

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ ABDELHAMID IBN BADIS – MOSTAGANEM



Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

RAPPORT DE PROJET DE MASTER EN INFORMATIQUE
Spécialité : Ingénierie des Systèmes Informatiques
Projet soutenu dans le cadre de l'arrêté 1275



Présenté par: MOGHTET ROKIA & SLIMANI FATIMA.

THEME :

Graphes de connaissances en épidémiologie : état de l'art,
opportunités et processus épidémiologiques dynamiques
« Smart Semep »

Président: Mr. Atmani Baghdad professeur.

Encadreur: Mr. Menad Si-Med Bekkai maître assistant B.

Co-Encadreur: Mr. Mokaddem Mostefa maître de conférences B.

Examineur: Mme. Betouati Fatiha maître de conférences B.

Représentant de l'incubateur de l'Université de Mostaganem: Mr. Moussa Mohammed maître de conférences B.

Représentant du partenaire économique : Mr Bouzar Zakaria médecin spécialiste en épidémiologie et chef de service SEMEP de l'EPSP de Mostaganem.

Année Universitaire : 2022-2023

Remerciements

Louange à Dieu, Seigneur des mondes, qui nous a donné de la force et nous a aidées à mettre fin à ce projet et à l'accomplir de cette manière.

Nous remercions les membres du jury qui ont accepté de juger ce travail :

« Mr Atmani Baghdad » qui nous fait le grand honneur d'accepter de présider le jury

« Mme. Betouati Fatiha » pour l'honneur qu'ils nous font d'examiner notre travail.

Nous voudrions tout d'abord adresser toute notre reconnaissance à « Mr. Menad Si-Med Bekkai » et « Mr Mokaddem Mostéfa », d'avoir toujours été à l'écoute et l'aide qu'ils nous ont apportée, pour leur patience, leur disponibilité et surtout leurs judicieux conseils, qui ont contribué à alimenter nos réflexions.

Nous adressons nos vifs remerciements à tous ceux qui par leur encouragement ou leur amitié ont contribué à l'aboutissement de ce projet startup responsable de l'incubateur de l'Université de Mostaganem « Ms Medjahed Mustéfa » et les membres d'incubateur.

Nous remercions le représentant de l'incubateur le professeur « Ms Moussa Mohammed » pour l'honneur qu'ils nous font de participer à ce jury.

Nous remercions particulièrement le chef de service du SEMEP de l'EPSP de Mostaganem « Mr Bouzar Zakaria » pour tout l'aide et les précieux conseils qu'il nous 'a apporté.

Nous voudrions exprimer notre reconnaissance envers les professeurs de notre faculté, les amis et les collègues qui nous ont apporté leur soutien moral et intellectuel tout au long de notre démarche.

Enfin, nous remercions nos amies Hafsa Sella, Aya Benhamiti, Fatima Gani et Nour el houda Gorine qui ont toujours été là pour nous soutenir et nous encourager, leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.

Dédicaces

Je dédie ce modeste travail

A mes chers parents qui n'ont jamais cessé de formuler des prières à mon égard
de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs

Aucune dédicace

Aucun mot ne pourrait exprimer à sa juste valeur la gratitude
et l'amour que je leur porte.

A toute ma famille

A tous mes amis qui je leurs souhaite une vie pleine de bonheur et de succès.

Et finalement à tous mes collègues
pour le soutien moral et l'ambiance familiale qu'ils n'ont pas manqué d'apporter durant
cette année. Et à tous ceux qui ont fait preuve de patience à mon égard.

Moghtet Rokia

Dédicaces

A ce grand homme qui a toujours voulu que ses yeux sachent me voir un jour comme
celui-ci

Mon grand-père, que Dieu lui fasse miséricorde

A celle qui s'agenouilla devant ses pieds, ma mère, que Dieu la protège, m'a soutenu

À qui je porte son nom avec fierté, mon père, que Dieu le protège

A mes frères Mohamed Fouad Ikhlass et Abdel Rahman

A la famille de Slimani et Maamar, qui leur sont proches par le cœur, pas par le sang

Aux amis des attitudes, pas des années de connaissances

Slimani Fatima

Résumé

En modélisation épidémiologie, les réseaux dynamiques complexes sont constitués de nœuds (citoyens/individus/contacts) reliés par des liens (interactions dynamiques et temporelles entre ces individus). Les données temporelles sur les états des individus et leurs interactions jouent un rôle important quant à la propagation et/ou l'immunisation d'une épidémie et cachent des connaissances qui pourraient aider à maîtriser le comportement de cette dernière. De ce fait, la modélisation épidémiologique nécessite de plus en plus une modélisation de ces réseaux sous forme de graphes de connaissances (KG) pour représenter toute interaction complexe et temporelle durant la dynamique des citoyens/individus d'une société. Les KGs sont mieux conçus pour extraire et découvrir les connaissances cachées et pertinentes à partir des données durant la diffusion de l'information épidémiologique, le comportement des maladies, le développement des communautés épidémiologiques. Il s'agit donc, de faire usage des KGs pour modéliser et représenter des relations complexes entre les maladies, les facteurs de risque et d'autres entités pertinentes pour permettre une médecine de précision et une meilleure surveillance de la santé du citoyen.

Dans ce projet, nous proposons une stratégie d'immunisation basée sur la construction d'un KG épidémiologique. Nous implémentons le KG.

Mots-clés : Graphe de Connaissance, Graphe de Connaissance Epidémiologique, Graphe RDF, Stratégies d'Immunisation.

Abstract

In Epidemic Modeling, complex dynamic networks are made up of nodes (citizens/individuals/contacts) connected by links (dynamic and temporal interactions between these individuals). Temporal data on the states of individuals and their interactions play an important role in the spread and/or immunization of an epidemic and hidden knowledge that could help to control the behavior of the latter. As a result, epidemiological modeling increasingly requires modeling of these networks in the form of knowledge graphs (KG) to represent any complex and temporal interaction during the dynamics of the citizens/individuals of a society. KGs are best designed to extract and discover hidden and relevant knowledge from data during dissemination of epidemiological information, disease behavior, and development of epidemiological communities. It is therefore a question of making use of KGs to model and represent complex relationships between diseases, risk factors and other relevant entities to enable a precision medicine and better monitoring of citizens health.

In this project, we propose an epidemiological KG based immunization strategy. We implement the Epidemiological KG.

Keywords: Knowledge Graph, Epidemiological Knowledge Graph, RDF Graph, Community Detection, Immunization Strategies, Deep Learning, Machine Learning (ML).

