsoft", si "Bill Gates" et "Microsoft" sont listés comme un triplet (entité1 : Bill Gates, entité2 : Microsoft, relation : fondateur_de) dans Freebase, alors ces entités illustrent la relation "fondateur_de" [35].

— **Approche basé apprentissage profond**

L'utilisation de l'apprentissage profond a transformé l'extraction de relations en permettant une représentation complexe et hiérarchique des données textuelles. Les réseaux de neurones profonds, tels que les RNN (section 2.4.1) et les transformers (section 2.4.3), sont particulièrement efficaces pour saisir les nuances et le contexte subtil des relations entre entités grâce à leur capacité à considérer la séquentialité du langage naturel. En outre, les modèles de langue pré-entraînés comme BERT (section 2.4.4) ont montré qu'ils peuvent capturer des informations sémantiques riches, essentielles pour l'extraction précise des relations [35].

2.5.3.3 Langues à faible ressource

Dans les langues comme l'anglais, qui bénéficient de ressources abondantes, un vaste corpus étiqueté est disponible pour former des modèles dans diverses tâches du TALN. En revanche, les langues à ressources limitées telles que l'arabe disposent de corpus étiquetés très restreints. Bien que les récents progrès des modèles d'apprentissage profond pour l'extraction de relations aient montré des résultats prometteurs, la performance de ces modèles diminue considérablement lorsque le nombre d'exemples d'entraînement est limité. [35].

2.5.3.4 Extraction de relations spécifiques au domaine

Les systèmes d'extraction de relations peuvent être appliqués à diverses sources textuelles telles que les articles médicaux, les documents juridiques, les articles scientifiques et les actualités. Avec le nombre croissant de documents textuels disponibles, il y a une demande accrue pour traiter ces documents et stocker l'information sous forme de bases de connaissances structurées, essentielles pour les applications en aval. Les systèmes RE peuvent accélérer la création de bases de connaissances dans différents domaines en extrayant automatiquement les relations et en les fournissant aux conservateurs humains pour vérification.

Utiliser des modèles de pointe pour l'extraction de relations à partir de textes spécifiques à un domaine présente deux principaux défis. Premièrement, les

3. [https://en.wikipedia.org/wiki/Freebase_\(database\)](https://en.wikipedia.org/wiki/Freebase_(database))

modèles de représentation des mots comme **word2vec** (section 2.3.2.1), **GloVe** (section 2.3.2.2) et **BERT** (section 2.4.4) sont généralement entraînés sur des textes de domaine général, tels que Wikipédia, qui diffèrent considérablement des corpus spécifiques au domaine, ce qui peut dégrader les performances du modèle RE. Ces représentations de mots doivent être entraînées sur des corpus spécifiques pour une performance optimale. Deuxièmement, des modèles comme BERT, entraînés sur des textes généraux, ne performant pas bien sur des tâches spécifiques sans adaptation. Des chercheurs ont développé des versions spécifiques de BERT, telles que BioBERT, SciBERT, ClinicalBERT et LegalBERT, en affinant BERT sur des corpus spécialisés, atteignant des performances de pointe. De plus, l'entraînement des modèles à partir de zéro sur des données spécifiques au domaine surpasse considérablement l'affinage des modèles généraux [35].

2.5.4 Classification de texte

La classification de texte est l'une des tâches du TALN les plus largement utilisées en raison de ses nombreuses utilisations dans le monde réel telles que l'analyse des sentiments, la recherche d'informations, la classification des actualités, et l'étiquetage de sujets.

Cette tâche consiste à attribuer des catégories ou des étiquettes à des documents textuels en fonction de leur contenu. Elle permet de rendre les informations non structurées plus accessibles et exploitables. En utilisant des algorithmes sophistiqués, la classification de texte aide à automatiser et à accélérer le processus de tri et d'organisation de grandes quantités de données textuelles, ce qui est essentiel dans de nombreux domaines, de la recherche académique à l'industrie [36].

2.5.5 Similarité des phrases

La similarité des phrases (*sentence similarity* en anglais) est la tâche qui consiste à déterminer à quel point deux textes sont similaires. Les modèles de similarité des phrases convertissent les textes d'entrée en vecteurs (embeddings) qui capturent les informations sémantiques et calculent à quel point ils sont proches (similaires) entre eux. Cette tâche est particulièrement utile pour la recherche d'informations et le clustering [37].

2.5.6 Détection de contradictions

La détection de contradictions est une tâche difficile dans le domaine du TALN en raison de la variété des façons dont les contradictions apparaissent dans les textes [38]. Elle implique l'identification et le traitement des énoncés qui sont mutuellement exclusifs ou en conflit de signification.

Une approche courante de la détection de contradiction consiste à utiliser des modèles d'apprentissage automatique, en particulier des réseaux neuronaux. Ces modèles sont entraînés sur des ensembles de données annotés contenant des paires de phrases étiquetées comme contradictoires ou non-contradictaires [39]. Des techniques telles que les réseaux siamois [40] ou les architectures basées sur les transformateurs comme BERT sont utilisées à cette fin. En apprenant des représentations contextuelles des phrases, ces modèles peuvent capturer les relations sémantiques et identifier les contradictions [41].

2.6 Systèmes de recommandation

2.6.1 Introduction

Les systèmes de recommandation sont des outils essentiels dans divers domaines, allant du commerce en ligne aux plateformes de streaming, en passant par les réseaux sociaux. Une définition des Systèmes de recommandation est la suivante : "Outils, logiciels, et techniques qui fournissent des suggestions personnalisées, guidant l'utilisateur, dans un grand espace de données, vers des ressources susceptibles de l'intéresser" [42]. Ils utilisent des techniques d'apprentissage automatique et de TALN pour analyser les préférences et les comportements des utilisateurs afin de proposer des contenus ou des produits pertinents [43]. Par exemple, Amazon et Netflix emploient des algorithmes de filtrage collaboratif et de filtrage basé sur le contenu pour améliorer l'expérience utilisateur [44]. En combinant des méthodes telles que les réseaux de neurones et les modèles de graphes de connaissances, ces systèmes peuvent fournir des recommandations personnalisées et contextuellement adaptées [45]. De plus, les avancées récentes dans l'utilisation des modèles de transformateurs, comme BERT et GPT, ont permis d'améliorer considérablement la précision des recommandations en comprenant mieux le contexte et les intentions des utilisateurs [46].

2.6.2 Les types des systèmes de recommandation

— Filtrage basé sur le contenu

Le filtrage basé sur le contenu (*Content-based filtering* en anglais) est une technique clé dans les systèmes de recommandation, utilisée pour proposer des éléments similaires à ceux que l'utilisateur a déjà appréciés. Contrairement au filtrage collaboratif, qui repose sur les préférences des autres utilisateurs, le filtrage basé sur le contenu analyse les caractéristiques des éléments eux-mêmes pour fournir des recommandations. Par exemple, dans le domaine de la musique ou des films, les systèmes de recommandation peuvent utiliser des

informations telles que les genres, les acteurs, ou les artistes pour suggérer des contenus similaires [47]. Cette méthode utilise souvent des techniques du TALN pour extraire et analyser les caractéristiques des textes, telles que les descriptions de produits ou les résumés de films [48]. De plus, l'intégration des réseaux de neurones et des modèles de représentation de texte comme TF-IDF ou les embeddings de mots a permis d'améliorer la précision et la pertinence des recommandations basées sur le contenu [49].

— **Filtrage collaboratif**

La mise en œuvre la plus simple et la plus originale de cette approche consiste à recommander à un utilisateur actif les items que d'autres utilisateurs ayant des goûts similaires ont aimés dans le passé. La similarité de goût entre deux utilisateurs est calculée en fonction de la similarité de leur historique de notation. Le filtrage collaboratif (*collaborative filtering* en anglais) est la technique la plus populaire et la plus répandue dans les systèmes de recommandation [50].

2.6.3 Recommandation des questions

La recherche spécifiquement dédiée aux systèmes de recommandation de questions dans l'analyse des auditions est limitée, mais le domaine plus large des systèmes de recommandation et de la recherche d'informations fournit des méthodologies pertinentes applicables à ce domaine. Les recherches spécifiques sur la recommandation de questions sont également limitées [51], mais il existe des systèmes à grande échelle qui utilisent la recommandation de questions. Par exemple, les communautés de questions comme **Quora**⁴ utilisent des systèmes de recommandation pour suggérer aux utilisateurs des questions qui pourraient les intéresser.

2.7 Travaux Connexes

Dans le domaine de l'analyse des auditions des services de maintien de l'ordre, en particulier en ce qui concerne la détection des contradictions et la recommandation de questions, il existe peu de systèmes ou de projets directement comparables à la portée et aux objectifs de notre travail. Alors que les technologies de l'IA ont été largement adoptées dans divers secteurs et applications, leur intégration dans le contexte spécialisé de l'analyse des auditions reste relativement peu explorée.

Malgré la reconnaissance croissante du potentiel de l'IA pour améliorer les processus d'investigation, notamment dans des tâches telles que l'examen de l'écriture manuscrite [52], la littérature et les systèmes existants ignorent largement

4. <https://www.quora.com>

les exigences nuancées de l'analyse des auditions des services de maintien de l'ordre. L'absence de systèmes dédiés à la détection des contradictions et à la recommandation de questions dans ce domaine souligne la nécessité d'approches adaptées aux défis spécifiques et aux objectifs des agences d'investigation.

De plus, bien que les systèmes pilotés par l'IA pour l'analyse textuelle et les applications du TALN aient été largement étudiés dans d'autres domaines, tels que le service client ou les soins de santé, leur adaptation aux subtilités de l'analyse des auditions des est limitée. Les recherches et projets existants dans des domaines adjacents ne parviennent pas à répondre aux exigences uniques de l'analyse des auditions dans les contextes d'enquête, laissant un écart significatif dans les capacités requises pour des pratiques d'investigation complètes et efficaces.