ملخص

في النمذجة الوبائية، تتكون الشبكات الديناميكية المعقدة من عقد (مواطنين/أفراد/جهات اتصال) متصلة بواسطة روابط (تفاعلات ديناميكية وزمنية بين هؤلاء الأفراد). تلعب البيانات الزمنية عن حالات الأفراد وتفاعلاتهم دورًا مهمًا في انتشار و/أو تحصين الوباء وإخفاء المعرفة التي يمكن أن تساعد في التحكم في سلوك هذا الأخير. نتيجة لذلك، تتطلب النمذجة الوبائية بشكل متزايد نمذجة هذه الشبكات في شكل رسوم بيانية معرفية (KG) لتمثيل أي تفاعل معقد وزمني خلال ديناميكيات المواطنين/الأفراد في المجتمع. تم تصميم KGs بشكل أفضل لاستخراج واكتشاف المعرفة المخفية وذات الصلة من البيانات أثناء نشر المعلومات الوبائية، وسلوك المرض، وتطوير الفئات الوبائية. وبالتالي، فإن الأمر يتعلق بالاستفادة من KG لنمذجة وتمثيل العلاقات المعقدة بين الأمراض وعوامل الخطر والكيانات الأخرى ذات الصلة لتمكين الطب الدقيق ومراقبة صحة المواطن بشكل أفضل. في هذا المشروع، نقترح إستراتيجية تحصين وبائية قائمة على KG من أجل تطبيق نهج جديد للكشف عن الفئات. نقوم بتنفيذ Epidemiological KG.

الكلمات الرئيسية: الرسم البياني المعرفي، الرسم البياني المعرفي الوبائي، الرسم البياني RDF، كشف الفئات، استراتيجيات التحصين، التعلم العميق، التعلم الآلي. (ML)

Liste des figures

Figure N°	Titre de la figure	Page
Figure 1	Réseau épidémiologique de transmission du VIH	7
Figure 2	Réseau temporel	9
Figure 3	Réseau-aléatoire-vs-réseau-libre-dechelle-Barabasi-Oltvai-2004	11
Figure 4	Procédure pour un graphe en forme d'anneau	12
Figure 5	Un triple dans un graphe orienté et étiqueté	14
Figure 6	Exemple d'une ontologie	15
Figure 7	Un graphe de connaissances suite à une détection d'objets dans une image	16
Figure 8	Graphes de connaissances pour la recherche et la recommandation	18
Figure 9	Constructed Knowledge graph connecting losartan (candidate drug in COVID-19)	21
Figure 10	Architecture de l'application	25
Figure 11	Les types des stores NoSQL	26
Figure 12	Graph store	28
Figure 13	Expéditrice de réseau global	29
Figure 14	Exemple de triples RDF	30
Figure 15	Architecture de KV-Store	31
Figure 16	Le sharding	32
Figure 17	Réseau temporel en graph RDF de la population initiale	36
Figure 18	Réseau temporel en graph RDF après la propagation	37
Figure 19	Diagramme de cas d'utilisation	38
Figure 20	Diagramme de séquences	38
Figure 21	Diagramme de classe	39
Figure 22	Topologie déployé après configuration de store	39
Figure 23	Lancement du store	40
Figure 24	La classe "HandleStore"	40
Figure 25	Lancement du GatewayServer	41
Figure 26	Lancement du Flask Server	41
Figure 27	Information d'un individu	42
Figure 28	Les positions d'un contact	42
Figure 29	Information sur une position « 2 » visitée par un individu	43

Liste des abréviations

Abréviation	Expression Complète	Page
KG	Knowledge Graph	1
HIV	Human Immunodeficiency Virus	2
NoSQL	Not Only SQL	3
Py4J	Python For Java	3
API	Application Programming Interface	3
RDF	Resource Description Framework	3
VIH	Virus d' Immunodéficience Humaine	7
IA	Intelligence Artificielle	14
W3C	World Wide Web Consortium	14
OWL	Web Ontology Language	14
SWRL	Semantic Web Rule Language	14
ML	Machine Learning	15
CRUD	Create Read Update Delete	24
SEMEP	Service d'Epidémiologie et de Médecine Préventive	27
ACID	Atomicité Cohérence Isolation Durabilité	26
URI	Uniform Ressource Identifier	28
KVStore	Key Value Store	29
SPARQL	SPARQL Protocol and RDF Query Language	31
HTML	HyperText Markup Langage	32
UML	Unified Modeling Language	35

Table des matières

Introduction Générale.....	1
Chapitre 1 Graphe de Connaissances en Epidémiologie	5
1.2. Les réseaux complexes	6
1.1 Principe	6
1.2.1.1. Caractéristiques topologiques.....	7
1.2.1.2. Modèles de réseau	7
1.2.1.3. Mesures de centralité	8
1.2.1.4. Communautés	8
1.2.1.5. Dynamique du réseau	8
1.2 Propriétés	8
1.3 Réseaux temporels	8
1.3.1 Types des réseaux temporels	9
1.3.2 Random graphs (les graphes aléatoires).....	10
1.3.3 Small World (les petits mondes)	12
1.4 Graphe de connaissance.....	13
1.4.1 Définition d'un graphe de connaissance	13
1.4.2 Graphe de connaissance en IA	14
1.4.3 Graphe de Connaissance en ML	15
1.5 Etat de l'art	16
1.5.1 Graphe de connaissances	16
1.5.2 Application épidémiologie	20
1.6 Conclusion	22
Chapitre 2 Conception.....	23
2.1. Introduction	24
2.2. Architecture de l'application.....	24
2.3. Rôle de l'architecture des données dans les stores NoSQL.....	25
2.3.1. Définition des stores NoSQL	26
2.3.2. Composants des Stores NOSQL	26
2.3.3. Les types des stores NoSQL	26

2.3.4.	Caractéristiques de NoSQL.....	28
2.3.5.	Graphes NOSQL orienté.....	29
2.3.5.1.	Réseau temporel en graphe RDF	29
2.3.6.	Graphe RDF	30
2.3.7.	KVStore	30
2.3.7.1.	Les nœuds de réplication et les shards.....	31
2.3.7.2.	Facture de réplication	32
2.3.7.3.	Partitions.....	32
2.3.7.4.	Zone	33
2.3.7.5.	Topologie.....	33
2.4.	SPARQL	33
2.5.	Gateway Server pour Py4J.....	34
2.6.	RESTful API avec Flask et Python.....	34
2.7.	Réseau temporel et graphe RDF	35
2.8.	Modélisation UML.....	38
2.8.1.	Smart city et génération de la population par simulation	38
2.8.1.1.	Diagramme de cas d'utilisation	38
2.8.1.2.	Diagramme de séquences	38
2.8.1.3.	Diagramme de classe	39
2.9.	Conclusion	39
Chapitre 3 : Implémentation.....		38
3.1.	Introduction.....	39
3.2.	Les langages de programmation utilisés	39
3.3.	La configuration des stores NOSQL.....	39
3.4.	L'accès au store NoSQL via Apache Jena API (Java)	40
3.5.	Gateway Server	41
3.6.	Flask server	41

3.7. Conclusion	43
Conclusion Générale	44

Introduction Générale

Introduction Générale

Contexte

Les épidémies représentent un danger important pour la santé des citoyens, car elles peuvent se propager rapidement et toucher un grand nombre de personnes en peu de temps. De ce fait, elles risquent également d'imposer un impact économique important, en particulier si elles entraînent des fermetures d'entreprises, des pertes d'emplois ou des restrictions de voyage et de commerce.

Actuellement les réseaux dynamiques complexes jouent un rôle important en modélisation épidémiologie car ils permettent de modéliser et d'étudier les comportements dynamiques complexes d'une épidémie selon les interactions entre les individus d'une société et ses agents pathogènes. Ils sont constitués de nœuds (individus ou contacts) reliés par des liens (interactions dynamiques et temporelles entre ces individus). Il s'agit de mieux comprendre les mécanismes de transmission des maladies contagieuses et le développement de stratégies de prévention et de contrôle de plus en plus efficaces. Encore si important est d'identifier les sous-populations les plus susceptibles de transmettre des maladies.

Récemment, la modélisation épidémiologique s'est orientée vers les graphes de connaissances (KG) afin d'extraire les connaissances cachées et pertinentes à partir des données durant la diffusion de l'information épidémiologique, le comportement (propagation / immunisation) des maladies, le développement des communautés épidémiologiques et leurs relations de collaboration. Il s'agit donc, de faire usage des KGs pour modéliser et représenter des relations complexes entre les maladies, les facteurs de risque et d'autres entités pertinentes pour permettre une médecine de précision et une meilleure surveillance de la santé du citoyen.

Problématique

D'une part, les êtres humains se déplacent tout le temps, d'un endroit à un autre, d'une ville à l'autre, d'un continent à l'autre. En quelques heures seulement, ils peuvent parcourir des milliers de kilomètres. Ces déplacements augmentent la vitesse de propagation d'une épidémie et la rendent difficile à contrôler et à éradiquer. Une première étape en épidémiologie afin de maîtriser la propagation d'une épidémie serait de contrôler les déplacements des personnes afin de détecter le/les groupes d'individus (communauté(s)) porteurs à immuniser. Cette communauté comprend des individus qui contribuent fortement à la propagation de l'épidémie

Introduction Générale

(individus infectés) ainsi que des individus à faible immunité qui risquent d'être infectés. La propagation d'une maladie aura lieu lorsqu'un individu infecté sera en contact direct sur une distance de 1 à 2 mètres pendant une durée de 1 seconde avec un individu susceptible. Cet infecté lui transmettra l'épidémie et ainsi de suite, chacun la transmettra à un autre jusqu'à ce que l'épidémie se propage au sein d'une population. On dit qu'il y a épidémie, si un seuil d'individus infectés est atteint [5].

D'autre part, les individus susceptibles de contracter la maladie sont caractérisés par un état qui influence leur degré d'immunité. Cet état est décrit par un ensemble de données qu'il faut collecter. Ces données concernent généralement leurs antécédents sociaux, sanitaires, de géolocalisation, etc. Par antécédents sociaux, on entend le mode de vie (degré de pauvreté, niveau de vie, etc.) qui puisse favoriser une transmission de maladie. Par antécédents sanitaires, on entend l'évolution de l'état de santé d'un individu. A-t-il déjà contracté d'autres maladies qui favorisent la contraction de la nouvelle maladie. A titre d'exemple, les individus atteints de HIV sont très susceptibles à la tuberculose. Par antécédents de géolocalisation, on entend les conditions climatiques (humidité, chaleur, etc.) et les conditions relatives au foyer dans lequel vit cet individu.

Il s'agit d'abord de modéliser les aspects du problème afin de prendre en compte l'évolution de la maladie dans un espace dynamique. Pour pouvoir modéliser un espace dynamique, les réseaux temporels répondent au mieux à la spécification de cette dynamique. Ainsi, cette modélisation permet le contrôle de la propagation d'une épidémie et permet de mettre en œuvre les meilleures stratégies d'immunisation associées. Effectivement, ces réseaux étant souvent caractérisés par leur grande taille et leur complexité, les technologies de stockage et de traitement de données massives (Big Data) sont nécessaires pour les analyser efficacement. A ce besoin, s'ajoute de nouvelles technologies et approches pour pouvoir modéliser et étudier ces réseaux. Parmi ces technologies, les graphes de connaissances (KG) peuvent être utilisés pour modéliser et représenter les relations complexes entre les malades et les maladies, les facteurs de risque et d'autres entités pertinentes. Encore plus, ils permettent d'extraire et de découvrir les connaissances cachées à partir des données du réseau. Ce big Data cache réellement beaucoup de connaissances qui puissent améliorer les performances des stratégies d'immunisation envisagées.

Ce travail présente une modélisation par graphe de connaissances suivie par une mise en œuvre d'une stratégie de propagation/immunisation sur ce graphe. Dans un premier temps, nous préparerons le graphe de connaissances qui couvre tous les individus avec leurs états ainsi que

Introduction Générale

tous les contacts entre ces individus qui ont eu lieu dans des espaces et des intervalles de temps dynamiques. A ce graphe de connaissances, nous appliquerons en parallèle les processus de propagation et d'immunisation. Le processus de propagation contribuera à la création de communautés infectées et exposées tandis que le processus d'immunisation contribuera à leur immunisation. Nous implémentons d'abord le KG Epidémiologique obtenu comme stratégie de propagation/immunisation.