2.8 Conclusion

Ce chapitre a présenté un état de l'art détaillé sur le TALN, couvrant des aspects principaux tels que la représentation du texte, les modèles de réseaux de neurones pour le TALN, et les diverses tâches associées. Nous avons exploré les représentations textuelles discrètes et continues, et analysé l'évolution des architectures de réseaux de neurones, notamment les réseaux récurrents, LSTM, transformateurs, BERT, et les transformateurs de phrases. De plus, nous avons abordé les principales tâches du TALN, y compris la reconnaissance d'entités nommées, l'extraction de relations, et la détection de contradictions. Enfin, nous avons discuté des systèmes de recommandation et des travaux connexes qui leurs sont associés. Ce cadre théorique et technologique nous permet d'avancer vers la proposition de notre solution, qui sera développée dans le prochain chapitre.

Chapitre 3

Conception

3.1 Introduction

Dans ce chapitre, nous détaillerons la conception de notre système. Nous commencerons par présenter les diagrammes de cas d'utilisation et de classes pour donner une vue d'ensemble. Ensuite, nous décrirons notre approche de détection des contradictions et de recommandation des questions.

3.2 Diagramme de cas d'utilisation

Un acteur est une personne, un matériel ou un logiciel qui interagit avec le système. Notre système interagit avec deux types d'acteurs : principal et secondaire.

L'acteur principal de notre système est l'agent. Il est responsable du processus d'audition. Il permet d'ouvrir une nouvelle audition, de saisir les informations liées à l'affaire traitée et à la personne auditionnée, et d'enregistrer fidèlement toutes les questions posées ainsi que les réponses données par la personne auditionnée. Il permet aussi de demander au système une recommandation de questions via un modèle intelligent, ainsi que de vérifier s'il y a d'éventuelles contradictions dans les déclarations de la personne auditionnée, soit dans la même affaire, soit en exploitant les résultats de plusieurs affaires.

La figure 3.1 illustre le diagramme de cas d'utilisation pour l'acteur "Agent". L'acteur secondaire est l'administrateur, qui s'occupe de l'administration du système. Il gère les droits d'accès des agents et les fichiers des affaires et des auditions.

La figure 3.2 illustre le diagramme de cas d'utilisation pour l'acteur "Administrateur".

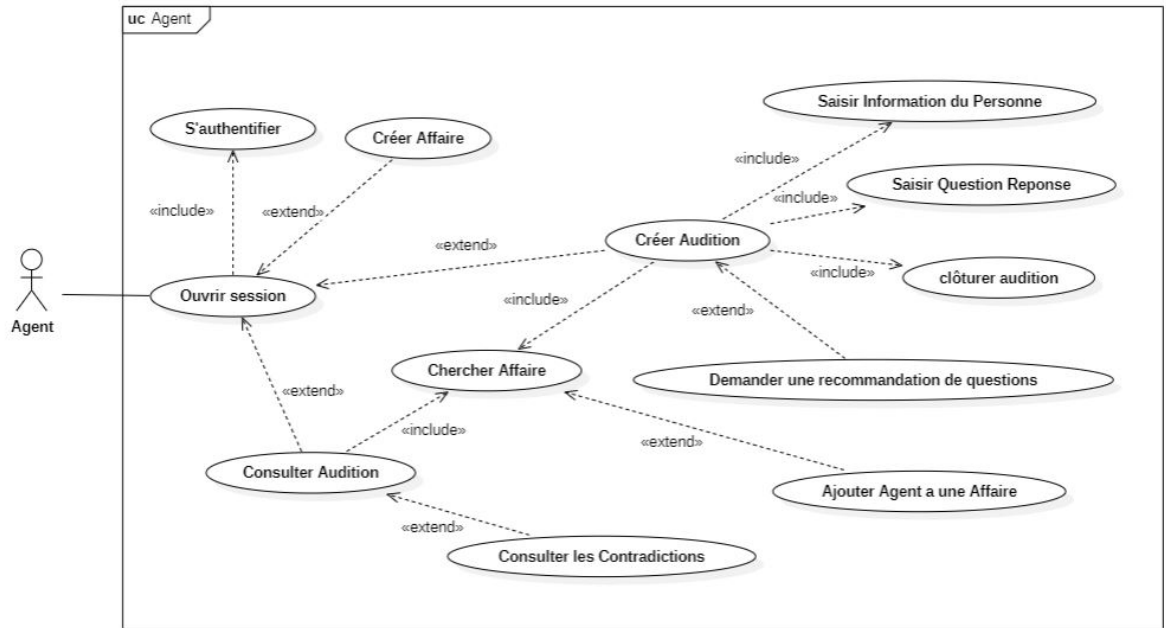


FIGURE 3.1 – Diagramme de cas d'utilisation de l'acteur « Agent »

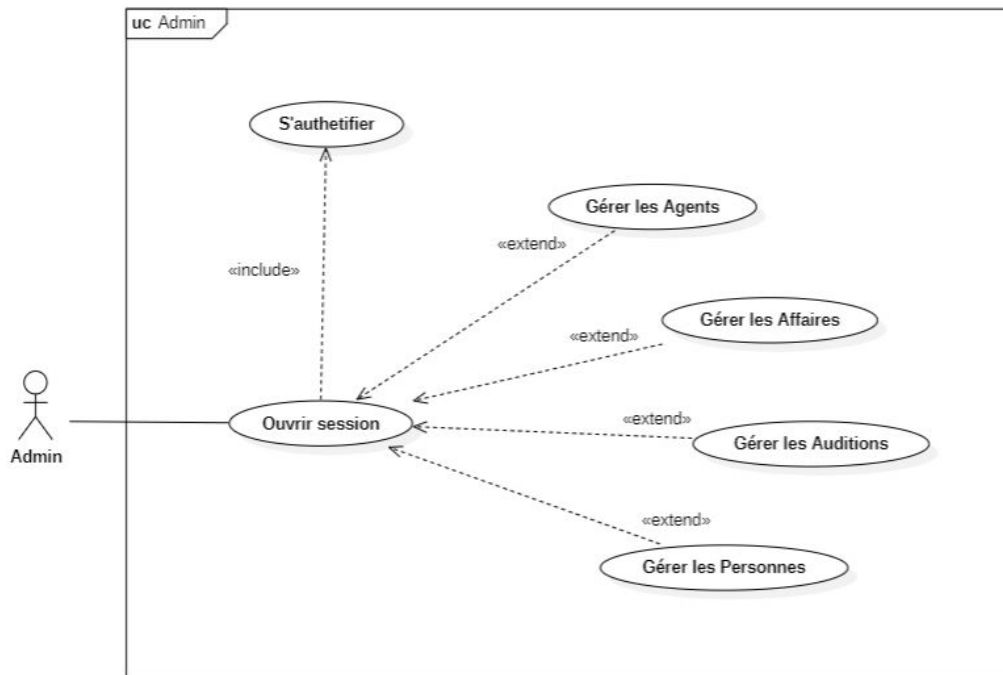


FIGURE 3.2 – Diagramme de cas d'utilisation de l'acteur « Administrateur »

3.3 Diagramme de classes

Dans cette section, nous décrirons les classes utilisées par notre système, où on se limite à ces classes : Agent (l'ensemble des agents responsables de l'audition), Affaire (les affaires judiciaires objets d'audition de plusieurs personnes soit victime,

témoin ou suspect), audition (l'ensemble d'audition qui est composée de plusieurs paires de question), Lieux (l'ensemble de lieux déclarés lors de l'audition. La figure 3.3, illustre le diagramme de classes proposé pour notre système.

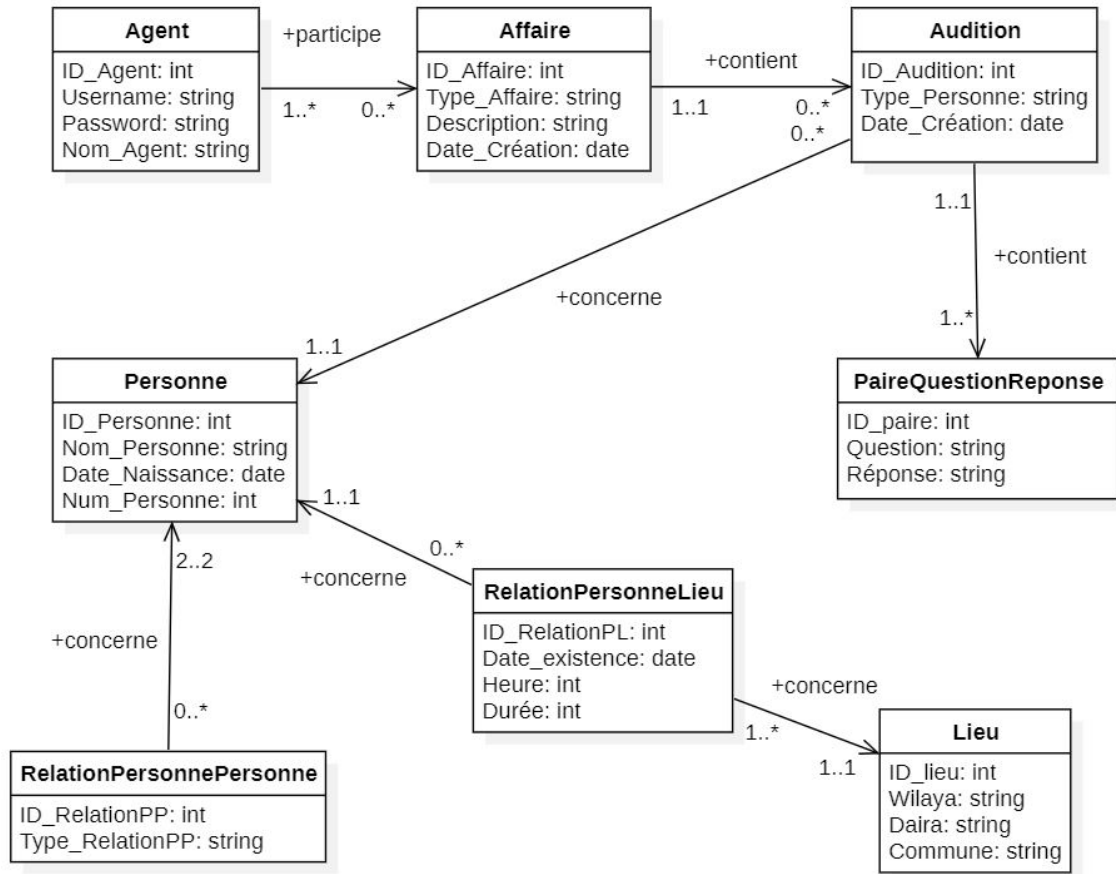


FIGURE 3.3 – Diagramme de Classe

3.4 Approche de détection de la contradiction

3.4.1 Introduction

La détection des contradictions dans les déclarations est une étape cruciale pour assurer la fiabilité et la cohérence des informations recueillies durant les auditions.

Cette section décrit la conception de notre approche de détection des contradictions, en se concentrant sur les relations entre les personnes et les relations entre une personne et un lieu.

3.4.2 Description de l'approche

Dans notre approche, nous nous concentrons spécifiquement sur la détection des contradictions dans les relations entre les personnes et d'autres entités. Cette décision est justifiée par les besoins de notre organisme d'accueil, qui considère les

personnes comme étant au centre de l'intérêt dans de telles enquêtes. Par conséquent, les contradictions n'impliquant pas les personnes sont souvent moins pertinentes ou significatives dans ce contexte. Ainsi, nous ne visons pas à résoudre la détection des contradictions de manière générale, mais nous nous concentrons plutôt sur les relations impliquant des individus.

Pour extraire et analyser efficacement ces relations, nous fournissons aux agents des questions prédéfinies conçues pour obtenir des réponses structurées. Ces questions sont élaborées pour recueillir systématiquement des informations sur les interactions, les associations et les événements impliquant les sujets des auditions. En utilisant cette méthode, nous veillons à ce que les données collectées soient à la fois complètes et cohérentes, facilitant ainsi une détection des contradictions plus précise, centrée sur les éléments essentiels de l'enquête.

3.4.3 Relations entre personnes

Pour les relations entre personnes, nous avons mis en place un ensemble de questions prédéfinies que l'agent peut utiliser. Par exemple, une question pourrait être "Quelle est votre relation avec...?" L'agent peut ensuite entrer le nom de la personne concernée pour compléter la question. La réponse de la personne est prise telle quelle et classifiée en fonction d'un ensemble de relations prédéfinies telles que "ami", "collègue", "parent", etc.

Nous utilisons des embeddings de mots pour la classification des réponses. Les embeddings de mots permettent de représenter les mots dans un espace vectoriel continu, facilitant ainsi la comparaison et la classification des réponses basées sur leur similarité sémantique.

Processus :

1. L'agent pose une question prédéfinie concernant la relation entre deux personnes.
2. La personne fournit une réponse.
3. La réponse est transformée en vecteur à l'aide des embeddings de mots.
4. Le vecteur est comparé aux vecteurs de relations prédéfinies pour classifier la réponse.
5. La relation identifiée est stockée dans le système.

3.4.4 Relations entre une personne et un lieu

Pour les relations entre une personne et un lieu, nous avons également des questions prédéfinies comme "Où étais-tu le ...?". L'agent peut spécifier l'année, le mois, le jour, l'heure et la durée. La réponse de la personne est formatée selon des catégories de localisation telles que l'état, la ville, etc.

Processus :

1. L'agent demande à la personne où elle se trouvait à une date et une heure précises ainsi que la durée.
2. La personne répond en précisant le lieu.
3. La relation entre la personne, le lieu, la date, l'heure et la durée est ensuite stockée dans le système.

3.4.5 Détection de Contradiction

Une fois les relations stockées, chaque nouvelle paire question-réponse est analysée pour extraire la relation qu'elle contient. Cette relation est ensuite comparée aux relations existantes dans le système pour détecter d'éventuelles contradictions.

Processus de Détection :

- Extraction de la relation à partir de la nouvelle paire question-réponse.
- Comparaison de cette relation avec les relations existantes dans le système.
- Identification des contradictions si une nouvelle relation contredit une relation déjà enregistrée.
- Signalement des contradictions détectées pour une révision par l'agent.

3.4.6 Pourquoi notre approche ?

La détection des contradictions est une tâche complexe, surtout dans le contexte de données limitées et spécifiques comme les auditions en dialecte algérien. Dans cette section, nous expliquons les raisons qui nous ont conduits à adopter notre approche particulière pour la détection des contradictions.

— Données en Arabe et en Dialecte Algérien

Le choix de notre approche est largement influencé par la nature des données disponibles. Les données en arabe, et plus particulièrement en dialecte algérien, sont rares. La plupart des techniques du TALN sont développées et optimisées pour l'anglais ou d'autres langues avec une abondance de données annotées [35]. En raison de cette rareté, il est difficile d'appliquer des méthodes sophistiquées de traitement des relations qui nécessitent de grandes quantités de données pour l'entraînement.

— Absence de Données d'audition

Une autre raison pour laquelle nous avons choisi notre approche est l'absence de données d'audition existantes. La plupart des approches de détection des relations reposent sur des corpus larges et annotés, tels que Wikipedia, qui ne sont pas disponibles dans notre contexte [35]. Notre méthode permet de contourner ce problème en utilisant des questions et réponses prédéfinies pour contrôler et classer les relations.

— **Précision des Informations**

Nous avons besoin de stocker des informations très précises, ce que les autres méthodes ne permettent pas toujours. Les approches générales de détection de relations peuvent manquer de granularité et ne pas capturer les nuances spécifiques de chaque relation. En utilisant des questions et réponses prédéfinies, nous pouvons garantir une classification précise et cohérente des relations.

— **Nature des Données**

Les autres méthodes de détection de relations sont souvent conçues pour des données structurées comme celles de Wikipedia, qui décrivent des faits de manière descriptive et à la troisième personne. En revanche, nos données consistent en des questions et réponses interrogatives, ce qui change la nature du texte et pourrait réduire l'efficacité des méthodes traditionnelles. Notre approche est spécialement conçue pour ce type de données, assurant une meilleure performance.

— **Collecte de Données Initiale**

Enfin, l'objectif principal de notre système est de collecter des données initiales. Une fois que nous aurons suffisamment de données, nous pourrions envisager d'autres approches de détection de relations. En attendant, notre méthode permet de construire une base de données solide et fiable, essentielle pour toute analyse future.

Notre approche de la détection des contradictions est adaptée à notre contexte spécifique. En utilisant des questions et réponses prédéfinies, nous pouvons garantir la qualité et la fiabilité des relations extraites, tout en posant les bases pour l'application future de méthodes plus avancées lorsque suffisamment de données seront disponibles.

3.5 Approche de recommandation des questions

3.5.1 Introduction

La recommandation des questions vise à assister les agents de la force publique pendant les auditions en générant des questions pertinentes et contextuelles. Ce module est crucial pour maintenir le flux de l'audition, surtout lorsque l'agent ne sait plus quelles questions poser pour obtenir des informations complètes et cohérentes de la part de la personne.

3.5.2 Description de l'approche

Pour la recommandation de questions, nous adoptons une approche en deux étapes : le filtrage basé sur le contenu et la similarité de phrases.

1. Filtrage basé sur le contenu

Nous sélectionnons les auditions ayant des caractéristiques communes avec l'audition en cours. Les caractéristiques utilisées pour le filtrage incluent le **type d'affaire** (type d'enquête) et le **type de personne** (victime, suspect, témoin).

2. Similarité de phrases

Pour chaque audition résultante du filtrage, nous comparons les paires question-réponse avec la dernière paire de l'audition en cours en utilisant un modèle de transformateur de phrases. Cela nous permet d'identifier les paires les plus similaires, et donc de recommander les questions suivantes à poser.

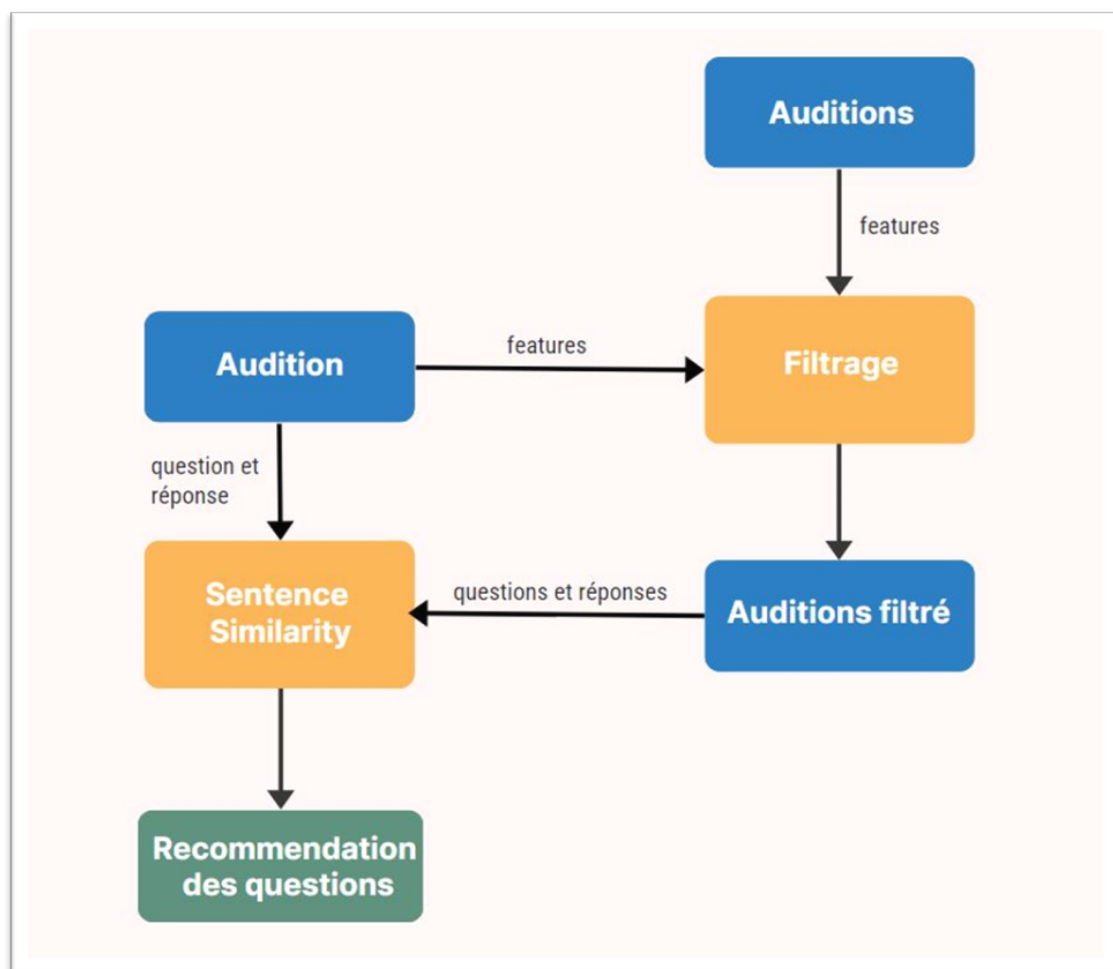


FIGURE 3.4 – Schéma global de la recommandation des questions

3.5.3 Filtrage

Le filtrage est la première étape cruciale pour réduire l'ensemble des auditions potentielles à un sous-ensemble plus gérable et pertinent. Nous utilisons les caractéristiques suivantes pour ce filtrage :

— **Type d'affaire :**

Cette caractéristique permet de s'assurer que les auditions sélectionnées concernent des affaires de même nature, ce qui augmente la pertinence des questions recommandées.