Contribution et approche

Le travail [34] a montré que la modélisation d'une population assez importante telle qu'une grande ville comme Oran nécessite beaucoup de ressources matérielles et d'algorithmes optimisés au maximum pour pouvoir manipuler cette taille massive de données. La solution utilisée pour y parvenir est le NoSQL qui présente de nombreux avantages en termes de performances et d'évolutivité. Le NoSQL permet, en fait, de faire du Big Data. Les stores NoSQL proposent des solutions Web sémantique intéressantes grâce à l'API Jena. Cela implique d'émettre des requêtes sur des graphes RDF maintenus dynamiquement. Ces graphes ne sont que des réseaux sémantiques temporels. Pour cette fin, les auteurs ont mis en œuvre des technologies comme les services web Restful en Python grâce à Flask, le serveur Gateway PY4J, l'API Jena usant du SPARQL.

Le travail [35] a exploité celui de [34] pour mettre en œuvre une visualisation graphique au niveau de la représentation graphique des données afin d'identifier les tendances et les corrélations, qui ne peuvent pas être facilement perçues dans les données brutes ou textuelles. Ce travail se concentre sur la visualisation de graphes de connaissances au format graphe RDF selon les caractéristiques de leurs nœuds en tenant compte de l'évolution de leur degré d'immunité lors de la propagation de la maladie. La visualisation a permis de montrer l'impact de ce degré sur l'évolution de la dynamique du graphe.

Structure du mémoire

Chapitre 1 : ce chapitre présente les concepts et l'état de l'art relatifs aux graphes de connaissances particulièrement en épidémiologie. Il cite les principales définitions et soulève les techniques d'exploration et de manipulation de ces graphes.

Introduction Générale

Chapitre 2 : ce chapitre aborde la spécification de la conception de la solution retenue. D'une part, il présente la modélisation des graphes de connaissances en graphe RDF. L'architecture globale de l'application y sera décrite.

Chapitre 3 : ce chapitre présente l'essentiel de la mise en œuvre effectuée. Par conséquent, il donne les détails de la configuration des stores NoSQL en vue d'héberger les graphes RDF. Enfin, une **Conclusion** clôture ce mémoire et trace les perspectives

Chapitre 1

Graphes de Connaissances en Epidémiologie

1.1. Introduction :

Les réseaux complexes sont des réseaux à grande échelle avec des topologies complexes qui ont attiré l'attention des chercheurs scientifiques. Ils sont modélisés sous forme de graphes, où les nœuds sont interconnectés par des liens représentant leurs interactions et leur dynamique [1]. Ces réseaux peuvent être utilisés pour cartographier les interactions entre les individus, les communautés et les agents pathogènes, et pour identifier les facteurs clés qui contribuent à la propagation d'une maladie.

Les graphes de connaissance pour représenter et organiser les connaissances épidémiologiques permettent aux chercheurs de synthétiser les informations provenant de multiples sources et de découvrir de nouvelles associations entre les différentes données et entités en relation [2]

Dans ce chapitre, nous allons présenter l'état de l'art de la modélisation de ces réseaux, dont les réseaux aléatoires, les réseaux Small-world (petits mondes), les réseaux temporels.

Les réseaux de contacts, étant un cas de réseaux complexes temporels intéressant en santé, nous allons expliciter ce cas. Plus particulièrement, nous allons nous intéresser à la spécification de réseaux de contacts en graphe de connaissance en général et en épidémiologie en particulier.

1.2. Les réseaux complexes :

Les réseaux complexes sont des systèmes complexes formés par des nœuds reliés entre eux par des liens ou des arcs. Les réseaux complexes peuvent être utilisés pour modéliser de nombreux phénomènes dans différents domaines, tels que la biologie, la sociologie, la physique et l'informatique [3].

Un réseau complexe est un graphe (réseau) avec des caractéristiques topologiques non triviales, des caractéristiques qui n'apparaissent pas dans des réseaux simples tels que des treillis ou des graphes aléatoires, mais qui se produisent souvent dans des réseaux représentant des systèmes réels.

1.1 Principe :

Pour analyser ces réseaux et en extraire des informations utiles, plusieurs principes importants doivent être pris en compte :

1.2.1.1. Caractéristiques topologiques

Les réseaux complexes ont souvent des caractéristiques topologiques particulières, telles que la distribution de degré, la longueur des chemins les plus courts, la connectivité, etc. L'analyse de ces caractéristiques peut fournir des informations importantes sur la structure du réseau et les propriétés de ses nœuds [4].

1.2.1.2. Modèles de réseau :

Les réseaux complexes peuvent souvent être modélisés à l'aide de modèles mathématiques tels que les graphes aléatoires, les réseaux de Watts-Strogatz ou les réseaux de Barabási-Albert. L'utilisation de ces modèles peut aider à comprendre la structure et les propriétés des réseaux complexes. La figure 1 représente un exemple de réseau épidémiologique de transmission du VIH, généré à partir des données. Les sommets sont des individus, et une arête entre deux représente la transmission du VIH (soit par contact sexuel, soit par l'usage de drogues par voie intraveineuse).