— **Type de personne :**

Il est essentiel de filtrer selon le type de personne interrogée (victime, suspect, témoin) afin d'aligner le contexte des questions recommandées avec le rôle de la personne dans l'enquête.

Une fois le filtrage effectué, nous obtenons un ensemble d'auditions pertinentes qui serviront de base pour la phase suivante de similarité de phrases.

3.5.4 Similarité de phrases

Après avoir identifié les auditions pertinentes, nous procédons à une comparaison sémantique entre la dernière paire question-réponse de notre audition courante et les paires contenues dans ces auditions filtrées. Les paires sont ensuite triées par ordre décroissant de similarité. La paire la plus similaire parmi toutes est choisie pour la recommandation. Nous recommandons à l'utilisateur les questions de l'audition qui suivent cette paire la plus similaire. Cette approche permet de fournir des suggestions pertinentes et adaptées, améliorant ainsi l'efficacité du système de recommandation.

Pour obtenir le score de similarité, nous utilisons une échelle de 0 à 1, où 0 représente la dissimilarité maximale et 1 l'identité parfaite. Nous comparons chaque question et chaque réponse individuellement, puis multiplions les résultats pour obtenir un score global entre 0 et 1. Ce score reflète la similarité globale entre les paires question-réponse, permettant ainsi de sélectionner la paire la plus pertinente pour la recommandation. Cette méthode assure une évaluation fine et précise des similarités, améliorant la qualité des suggestions fournies à l'utilisateur.

Pour ce faire, nous utilisons un modèle de transformateur de phrases. Ce modèle permet de convertir les textes en vecteurs denses de manière à capturer les significations sémantiques de manière efficace.

Le processus se déroule en plusieurs étapes :

1. **Encodage des phrases :**

Chaque phrase (question et réponse) est encodée en un vecteur dense de dimensions fixes, représentant son contenu sémantique.

2. **Calcul de similarité :**

Les vecteurs de la dernière paire question-réponse de l'audition courante sont comparés avec les vecteurs des paires question-réponse des auditions filtrées en utilisant des mesures de similarité cosinus. La similarité retournée est un score entre 0 et 1, où 0 indique aucune relation sémantique et 1 indique des phrases identiques.

Le processus de similarité de phrases permet de recommander les questions qui sont les plus pertinentes et contextuellement appropriées pour poursuivre l’audition de manière efficace.

3.5.5 Pourquoi notre approche ?

— **Pertinence contextuelle :**

Notre méthode commence par un filtrage basé sur des caractéristiques telles que le type d’affaire et le type de personne (victime, suspect, témoin). Ce filtrage assure que seules les auditions pertinentes et contextuellement appropriées sont considérées pour la recommandation de questions. Cette approche garantit que les questions proposées sont non seulement pertinentes mais aussi adaptées à la situation spécifique de l’audition.

— **Élimination du bruit :**

En filtrant les données avant d’appliquer des techniques de similarité, nous réduisons la quantité de données à traiter et éliminons les informations non pertinentes, ce qui améliore la performance globale du système et la qualité des recommandations.

— **Approche sémantique :**

Utiliser la similarité de phrases permet de comparer les paires question-réponse sur une base sémantique plutôt que simplement syntaxique. Cela signifie que nous pouvons identifier des questions qui sont similaires en termes de sens, même si elles sont formulées différemment, ce qui est crucial pour capturer la diversité des expressions linguistiques dans les auditions.

Notre approche combine le filtrage basé sur le contenu et la similarité de phrases avec des transformateurs de phrases pour maximiser la pertinence et la précision des recommandations de questions dans les auditions policières.

3.6 Conclusion

Dans ce chapitre, nous avons détaillé la conception de notre système de détection des contradictions et de recommandation des questions pour les auditions. Nous avons présenté les diagrammes de cas d’utilisation et de classes, suivi de la description approfondie de l’approche proposée. En utilisant des questions prédéfinies et des modèles de TALN, nous avons mis en place une méthode robuste pour extraire, analyser et vérifier les informations collectées. Cette approche permet non seulement de garantir la fiabilité des données, mais aussi d’assister efficacement les agents pendant les auditions.

Le chapitre suivant portera sur la mise en œuvre pratique de ce système, illustrant l’application de ses concepts clés dans un contexte réel.

Chapitre 4

Implémentation

4.1 Introduction

Ce chapitre est consacré à la réalisation de notre système. A cet effet, nous allons exposer les différents aspects techniques : langages, environnement ainsi que les outils et les plates-formes de développement choisies. A la fin, nous présenterons les modules de notre solution, appuyés par des captures d'écrans, en vue d'illustrer ses différentes fonctionnalités offertes.

4.2 Présentation de l'environnement de développement

4.2.1 Interface

— React

React est une bibliothèque JavaScript open source, développée par Facebook, pour la création d'interfaces utilisateur interactives et réactives. Elle permet de construire des composants d'interface utilisateur réutilisables et de gérer efficacement l'état et le rendu de ces composants.

Dans notre projet, nous avons utilisé React v18.2.0 pour développer l'interface utilisateur de notre application. React nous a permis de créer une interface utilisateur dynamique et réactive, en facilitant la gestion des états et des interactions utilisateur. En utilisant des composants réutilisables, nous avons pu développer rapidement et maintenir facilement notre code d'interface.

— Vite

Vite est un outil de compilation de code et de développement rapide pour les projets d'interface, conçu pour offrir une expérience de développement moderne et performante. Il supporte les frameworks populaires comme React,

Vue, et Svelte, et se distingue par sa capacité à fournir un rechargement à chaud ultra-rapide et des temps de compilation de code réduits.

Dans notre projet, nous avons utilisé Vite v5.2.0 pour gérer le processus de développement et de Compiler le code de notre application React. Vite nous a permis de bénéficier d'un rechargement à chaud rapide, ce qui a considérablement amélioré notre productivité en offrant un retour immédiat sur les modifications de code. De plus, les temps de compilation de code optimisés ont réduit le temps nécessaire pour compiler et déployer notre application.

— NodeJS

Node.js est un environnement d'exécution JavaScript open source, construit sur le moteur V8 de Chrome. Il permet d'exécuter du code JavaScript en dehors d'un navigateur, ce qui est particulièrement utile pour le développement d'outils de build et de gestion de projets d'interface. Node.js est connu pour sa capacité à gérer des opérations d'E/S non bloquantes, et il est largement utilisé pour automatiser les tâches de développement d'interface.

Dans notre projet, nous avons utilisé Node.js v20.11.1 et npm v9.5.0 pour gérer notre environnement de développement d'interface. Node.js a servi de base pour exécuter des outils comme Vite et des gestionnaires de paquets tels que npm. Cela nous a permis de compiler et de packager notre code, de gérer les dépendances, et de lancer des serveurs de développement pour prévisualiser notre application en temps réel.

— Redux Toolkit

Redux Toolkit est une bibliothèque officielle pour la gestion de l'état dans les applications Redux, conçue pour simplifier l'écriture du code Redux et améliorer l'expérience des développeurs. Elle offre des abstractions et des utilitaires puissants pour réduire le boilerplate, faciliter la création des slices de l'état et gérer les effets secondaires.

Dans notre projet, nous avons utilisé Redux Toolkit v2.2.5 et react-redux v9.1.2 pour gérer l'état global de notre application React. Redux Toolkit nous a permis de structurer notre état de manière claire et maintenable, en définissant des slices pour différentes parties de notre application.

— RadixUI et TailwindCSS

RadixUI est une collection de composants d'interface utilisateur non stylisés et accessibles pour React, conçus pour offrir une base robuste et flexible à partir de laquelle construire des interfaces utilisateur personnalisées. TailwindCSS, quant à lui, est un framework CSS utilitaire qui permet de styliser des applications rapidement et efficacement en utilisant des classes prédéfinies.

Dans notre projet, nous avons utilisé Radix v1.0.1 pour fournir les composants de base de notre interface utilisateur React, en tirant parti de ses fonctionnalités d'accessibilité et de ses comportements par défaut. Nous avons ensuite utilisé TailwindCSS v3.4.3 pour styliser ces composants, en appliquant des classes utilitaires directement dans notre JSX. Cette combinaison nous a permis de créer une interface utilisateur cohérente et esthétique, tout en conservant une grande flexibilité dans le design.

— **React-admin**

React Admin est un framework open source pour la création d'interfaces d'administration dans les applications React. Il permet de développer rapidement des back-offices fonctionnels en fournissant des composants prêts à l'emploi pour les vues CRUD (Create, Read, Update, Delete), l'authentification, la gestion des permissions, et bien plus encore.

Dans notre projet, nous avons utilisé React Admin v4.16.18 pour construire l'interface d'administration de notre application.

4.2.2 Partie serveur

— **Python**

Pour développer la partie serveur, nous avons opté pour le langage de programmation Python v3.10 64-bit.

Python est un langage de programmation orienté objet, comparable à Perl, Ruby ou Java. Il est apprécié pour sa syntaxe simple, facilitant la lecture et l'écriture des programmes. Python est open source et peut être modifié et redistribué librement. Sa popularité a conduit à la création de nombreuses bibliothèques pour des tâches courantes telles que la connexion à des serveurs Web, la manipulation de textes, le traitement de données, et la gestion de fichiers. Notamment, il dispose de bibliothèques et frameworks d'apprentissage automatique largement utilisés.