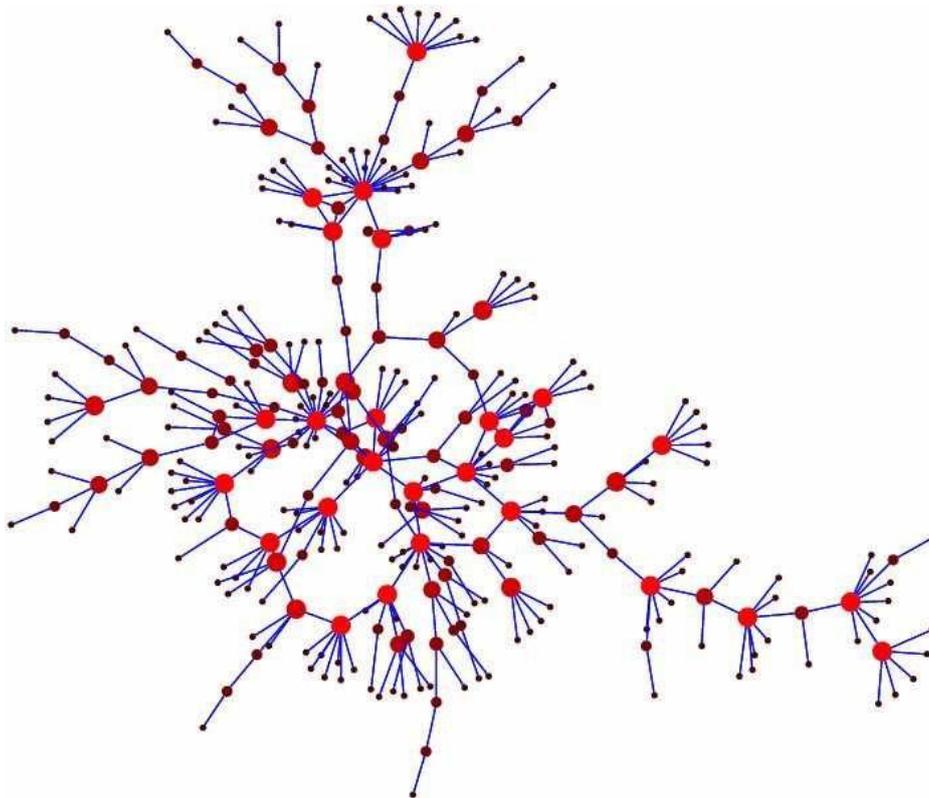


Figure 1 : Réseau épidémiologique de transmission du VIH

1.2.1.3. Mesures de centralité :

Les mesures de centralité permettent de déterminer l'importance relative des nœuds dans un réseau, en fonction de leur position dans le réseau. Les mesures courantes de centralité incluent le degré, la centralité d'intermédiarité, la centralité de proximité, etc.

1.2.1.4. Communautés :

Les réseaux complexes peuvent souvent être divisés en groupes ou communautés de nœuds qui sont fortement connectés les uns aux autres, mais moins connectés avec les nœuds en dehors de leur communauté. L'identification de ces communautés peut aider à comprendre la structure et les fonctions du réseau.

1.2.1.5. Dynamique du réseau :

Les réseaux complexes peuvent être dynamiques, avec des changements dans les connexions entre les nœuds au fil du temps. L'analyse de la dynamique du réseau peut aider à comprendre comment les interactions entre les nœuds évoluent au fil du temps.

1.2 Propriétés :

Les réseaux complexes sont des structures complexes qui présentent des caractéristiques telles que des propriétés de petits mondes, de distribution libre d'échelle et de modularité, qui se prêtent bien à l'identification de communautés. L'identification de communautés dans les réseaux complexes est une tâche importante dans l'analyse des réseaux, car elle peut aider à révéler la structure et les modèles cachés dans les données. Les méthodes pour identifier les communautés dans les réseaux complexes incluent des approches basées sur la détection de clusters, l'optimisation de la modularité, les mesures de centralité, les algorithmes de propagation de labels, et d'autres techniques avancées telles que les réseaux de neurones. Les principales propriétés des réseaux complexes comprennent la connexité, le plus court chemin, l'hétérogénéité, la modularité, la résilience et les propriétés émergentes.

1.3 Réseaux temporels :

Les réseaux temporels sont souvent représentés sous la forme de graphes pondérés dans le temps, où chaque lien entre deux nœuds est associé à une valeur numérique qui représente la durée ou le moment de ce lien. Les réseaux temporels peuvent être analysés et modélisés à l'aide de techniques d'analyse de réseaux, telles que les modèles de Markov en temps continu, les

processus de diffusion sur les réseaux, les algorithmes de détection de communautés temporelles, etc.

Les réseaux temporels ont des applications dans de nombreux domaines, notamment en biologie, en physique, en sociologie, en informatique et en ingénierie. Les réseaux temporels permettent de mieux comprendre la dynamique des phénomènes complexes et de prédire l'évolution de ces phénomènes au fil du temps.

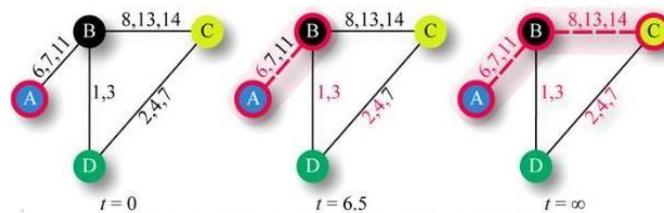


Figure 2: Réseau temporel

La figure 2 est une illustration du problème d'accessibilité et de l'intransitivité des réseaux temporels (plus précisément une séquence de contacts). Les temps des contacts entre les sommets A–D sont indiqués sur les arêtes. Supposons que, par exemple, une maladie commence à se propager au sommet A et se propage davantage dès qu'un contact se produit. Les lignes en pointillés et les sommets montrent ce processus d'étalement à quatre moments différents. La propagation ne se poursuivra pas plus loin que ce qui est indiqué dans l'image $t = \infty$, c'est-à-dire que D ne peut pas être infecté. Cependant, si la propagation commençait au sommet D, l'ensemble des sommets finirait par être infecté. L'agrégation des arêtes dans un seul graphe statique ne peut pas capturer cet effet qui découle de l'ordre temporel des contacts.

1.3.1 Types des réseaux temporels :

1.3.1.1 Contacts de personne à personne :

Ce type de réseau est dynamique car les liens entre les personnes peuvent évoluer au fil du temps. Les contacts peuvent apparaître ou disparaître en fonction des interactions entre les individus, et leur poids peut varier en fonction de facteurs tels que la fréquence ou la durée des contacts. Les réseaux de contacts de personne à personne sont souvent utilisés pour étudier la propagation de maladies contagieuses, car ils permettent de modéliser la transmission de la maladie à travers les interactions entre les individus.

1.3.1.2 La diffusion d'information un à plusieurs :

Les réseaux de diffusion d'information peuvent être modélisés comme des réseaux temporels, où les nœuds représentent les individus et les liens représentent la transmission de l'information entre eux. Les réseaux de diffusion d'information sont souvent utilisés pour étudier la propagation de fausses informations, de rumeurs et de propagandes dans les réseaux sociaux, ainsi que pour identifier les personnes clés dans la diffusion de l'information. La modélisation de ces réseaux en tant que réseaux temporels peut fournir une compréhension plus approfondie de la dynamique de la propagation de l'information au fil du temps.

1.3.1.3 Proximité physique :

Les réseaux de proximité physique sont souvent utilisés pour étudier la transmission de maladies contagieuses et pour comprendre les interactions sociales dans des environnements tels que les écoles, les lieux de travail et les zones résidentielles. Les réseaux de proximité physique peuvent être analysés à l'aide de l'analyse de réseau temporel et comment cette approche peut être utilisée pour mieux comprendre la dynamique des interactions sociales et la propagation des maladies [5].

1.3.2 Random graphs (les graphes aléatoires):

Les graphes aléatoires ont été introduits par Erdős et Rényi en 1959 pour prouver certains résultats combinatoires sur les graphes, Ils sont des modèles mathématiques utilisés pour représenter des réseaux complexes, tels que les réseaux sociaux, les réseaux de communication ou les réseaux d'interaction entre individus dans le cadre de la propagation d'une épidémie. Ces graphes sont construits de manière aléatoire, en attribuant des connexions entre les nœuds selon une certaine probabilité [6].