— **Flask**

Flask est un micro-framework web en Python, léger et flexible, idéal pour le développement d'applications web et d'API. Il est basé sur Werkzeug et Jinja2, offrant une structure minimale mais extensible pour la création d'applications web. Flask permet aux développeurs de choisir les composants et les bibliothèques qu'ils souhaitent utiliser, favorisant ainsi une grande flexibilité et personnalisation.

Dans notre projet, nous avons utilisé Flask v3.0.3 et Flask-Cors v4.0.1 pour connecter la partie serveur à l'interface utilisateur. Flask a été essentiel pour gérer les requêtes HTTP, permettre la communication entre le serveur et l'interface, et fournir les données nécessaires à l'interface.

- **gensim**

Gensim est une bibliothèque open source en Python, spécialisée dans le TALN et l'apprentissage automatique.

Dans notre projet, nous avons principalement utilisé Gensim v4.3.2 pour exploiter un modèle Word2Vec pré-entraîné, afin de classer les relations entre différents termes. Gensim nous a permis de charger et de manipuler facilement le modèle Word2Vec, transformant les mots en vecteurs de haute dimension.

- **Sentence-Transformers**

Sentence-Transformers est une bibliothèque en Python qui permet de générer des représentations vectorielles pour des phrases ou des textes complets. Elle est basée sur les modèles de transformateurs, tels que BERT, RoBERTa et autres, et est spécialement optimisée pour calculer la similarité entre des phrases.

Dans notre projet, nous avons utilisé Sentence-Transformers v2.6.1 pour mettre en œuvre un modèle de similarité de phrases. Cela nous a permis de comparer des phrases en termes de similarité sémantique, en transformant les phrases en vecteurs denses et de haute dimension.

- **PostgreSQL**

PostgreSQL est un système de gestion de base de données relationnelle open source, reconnu pour sa robustesse, sa fiabilité et ses nombreuses fonctionnalités avancées. Il prend en charge les transactions ACID, l'intégrité référentielle, les vues, les déclencheurs, les procédures stockées, et offre une extensibilité via des plugins.

Dans notre projet, nous avons utilisé PostgreSQL v16.3 pour stocker et gérer les données nécessaires à notre application.

- **pgvector**

pgvector est une extension pour PostgreSQL qui ajoute la prise en charge des vecteurs, permettant ainsi de stocker et de manipuler des données vectorielles directement dans la base de données. Elle est particulièrement utile pour les applications nécessitant des recherches de similarité.

Dans notre projet, nous avons utilisé pgvector v0.7.2 pour stocker les représentations vectorielles générées par notre modèle de transformateur de phrase. Cette extension nous a permis d'effectuer des comparaisons et des recherches de similarité directement au niveau de la base de données, améliorant ainsi l'efficacité et la rapidité de nos opérations.

- **Psycopg2**

Psycopg2 est un adaptateur PostgreSQL pour le langage de programmation Python, permettant une interface fiable et performante pour interagir avec

des bases de données PostgreSQL. Il est conçu pour être entièrement compatible avec les spécifications DB-API 2.0 de Python et prend en charge les fonctionnalités avancées de PostgreSQL.

Dans notre projet, nous avons utilisé Psycopg2 v2.9.9 pour connecter notre application Python à notre base de données PostgreSQL. Psycopg2 nous a permis d'exécuter des requêtes SQL, de récupérer des données, et de gérer les transactions de manière efficace et sécurisée.

4.2.3 Logiciels

— Visual Studio Code

Visual Studio Code (VSCode) est un éditeur de code source puissant et léger développé par Microsoft. Il est conçu pour supporter plusieurs langages de programmation et offre une large gamme de fonctionnalités adaptées aux développeurs. Parmi ses principales caractéristiques, on trouve la coloration syntaxique, l'auto-complétion intelligente, et un débogueur intégré.

— DBeaver

DBeaver est un outil open source de gestion de bases de données, compatible avec de nombreux systèmes de gestion de bases de données relationnelles et non relationnelles tels que PostgreSQL, MySQL, Oracle, SQLite, et bien d'autres. Il offre une interface utilisateur graphique intuitive pour administrer, développer et maintenir des bases de données.

Dans notre projet, nous avons utilisé DBeaver v24.1.0 pour administrer notre base de données PostgreSQL. DBeaver nous a permis de visualiser et de gérer les schémas, d'exécuter des requêtes SQL, de superviser les performances de la base de données et de manipuler les données de manière efficace.

— Postman

Postman est un outil de collaboration et de test pour les API, largement utilisé par les développeurs pour la conception, le test, et la documentation des API. Il offre une interface utilisateur conviviale qui permet de créer et d'envoyer des requêtes HTTP de manière simple et efficace, tout en visualisant les réponses reçues.

Dans le cadre de notre projet, nous avons utilisé Postman pour tester et valider nos endpoints API. Grâce à ses capacités de simulation de requêtes et de visualisation des réponses, nous avons pu identifier et corriger rapidement les erreurs dans notre partie serveur.

— Chrome DevTools

Chrome DevTools est un ensemble d'outils de développement intégrés au navigateur Google Chrome. Ces outils sont essentiels pour les développeurs web, leur permettant de déboguer, tester et optimiser leurs applications web directement dans le navigateur.

Dans le cadre de notre projet, nous avons utilisé Chrome DevTools pour déboguer et optimiser notre interface utilisateur. Grâce à l'inspection du DOM et aux outils de console comme `redux devtools`, nous avons pu identifier et corriger rapidement les erreurs de rendu et les problèmes de script. Les outils de réseau nous ont permis d'analyser les requêtes API et de garantir des temps de réponse optimaux. Enfin, les outils de performance et de profilage nous ont aidés à améliorer la vitesse de chargement et l'efficacité de notre application, garantissant une expérience utilisateur fluide et réactive.

4.3 Les modèles utilisés

4.3.1 Word2vec CBOW

Pour la détection des relations personne personne, nous avons utilisé un modèle Word2Vec CBOW (Continuous Bag of Words) provenant d'un projet appelé AraVec, qui a développé plusieurs modèles de word embeddings pour l'arabe [53]. Le modèle que nous avons choisi est un modèle unigram de 300 dimensions, construit en utilisant des corpus de données provenant de Twitter.

La sélection de 300 dimensions pour notre modèle Word2Vec se justifie par plusieurs raisons :

- **Équilibre entre Performance et Efficacité**

Un modèle de 300 dimensions offre un bon compromis entre la capacité à capturer les nuances sémantiques et la complexité computationnelle. Des dimensions plus élevées pourraient capturer davantage de détails sémantiques, mais au coût d'une augmentation significative des ressources de calcul et de la mémoire nécessaire.

- **Standard de l'industrie**

De nombreux travaux de recherche et applications pratiques utilisent des vecteurs de 300 dimensions, ce qui en fait un choix éprouvé pour de nombreux cas d'utilisation dans le TALN [54].

L'utilisation de corpus de données provenant de Twitter pour entraîner le modèle présente plusieurs avantages spécifiques à notre projet :

- **Langage Naturel et Informel**

Les données de Twitter sont caractérisées par un langage naturel et souvent informel, avec de nombreuses variations linguistiques, abréviations et

expressions idiomatiques. Cela permet au modèle d'apprendre des représentations riches et diversifiées qui sont utiles pour traiter des textes provenant des données d'audition.

— **Richesse du Corpus**

Twitter génère un volume massif de données textuelles, offrant une large couverture lexicale et contextuelle. Entraîner le modèle sur ces données permet de capturer une grande variété de contextes et de relations sémantiques.

— **Actualité et Variabilité**

Les tweets reflètent souvent des événements actuels, des tendances et des discussions populaires, ce qui rend les embeddings particulièrement adaptés pour des applications nécessitant une compréhension des contextes modernes et variés.

4.3.2 Transformateur de phrase

Pour la similarité des phrases, nous avons utilisé un modèle de transformateur de phrases appelé all-MiniLM-L6-v2 [29].

Nous avons choisi d'utiliser ce modèle car il offre un excellent compromis entre performance et efficacité. Le all-MiniLM-L6-v2 est un modèle compact et rapide qui permet de calculer les similarités entre phrases avec une grande précision, tout en réduisant les besoins en ressources computationnelles. Son architecture légère le rend particulièrement adapté pour des applications nécessitant une évaluation rapide des similarités de texte dans des environnements limités en ressources comme notre.

Pour mesurer la similarité entre deux vecteurs, nous avons utilisé la similarité cosinus. La similarité cosinus est une mesure de la similarité entre deux vecteurs dans un espace vectoriel, qui est calculée en prenant le produit scalaire des deux vecteurs et en le divisant par le produit des normes des deux vecteurs. La formule de la similarité cosinus pour deux vecteurs \vec{A} et \vec{B} est la suivante :

$$\text{similarité cosinus} = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|}$$

4.4 Présentation de l'application

Dans cette partie, nous présentons un ensemble d'interface qui représentent notre application.

- **Vue de l'agent**

L'interface de connexion est présentée comme le montre la figure 4.1.




FIGURE 4.1 – Interface d'authentification pour l'agent

Après s'être authentifié, l'agent accède à la page d'accueil des affaires où il peut voir toutes les affaires disponibles pour lui. Comme illustré dans la figure suivante, il peut choisir de créer une nouvelle affaire, de consulter les affaires déjà créées, ou de se déconnecter.

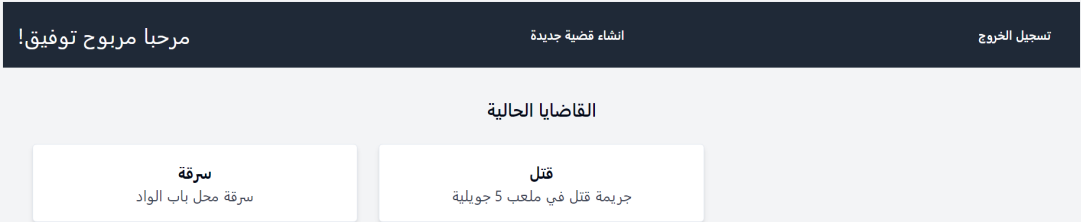


FIGURE 4.2 – Interface d'accueil pour l'agent

L'agent peut créer de nouvelles affaires en entrant les informations de l'affaire (type d'affaire, description de l'affaire) et en cliquant sur "Créer une nouvelle affaire", comme illustré dans la figure 4.3.