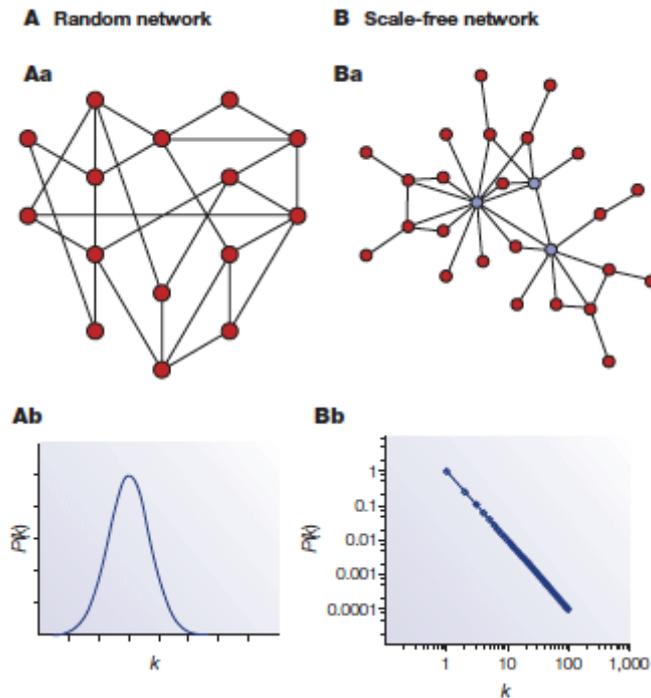


Figure 3 : Réseau-aléatoire-vs-réseau-libre-dechelle-Barabasi-Oltvai-2004

1.3.2.1 Principe :

- **Le modèle d'Erdős-Rényi:** Les auteurs décrivent un modèle de création de graphes aléatoires dans lequel un graphe à n nœuds est créé en ajoutant chaque arête avec une probabilité p indépendante des autres arêtes.
- **La phase de seuil :** Les auteurs montrent qu'il existe un seuil critique de probabilité, en dessous duquel le graphe aléatoire est très probablement non connexe et au-dessus duquel il est très probablement connexe.
- **Le nombre de composantes connexes :** Les auteurs étudient le nombre de composantes connexes dans le graphe aléatoire et montrent que lorsque la probabilité d'ajouter une arête est suffisamment grande, le graphe a une seule composante connexe presque sûrement.
- **La distribution de degré :** Les auteurs étudient la distribution de degré des nœuds dans le graphe aléatoire et montrent que pour des valeurs suffisamment grandes de n et p , la distribution de degré suit une loi de Poisson [7].

1.3.2 Propriétés :

Pour la probabilité p de connecter deux nœuds dans le graphe aléatoire est petite, le nombre de liens entre les nœuds est réduit et le nombre de composantes connexes est faible. Cependant, une composante géante émerge lorsque p atteint une certaine valeur seuil, où la plus grande composante du graphe contient un nombre linéaire de nœuds en fonction de la taille du graphe. Cette transition entre une faible connectivité et une grande composante géante est appelée phase de transition et a été largement étudiée dans la littérature scientifique.

1.3.3 Small World (les petits mondes) :

Les petits mondes sont des réseaux où chaque nœud (ou acteur) est généralement très connecté à ses voisins immédiats, mais il y a également des connexions plus distantes qui créent des raccourcis pour atteindre d'autres parties du réseau. Cette structure de réseau complexe crée des boucles courtes et des chemins directs qui permettent de naviguer facilement entre les différents nœuds. Ces réseaux sont construits à partir d'un réseau régulier dans lequel chaque nœud est initialement connecté à ses k voisins les plus proches.

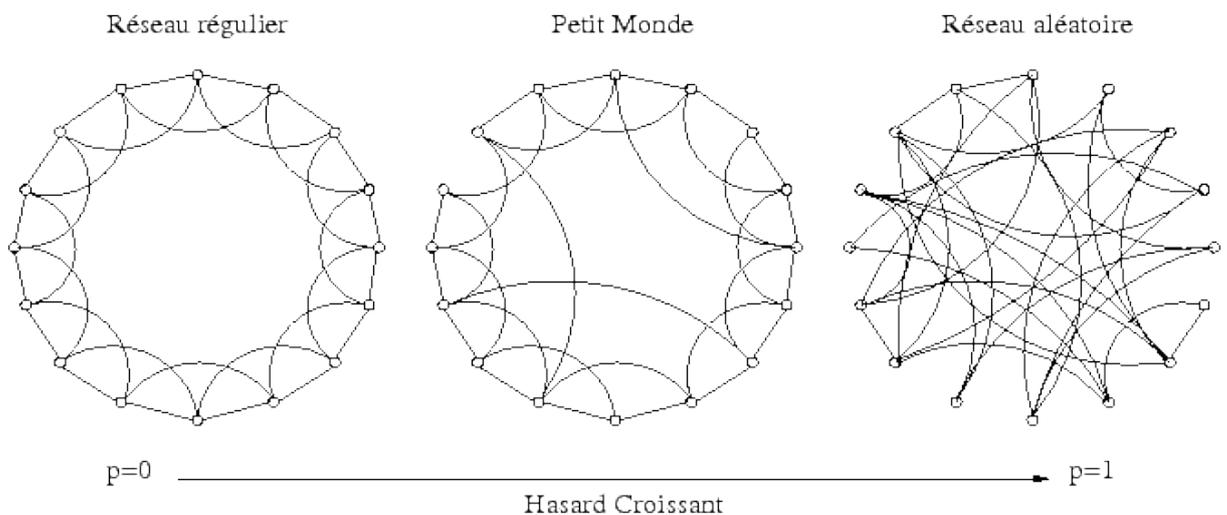


Figure 4 : Procédure pour un graphe en forme d'anneau

La figure 4 illustre cette procédure pour un graphe en forme d'anneau. Cette construction permet de créer des réseaux ayant une courte distance moyenne entre les nœuds tout en conservant une structure locale régulière, caractéristique des réseaux naturels tels que le cerveau ou les réseaux sociaux.