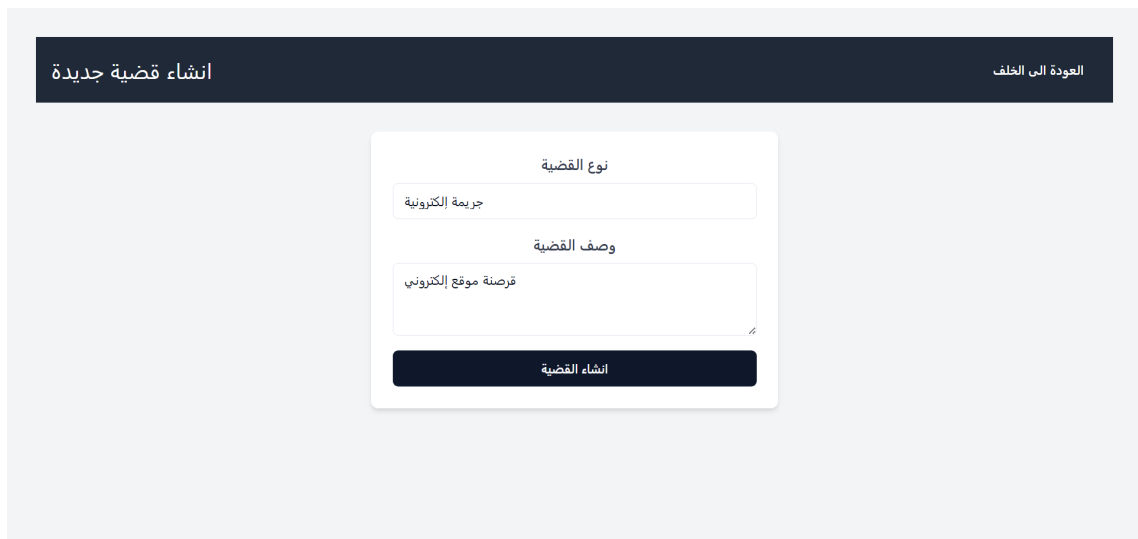


FIGURE 4.3 – Interface pour la création des affaires

Après avoir choisi une affaire, l'agent est transféré à la page d'accueil de l'affaire où il peut choisir de voir les anciennes auditions de l'affaire, d'ajouter d'autres agents à l'affaire, de démarrer une nouvelle audition, ou de revenir simplement à la page d'accueil des affaires, comme montré dans la figure suivante.

قضية قتل : جريمة قتل في ملعب 5 جويلية

[العودة إلى القضايا](#) [بدء جلسة جديدة](#) [رؤية الجلسات السابقة](#) [إضافة ضباط آخرين إلى القضية](#)

FIGURE 4.4 – Interface d'accueil d'une affaire

Un agent peut ajouter d'autres agents à l'affaire en entrant leur nom d'utilisateur et en cliquant sur "Ajouter un agent" comme le montre la figure 4.5.

The screenshot shows a web interface with a dark blue header. On the left, it says 'إضافة ضباط اخرين الى القضية' (Add other officers to the case). On the right, it says 'العودة الى الخلف' (Go back). Below the header is a white box containing the title 'اسم المستخدم' (Username). Underneath is a text input field with the placeholder 'رجاء ادخال اسم مستخدم الضابط' (Please enter the officer's username). At the bottom of the box is a dark blue button labeled 'إضافة الضابط' (Add officer).

FIGURE 4.5 – Interface pour l'ajoute des agents a l'affaire

Si l'agent choisit de "Voir les anciennes auditions", l'agent accède à une page qui montre toutes les auditions réalisées auparavant concernant l'affaire sélectionnée.

The screenshot shows a web interface with a dark blue header. On the left, it says 'مرحبا مبروح توفيق!' (Hello Mubrah Tawfiq!). On the right, it says 'تسجيل الخروج' (Logout). Below the header is the title 'الجلسات السابقة' (Previous sessions). Underneath is a white box containing the title 'شاهد' (View) and the name 'بلماضي شعيب' (Bilmaضي Shaiib).

FIGURE 4.6 – Interface des Auditions déjà fait pour l'affaire

Sinon si il choisit de "Créer une nouvelle audition", l'agent reçoit une page contenant un formulaire où il saisit les informations de l'audition : type de personne (suspect, témoin, victime), nom complet, date de naissance et numéro. Il doit ensuite cliquer sur "Confirmer" pour ouvrir une nouvelle audition.

FIGURE 4.7 – Interface pour la création d'une nouvelle audition

Après avoir confirmé qu'il veut créer une nouvelle audition, l'agent est dirigé vers la page de l'audition, où il peut saisir toutes les questions et réponses de l'audition.

FIGURE 4.8 – Interface de saisie des questions et réponses

- **Saisie des questions et réponses**

Un agent peut saisir des questions non prédéfinies et leurs réponses comme le montre la figure 4.9.

L'agent peut aussi utiliser une question prédéfinie comme montré dans la figure 4.10. Pour les relations de type personne-personne, l'agent doit seulement saisir le nom de la personne concerné et la réponse de la personne auditionnée.

Il peut aussi sélectionner une question prédéfinie pour la relation personne-lieu. Il peut ensuite saisir la date exacte, l'heure, la période, ainsi que la réponse de la personne en termes de wilaya, daïra, commune et adresse, comme montré dans la figure 4.11.

The screenshot shows a web application interface with a dark blue header. On the left, there are two links: 'إلغاء الجلسة' (End Session) and 'إغلاق الجلسة' (Close Session). On the right, the user's profile is displayed: 'مشتبه به' (Suspect), 'الاسم : زرقاوي خيثر' (Name: Zarkaoui Khithar), 'تاريخ الميلاد : 15-10-1970' (Date of Birth: 15-10-1970), and 'الرقم : 25849462' (Number: 25849462). Below the header, there is a large white text area for the question. To the right of this area is a button labeled 'طلب اسئلة مقترحة' (Request suggested questions). At the bottom, there are two input fields: the first is labeled 'ما اسمك الكامل' (What is your full name) and contains the text 'زرقاوي خيثر'; the second is labeled 'استخدم سؤال محدد مسبقاً' (Use a specific question in advance). Below these fields is a 'تأكيد' (Confirm) button.

FIGURE 4.9 – saisie des question non prédéfinie

The screenshot shows the same application interface as Figure 4.9. The header and user profile information are identical. In the main question area, the text 'سؤال : ما اسمك الكامل؟' (Question: What is your full name?) and 'جواب : زرقاوي خيثر' (Answer: Zarkaoui Khithar) is displayed. To the right of this area is a button labeled 'طلب اسئلة مقترحة' (Request suggested questions). At the bottom, there are two input fields: the first is labeled 'زرقاوي سمير' (Zarkaoui Samir) and contains the text 'زرقاوي سمير'; the second is labeled 'ما علاقتك مع ___' (What is your relationship with ___) and contains a dropdown arrow. Below these fields is a 'تأكيد' (Confirm) button.

FIGURE 4.10 – saisie des questions prédéfinie, cas relation personne personne

The screenshot shows a web application interface. At the top, there is a dark blue header with the text 'إغلاق الجلسة' (Close session) and 'إلغاء الجلسة' (Cancel session). On the right side of the header, there is a user profile section titled 'مشيتيه به' (My profile) with the following information: 'الاسم : زرقاوي خيثر' (Name: Zarkaoui Khithar), 'تاريخ الميلاد : 1970-10-15' (Date of birth: 1970-10-15), and 'الرقم : 25849462' (Number: 25849462). Below the header, there is a main content area with two question boxes. The first box contains the text 'سؤال : ما اسمك الكامل؟' (Question: What is your full name?) and 'جواب : زرقاوي خيثر' (Answer: Zarkaoui Khithar). The second box contains the text 'سؤال : ما هي علاقتك مع زرقاوي سمير؟' (Question: What is your relationship with Zarkaoui Samir?) and 'جواب : صديقي' (Answer: My friend). To the right of these boxes, there is a sidebar with a button labeled 'طلب اسئلة مقترحة' (Request suggested questions). At the bottom of the interface, there is a dark blue footer with a form containing several input fields: '2024', '5', '20', '16', '8', 'أين كنت يوم' (Where were you on the day), 'استخدم سؤال خاص' (Use a specific question), 'الجزائر' (Algeria), 'باب الواد' (Bab el Ward), 'البلدية' (Municipality), 'العنوان' (Address), and a 'تأكيد' (Confirm) button.

FIGURE 4.11 – saisie des questions prédéfinie, cas relation personne lieu

• recommandation de questions

À tout moment, un agent peut demander une recommandation de question en cliquant sur le bouton "Demander une recommandation de question".

The screenshot shows the same web application interface as Figure 4.11, but with the 'Demande de question recommandée' button clicked. The main content area now displays four recommended questions in a list. The first question is 'سؤال : ما اسمك الكامل؟' (Question: What is your full name?) with the answer 'جواب : زرقاوي خيثر' (Answer: Zarkaoui Khithar). The second question is 'سؤال : ما هي علاقتك مع زرقاوي سمير؟' (Question: What is your relationship with Zarkaoui Samir?) with the answer 'جواب : صديقي' (Answer: My friend). The third question is 'سؤال : أين كنت يوم 8-16-20-5-2024؟' (Question: Where were you on 8-16-20-5-2024?) with the answer 'جواب : الجزائر-باب الواد' (Answer: Algeria-Bab el Ward). The fourth question is 'سؤال : ملك من هذا السكن؟' (Question: Who owns this house?) with the answer 'جواب : ملكي' (Answer: My own). The sidebar still contains the 'طلب اسئلة مقترحة' button. The footer form is also present, with the 'تأكيد' button.

FIGURE 4.12 – Fin de saisie, avant recommandation de questions

La figure suivante montre le résultat après que l'agent a cliqué sur le bouton de recommandation. Des questions recommandées apparaissent.

The screenshot shows a web application interface. At the top, there is a dark blue header with the text 'إغلاق الجلسة' (Close Session) and 'إلغاء الجلسة' (Cancel Session). On the right side of the header, there is a user profile section titled 'مشيتيه به' (My Profile) with the following details: الاسم : زرقاوي خيثر (Name: Zarkaoui Khithar), تاريخ الميلاد : 1970-10-15 (Date of Birth: 1970-10-15), and الرقم : 25849462 (Number: 25849462).

The main content area contains a list of four questions, each with a 'سؤال : ما اسمك الكامل؟' (Question: What is your full name?) and a 'جواب : زرقاوي خيثر' (Answer: Zarkaoui Khithar) field. The questions are:

- سؤال : ما اسمك الكامل؟
جواب : زرقاوي خيثر
- سؤال : ما هي علاقتك مع زرقاوي سمير؟
جواب : صديقي
- سؤال : اين كنت يوم 8-16-20-5-2024؟
جواب : الجزائر-باب الواد
- سؤال : ملك من هذا السكن؟
جواب : ملكي

On the right side, there is a sidebar with a button 'طلب اسئلة مقترحة' (Request Suggested Questions) and a section titled 'الأسئلة المقترحة:' (Suggested Questions:). Below this, there are two input fields: 'هل يمكنك تفسير لماذا وجد في مسرح الجريمة؟' (Can you explain why you were found in the crime scene?) and 'من الذي أحضره؟' (Who brought him?).

At the bottom, there is a dark blue footer with a 'تأكيد' (Confirm) button and two input fields: 'اكتب سؤالك' (Write your question) and 'اكتب الاجابة' (Write the answer).

FIGURE 4.13 – recommandation de questions

• détection des contradictions

Lorsque l'agent souhaite clôturer l'audition, il peut le faire en cliquant sur le bouton "Clôturer l'audition". L'audition est alors clôturée et le processus de détection des contradictions commence. La figure suivante montre qu'aucune contradiction n'a été détectée.

The screenshot shows the same web application interface as Figure 4.13, but with a different sidebar. The sidebar now displays the message 'لا توجد تناقضات' (No contradictions found) in a light blue box. The rest of the interface, including the header, the list of questions, and the footer, remains the same.

FIGURE 4.14 – Après Clôture de l'audition

Pour les besoins de l'exemple, supposons qu'une nouvelle audition a été réalisée après la précédente avec la même personne. Cette audition contient deux contradictions par rapport à l'audition précédente, chaque contradiction étant pour un type de relation différent.

إغلاق الجلسة

إلغاء الجلسة

مشتبه به
الاسم : زرقاوي خيثر
تاريخ الميلاد : 1970-10-15
الرقم : 25849462

طلب اسئلة مقترحة

سؤال : ما اسمك الكامل؟
جواب : زرقاوي سمير

سؤال : ما هي علاقتك مع زرقاوي خيثر؟
جواب : ولدي

سؤال : اين كنت يوم 8-16-20-2024؟
جواب : الجزائر-بوزريعة

اتعرف من قتله

استخدم سؤال محدد مسبقاً

لا

تأكيد

FIGURE 4.15 – Saisie d'une deuxième audition

Après que l'agent a clôturé l'audition, des contradictions ont été détectées comme montré dans la figure suivante :

العودة الى الجلسات

جلسة مغلقة

مشتبه به
الاسم : زرقاوي خيثر
تاريخ الميلاد : 1970/10/15
الرقم : 25849462

التناقضات التي وجدت

تناقض بين علاقة زرقاوي خيثر مع زرقاوي سمير ابني ضد صديقي

تناقض بين مكان الوجود يوم 5-2024-8-16-20 الجزائر-باب الواد ضد الجزائر-بوزريعة

سؤال : ما اسمك الكامل؟
جواب : زرقاوي سمير

سؤال : ما هي علاقتك مع زرقاوي خيثر؟
جواب : ولدي

سؤال : اين كنت يوم 8-16-20-2024؟
جواب : الجزائر-بوزريعة

سؤال : اتعرف من قتله؟
جواب : لا

FIGURE 4.16 – Après Clôture de la deuxième audition, détection de contradiction

- **Vue de l'administrateur**

Interface d'authentification pour l'administrateur est présentée dans la figure suivante :

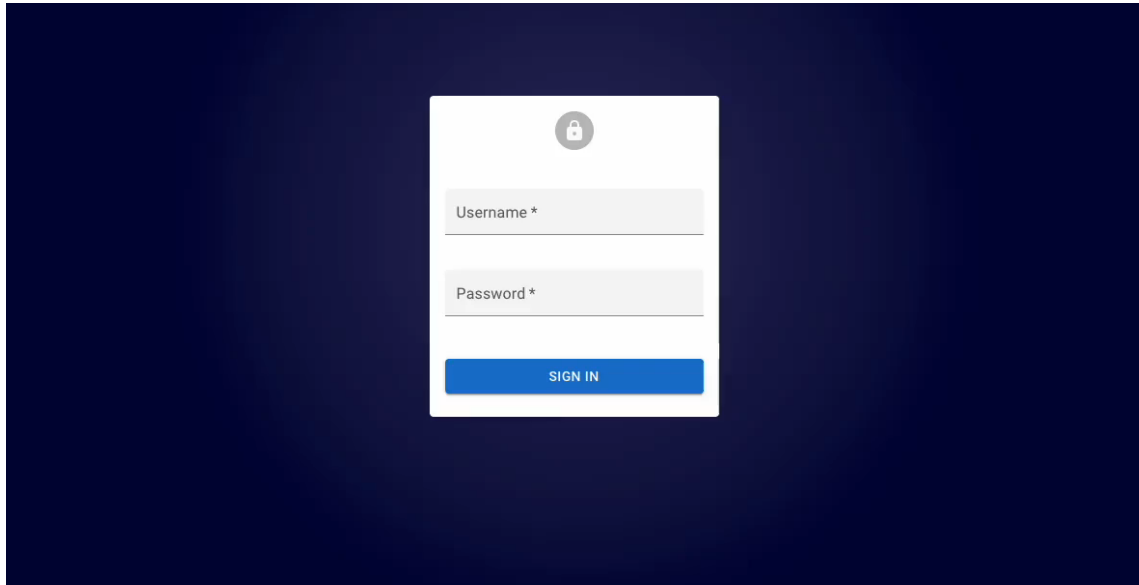


FIGURE 4.17 – Interface d'authentification pour l'admin

Après s'être authentifié, l'administrateur accède à la page d'accueil où il peut gérer tous les entités du système :

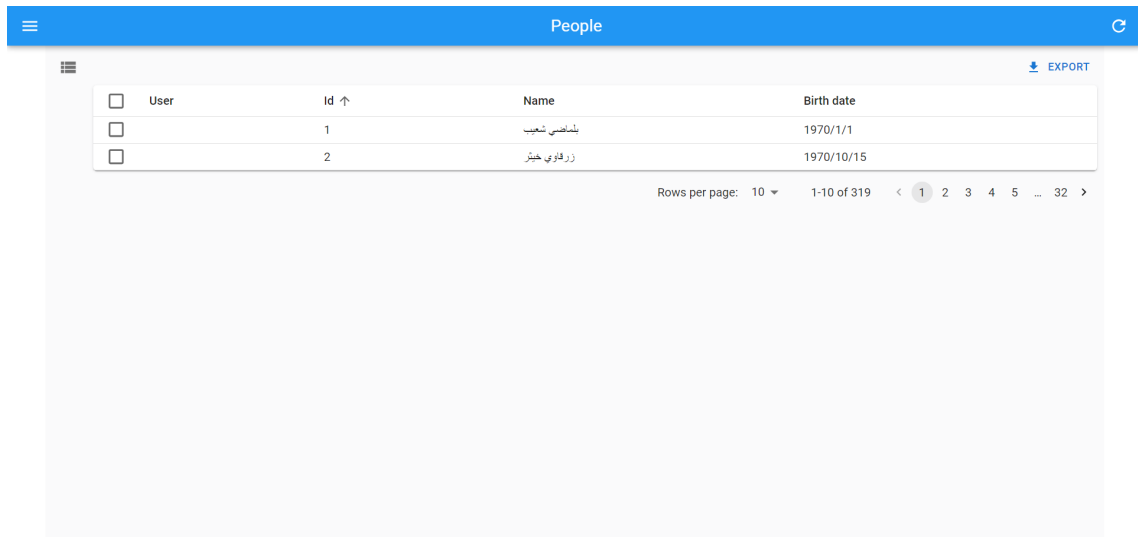


FIGURE 4.18 – Interface de la gestion pour l'admin

4.5 Conclusion

Dans ce chapitre, nous avons détaillé le processus d'implémentation de notre système, en décrivant les différents outils et technologies utilisés. Nous avons développé l'interface utilisateur avec React, optimisé notre flux de travail via Vite, et géré l'état global de l'application à l'aide de Redux Toolkit. La partie serveur a été réalisée en utilisant Python et Flask, avec le support de PostgreSQL pour la gestion des données. Des modèles avancés du TALN, tels que Word2Vec et le transformateur de phrase, ont été intégrés pour améliorer les fonctionnalités de notre application. Nous avons également présenté l'interface utilisateur et décrit ses principales fonctionnalités à travers une série de captures d'écran. Cette implémentation a permis de mettre en oeuvre un système robuste et performant, capable de répondre aux besoins de l'organisme d'accueil.

Conclusion générale & travaux futurs

Dans ce projet, nous avons travaillé avec une institution gouvernementale à savoir le Commandement de la Gendarmerie Nationale, où nous avons développé un système intelligent d'aide à l'audition, destiné à améliorer le processus d'enquête. Notre système s'appuie sur les technologies d'IA et de TALN pour offrir des fonctionnalités avancées telles que la détection des contradictions et la recommandation de questions. Ces fonctionnalités visent à améliorer la fiabilité et la précision des enquêtes tout en réduisant la charge cognitive des enquêteurs.

Contributions Clés

Les principales contributions de notre projet peuvent être énumérées ainsi :

- **Réalisation d'un Système d'audition Intégré**

Nous avons conçu et mis en œuvre un système d'audition intégré qui permet aux agents de créer, de gérer et de conduire des auditions de manière efficace et structurée. Ce système comprend une interface utilisateur intuitive pour la saisie des questions et réponses, la recommandation de questions, et la détection des contradictions. Il intègre également des fonctionnalités de gestion des affaires et des auditions, améliorant ainsi la fluidité du processus d'enquête et assurant une meilleure organisation et suivi des informations collectées.