1.3.3.1 Propriétés :

Le concept de petits mondes a été introduit par les chercheurs Watts et Strogatz pour décrire des réseaux complexes [8]. Depuis lors, la théorie des petits mondes a été appliquée à de nombreux domaines de la science et de la technologie, y compris l'étude des réseaux sociaux, des réseaux de neurones, des réseaux informatiques, des réseaux biologiques, des réseaux de transport, etc.

Les réseaux de petits mondes ont une structure à la fois ordonnée et aléatoire, et sont souvent caractérisés par un faible diamètre (distance moyenne entre les nœuds), une forte connectivité locale (clustering) et une grande efficacité de transmission de l'information.

1.4 Graphe de connaissance:

Un graphe de connaissances, également connu sous le nom de réseau sémantique, représente un réseau d'entités du monde réel, c'est-à-dire des objets, des événements, des situations ou des concepts et illustre la relation entre eux. Ces informations sont généralement stockées dans une base de données de graphes et visualisées sous la forme d'une structure de graphe, d'où le terme « graphe » de connaissances. Un graphe de connaissances est composé de trois composants principaux : les nœuds, les arêtes et les étiquettes. Tout objet peut être un nœud. Une arête définit la relation entre les nœuds.

Les graphes de connaissances (KG) sont apparus comme une abstraction convaincante pour organiser les connaissances structurées du monde et comme un moyen d'intégrer des informations extraites de plusieurs sources de données. Les graphes de connaissances ont commencé à jouer un rôle central dans la représentation des informations extraites par application du Machine Learning au traitement du langage naturel, de la parole et de la vision. La connaissance du domaine exprimée en KG est entrée dans des modèles d'apprentissage automatique pour produire de meilleures prédictions.

Nos objectifs sont de (a) expliquer la terminologie de base, les concepts et l'utilisation des KG, (b) mettre en évidence les applications récentes des KG qui ont conduit à une augmentation de leur popularité, et (c) situer les KG dans l'IA.

1.4.1 Définition d'un graphe de connaissance:

Un graphe orienté étiqueté est un 4-uplet $G = (N, E, L, f)$, où N est un ensemble de nœuds, $E \subseteq N \times N$ est un ensemble d'arêtes, L est un ensemble d'étiquettes et $f : E \rightarrow L$, est une fonction

d'affectation des arêtes aux étiquettes. Une affectation d'une étiquette B à une arête $E=(A,C)$ peut être considérée comme un triplet (A, B, C) et visualisée comme le montre la figure 5.

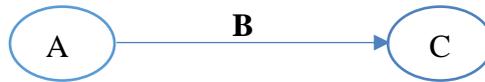


Figure 5 : Un triple dans un graphe orienté et étiqueté

Un graphe de connaissances est un graphe orienté étiqueté dans lequel sont associées des significations spécifiques à un domaine avec des nœuds et des arêtes. Tout peut agir comme un nœud, par exemple, des personnes, une entreprise, un ordinateur, etc. Une étiquette de bord capture la relation d'intérêt entre les nœuds, par exemple, une relation d'amitié entre deux personnes, une relation client entre une entreprise et une personne, ou une connexion réseau entre deux ordinateurs, etc.

La représentation graphique étiquetée dirigée est utilisée de diverses manières en fonction des besoins d'une application. Un graphe orienté étiqueté tel que celui dans lequel les nœuds sont des personnes et les arêtes capturent la relation parent est également appelé graphe de données. Un graphe dirigé étiqueté dans lequel les nœuds sont des classes d'objets (par exemple, livre, manuel, etc.) et les arêtes capturent la relation de sous-classe, est également connu sous le nom de taxonomie. Dans certains modèles de données, étant donné un triplet (A, B, C) , nous nous référons à A, B, C comme le sujet, le prédicat et l'objet du triplet respectivement.

Un graphe de connaissances sert de structure de données dans laquelle une application stocke des informations. Les informations pourraient être ajoutées au graphe de connaissances grâce à une combinaison d'entrées humaines, de méthodes automatisées et semi-automatisées. Quelle que soit la méthode d'entrée des connaissances, on s'attend à ce que les informations enregistrées puissent être facilement comprises et vérifiées par les humains.

De nombreux calculs intéressants sur un graphe peuvent être réduits à sa navigation. Par exemple, dans une amitié KG, pour calculer les amis des amis d'une personne A, nous pouvons parcourir le graphe de A à tous les nœuds B qui lui sont connectés par une relation étiquetée comme ami, puis récursivement à tous les nœuds C connectés par la relation d'ami à chaque B.

1.4.2 Graphe de connaissance en IA :

L'IA maintient des représentations du monde réel et les utilise pour raisonner. Trouver une bonne représentation est un problème central de l'IA car elle permet de stocker des informations et d'en tirer de nouvelles conclusions.

Une application largement connue des approches issues des réseaux sémantiques est l'ontologie. Une ontologie est une spécification formelle des relations utilisées dans un graphe de connaissances. Par exemple, dans la figure 6, les concepts tels que Ville, Pays, etc. et les relations comme *relation administrative*, *ville de*, *région de*, *comme*, etc., et leurs définitions formelles constituent une ontologie. En utilisant cette ontologie, nous pouvons tirer des conclusions telles qu'Oran est situé à l'ouest de l'Algérie et que comme Alger elle possède des hôpitaux et des universités.

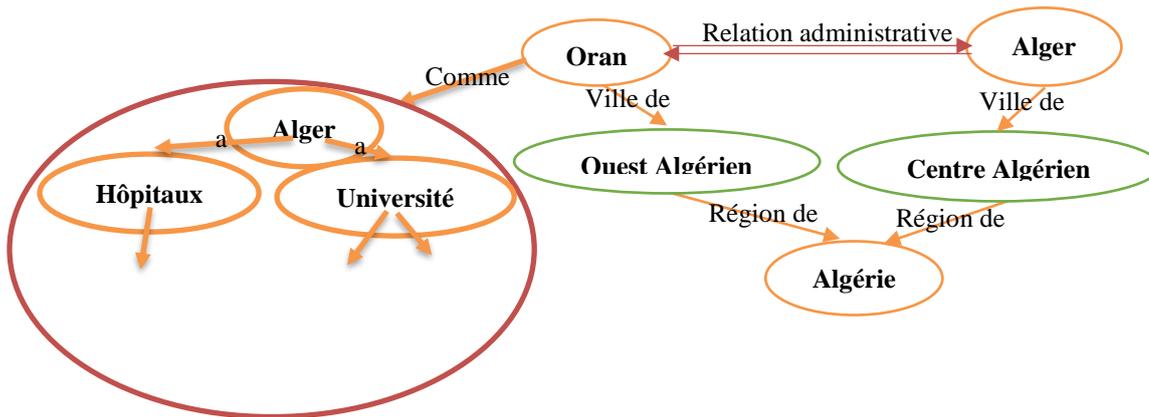


Figure 6 : Exemple d'une ontologie

Pour rendre Internet plus intelligent, le World Wide Web Consortium (W3C) a normalisé une famille de langages de représentation des connaissances qui sont désormais largement utilisés pour capturer des connaissances sur Internet. Ces langages incluent le Resource Description Framework (RDF), le Web Ontology Language (OWL) et le Semantic Web Rule Language (SWRL).