- **Détection des contradictions**

Nous avons développé un module de détection des contradictions qui analyse les relations entre personnes et lieux en utilisant des questions prédéfinies. Cette approche assure une classification précise et cohérente des relations, même dans des contextes de données limitées en arabe et en dialecte algérien.

- **Recommandation de Questions**

Nous avons proposé un module de recommandation de questions basé sur une approche en deux étapes combinant le filtrage basé sur le contenu et la similarité de phrases. Cette méthode garantit que les questions proposées

sont pertinentes et contextuelles, ce qui améliore la fluidité et l'efficacité des auditions.

Travaux Futurs

En continuité de nos travaux, les perspectives suivantes sont envisagées :

- **Amélioration de la Détection des Contradictions**

Puisque notre approche exige un effort considérable et un investissement en temps opportun, il serait bénéfique d'explorer des techniques d'apprentissage automatique avancées, telles que les réseaux de neurones profonds, en particulier ceux formés sur de grandes quantités de données annotées. La collecte de données d'audition en arabe et en dialecte algérien sera cruciale pour ce développement.

- **Amélioration de la recommandation de questions**

Augmenter la base de données des questions prédéfinies et des réponses pourrait améliorer la capacité du système à couvrir un éventail plus large de scénarios d'enquête. L'intégration de techniques de génération automatique de questions basées sur des modèles de langage avancés [55] et les techniques de génération augmentée par récupération (RAG pour *retrieval-augmented generation* en anglais) [56] pourrait également être explorée.

- **Validation et Tests sur le Terrain**

Une validation extensive du système sur le terrain permettra d'identifier les limitations pratiques et les aspects nécessitant des ajustements. Par exemple, le modèle utilisé pour la classification de la relation peut donner des résultats erronés, ce qui requiert son amélioration au fur et à mesure.

Les retours (feedback) des agents sur terrain seront précieux pour affiner et optimiser les fonctionnalités du système.

En conclusion, notre projet a démontré le potentiel des technologies d'IA et de TALN pour améliorer les pratiques d'enquête en automatisant des tâches complexes et en fournissant des analyses avancées. Les travaux futurs se basent sur l'amélioration de la robustesse et de la précision du système, tout en explorant de nouvelles pistes technologiques pour enrichir ses capacités et son efficacité dans le domaine des enquêtes judiciaires et administratives.

Bibliographie

- [1] Advancing policing through ai : Insights from the global law enforcement community. <https://www.police1.com/iacp/articles/advancing-policing-through-ai-insights-from-the-global-law-enforcement-community>
lien réduit : <http://bit.ly/4cSpfNs>.
- [2] Greg Friese. Ai in law enforcement : Predictive policing and crime analysis. <https://thideai.com/ai-in-law-enforcement-predictive-policing-and-crime-analysis>.
- [3] Jérémy Robert. Natural language processing (nlp) : Définition et principes. <https://datascientest.com/introduction-au-nlp-natural-language-processing>.
- [4] Yoav Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57 :345–420, 2016.
- [5] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Pearson, Feb. 2024.
- [6] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.
- [7] Ayush Thakur, Laxmi Ahuja, Rashmi Vashisth, and Rajbala Simon. Nlp ai speech recognition : An analytical review. In *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1390–1396, 2023.
- [8] Venkata Sai Rishita Middi, Middi Raju, and Tanvir Ahmed Harris. Machine translation using natural language processing. *MATEC Web of Conferences*, 277 :02004, 01 2019.
- [9] R. Patil, S. Boit, V. Gudivada, and J. Nandigam. A survey of text representation and embedding techniques in nlp. *IEEE Access*, 11 :36120–36146, 2023.
- [10] Glossaire du machine learning - one hot encoding. <https://developers.google.com/machine-learning/glossary?hl=fr#one-hot-encoding>.
- [11] Mustafa Abdul Salam. Sentiment analysis of product reviews using bag of words and bag of concepts. *International Journal of Electronics*, 11 :49–60, 2019.

- [12] Djoerd Hiemstra. A probabilistic justification for using tf-idf term weighting in information retrieval. *International Journal on Digital Libraries*, 3(2) :131–139, Aug. 2000.
- [13] Samir KECHID. Module recherche d’information (ri), chapitre 3 - pondération des termes, 2023-2024. Master 2, Systèmes Informatiques Intelligents.
- [14] K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, Computers*, 28(2) :203–208, June 1996.
- [15] Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*, Scottsdale, Arizona, USA, May 2-4 2013.
- [16] S.J. Johnson, M.R. Murty, and I. Navakanth. A detailed review on word embedding techniques with emphasis on word2vec. *Multimed Tools Appl*, 83 :37979–38007, 2024.
- [17] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove : Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics.
- [18] Robin M. Schmidt. Recurrent neural networks (rnns) : A gentle introduction and overview. *CoRR*, abs/1912.05911, 2019.
- [19] *all-about-recurrent-neural-networks*, 05 2024. <https://medium.com/@jianqiangma/all-about-recurrent-neural-networks-9e5ae2936f6e>.
- [20] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.
- [21] Ebin Babu Thomas. Understanding lstm : An in-depth look at its architecture, functioning, and pros cons. <https://www.linkedin.com/pulse/understanding-lstm-in-depth-look-its-architecture-pros-babu-thomas/>.
- [22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008, Long Beach, CA, USA, December 4-9 2017.
- [23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. In

- Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [24] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks : Distributional similarity for relation learning, 2019.
- [25] Nils Reimers and Iryna Gurevych. Sentence-bert : Sentence embeddings using siamese bert-networks, 2019.
- [26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta : A robustly optimized bert pretraining approach, 2019.
- [27] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1 : Semantic textual similarity multilingual and crosslingual focused evaluation. In Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens, editors, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics.
- [28] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm : Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020.
- [29] modèle de transformateur de phrase all-minilm-l6-v2. <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>.
- [30] Modèle minilm-l6-h384-uncased. <https://huggingface.co/nreimers/MiniLM-L6-H384-uncased>.
- [31] Martin Krallinger, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. Overview of the protein–protein interaction annotation extraction task of biocreative ii. *Genome Biology*, 9(2) :1–19, 2008.
- [32] Beth M. Sundheim. Overview of the fourth message understanding evaluation and conference. Technical report, Naval Command Control and Ocean Surveillance Center, RDT & E Division, San Diego, CA, 1992.
- [33] Arya Roy. Recent trends in named entity recognition (ner), 2021.
- [34] Nguyen Bach and Sameer Badaskar. A review of relation extraction. 05 2011.
- [35] Kartik Detroja, C.K. Bhensdadia, and Brijesh S. Bhatt. A survey on relation extraction. *Intelligent Systems with Applications*, 19 :200244, 2023.
- [36] Andrea Gasparetto, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. A survey on text classification algorithms : From text to predictions. *Inf.*, 13 :83, 2022.

- [37] Sentence similarity. <https://huggingface.co/tasks/sentence-similarity>.
- [38] Valentina Dragos. Detection of contradictions by relation matching and uncertainty assessment. *Procedia Computer Science*, 112 :71–80, 2017.
- [39] Samuel R. Bowman et al. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.
- [40] Jonas Mueller and Aditya Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.
- [41] Shane Storks, Qiaozi Gao, and Joyce Chai. Recent advances in natural language inference : A survey of benchmarks, resources, and approaches, 11 2019.
- [42] Mohamed Boubenia. *Mobile Recommendation System*. PhD thesis, Unknown, 2020.
- [43] Oren Sar Shalom, Haggai Roitman, and Pigi Kouki. *Natural Language Processing for Recommender Systems*, pages 447–483. 11 2021.
- [44] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations : Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1) :76–80, 2003.
- [45] Yue Shi et al. Collaborative filtering beyond the user-item matrix : A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1) :3, 2014.
- [46] Yu Sun et al. Bert4rec : Sequential recommendation with bidirectional encoder representations from transformers. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.
- [47] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based recommender systems : State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. Springer, Boston, MA, 2011.
- [48] Michael J. Pazzani and Daniel Billsus. *Content-Based Recommendation Systems*, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [49] Alexandros Karatzoglou et al. Content-based recommendation systems. In *Recommender Systems Handbook*, pages 627–681. Springer, 2015.
- [50] Ruisheng Zhang, Qi-dong Liu, Chun-Gui, Jia-Xuan Wei, and Huiyi-Ma. Collaborative filtering for recommender systems. In *2014 Second International Conference on Advanced Cloud and Big Data*, pages 301–308, 2014.
- [51] Jie Zou, Yifan Chen, and Evangelos Kanoulas. Towards question-based recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20*. ACM, July 2020.

- [52] M Yavorsky, RZ Useev, SA Kurushin, et al. Information technologies in law enforcement : Overview of implements and opportunities. *European Proceedings of Social and Behavioural Sciences*, 2021.
- [53] Abu Bakr Soliman, Kareem Eisa, and Samhaa R. El-Beltagy. AraVec : A set of arabic word embedding models for use in arabic NLP. In *Proceedings of the 3rd International Conference on Arabic Computational Linguistics (ACLing 2017)*, Dubai, UAE, 2017.
- [54] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.
- [55] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models, 2024.
- [56] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

Résumé

Cette thèse explore le développement d'un système intelligent et automatisé pour aider les enquêteurs des agences de maintien de l'ordre en utilisant les technologies d'intelligence artificielle (IA) et de traitement automatique du langage naturel (TALN). Le système améliore le processus d'interrogatoire en analysant et en comparant avec précision les déclarations des suspects, des témoins et des victimes, en détectant les contradictions et en recommandant des questions de suivi pertinentes. Réalisé au sein du Service Central Informatique du Commandement de la Gendarmerie Nationale, le projet inclut une conception et une mise en œuvre détaillées du système, démontrant sa capacité à améliorer la précision et l'efficacité des enquêtes tout en réduisant la charge cognitive des enquêteurs.

Mots-clés : Sécurité publique, Police judiciaire, Systèmes Intelligents, Intelligence Artificielle (IA), Traitement Automatique du Langage Naturel (TALN), Détection des contradictions, Recommandation de questions.