Actuellement, il y a un regain d'activité sur les approches ascendantes de l'IA, c'est-à-dire le développement d'algorithmes capables de traiter les données à partir desquelles les algorithmes peuvent tirer des conclusions et des idées.

1.4.3 Graphe de Connaissance en ML :

La compréhension complète d'une image, c'est-à-dire la détection d'objets, la description de leurs attributs et la reconnaissance de leurs relations permettrait des applications importantes telles que la classification d'images et la détection d'objets. Les algorithmes de vision par ordinateur font un usage intensif des méthodes d'apprentissage automatique telles que la classification, le clustering, les plus proches voisins et les méthodes d'apprentissage en DL telles que les réseaux de neurones récurrents.

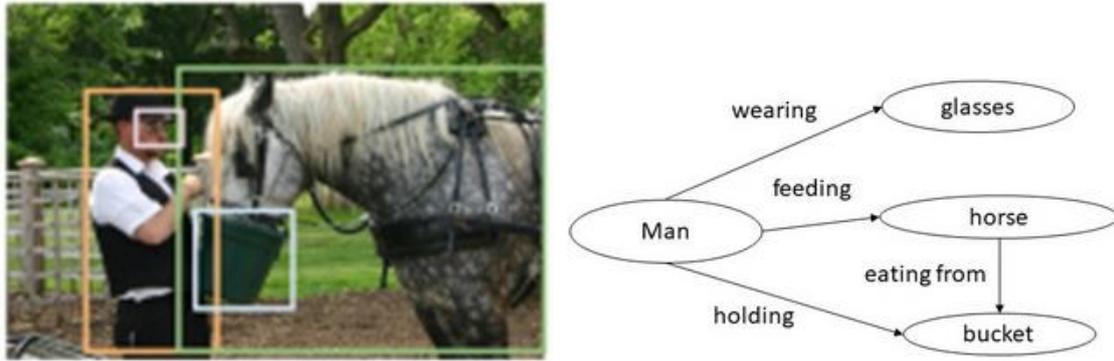


Figure 7 : Un graphe de connaissance suite à une détection d'objets dans une image

À partir de l'image illustrée à la figure 7, un système de compréhension d'image devrait produire un KG illustré à droite. Les nœuds du graphe de connaissances sont les sorties d'un détecteur d'objet. La recherche actuelle se concentre sur le développement de techniques permettant de déduire correctement les relations entre les objets, telles que l'homme tenant un seau et le cheval se nourrissant du seau, etc. Le KG illustré à droite est un exemple de graphe de connaissances qui fournit une base pour la réponse visuelle aux questions.

1.5 Etat de l'art :

1.5.1 Graphe de connaissances :

Bien que le terme KG soit devenu largement utilisé depuis sa création par Google en 2012, il souffre d'une définition bien concise [9]. Les KG ont une importance dans l'environnement commercial actuel, car ils sont essentiels pour de nombreuses entreprises. Les KG fournissent des données structurées et des connaissances factuelles, qui sont essentielles pour piloter les produits et les informations et les rendre plus intelligents, avancés, efficaces et précieux [10]. Les KG sont un outil puissant pour organiser et intégrer de grandes quantités de données provenant de diverses sources, et ils sont de plus en plus utilisés dans un large éventail de domaines, de la finance et du marketing aux soins de santé et à l'éducation [11].

Dans le domaine des finances, [12] démontre le potentiel de l'utilisation de KG et du réseau de neurones artificiels (ANN) pour améliorer la détection des hyperonymes dans le domaine des services financiers. Il a d'abord construit un KG en utilisant diverses sources de données financières, notamment des sites Web d'entreprises, des articles d'actualités financières, et les dépôts réglementaires. Ensuite, il a utilisé le KG résultant pour former un modèle d'apprentissage automatique afin d'identifier les relations hyperonymes entre les termes

financiers. Dans le domaine du marketing [13] décrit la conception et la mise en œuvre d'une plateforme de graphes de connaissances (KGP) pour le marketing de haute qualité. Il décrit également la mise en œuvre d'une application prototype qui utilise le KGP pour fournir une analyse et une prédiction du marché en temps réel. L'application utilise le KG pour analyser les tendances du marché et identifie les opportunités potentielles d'investissement. Dans le domaine de l'éducation également, [14] propose une approche basée sur le KG pour organiser et accéder à la littérature scientifique, qui implique la création d'un graphe qui relie les publications scientifiques aux concepts, auteurs et institutions connexes. Il décrit le processus de création d'un tel graphe, y compris l'utilisation de techniques de traitement du langage naturel pour extraire les informations pertinentes des publications scientifiques, et l'utilisation d'approches basées sur l'ontologie pour représenter et organiser ces informations.

En ce qui concerne le domaine de la santé, [15] décrit la construction d'un KG pour les complications du diabète en utilisant des preuves cliniques examinées par des experts. Le KG a été construit par extraction d'informations de la littérature sur les complications du diabète. Il a été structuré à l'aide d'une ontologie standard pour la représentation des connaissances médicales. Des applications potentielles du KG, notamment la prise de décision clinique et le développement de plans de traitement personnalisés pour les patients souffrant de complications du diabète ont été exposées. En outre, [16] note comment le développement d'un (KG) peut faciliter l'utilisation de données scientifiques pour permettre une médecine de précision. Il décrit le processus de construction d'un KG qui intègre des données provenant de plusieurs sources, y compris les dossiers de santé électroniques, les données génomiques et la littérature scientifique. Le KG est conçu pour prendre en charge les requêtes qui peuvent aider les cliniciens à prendre des décisions plus éclairées, en particulier en ce qui concerne le choix du traitement. L'analyse KG de [17] peut aider les chercheurs à mieux comprendre les relations complexes entre le changement climatique et la santé humaine, et les impacts les plus étudiés du changement climatique sur la santé étaient liés à la pollution de l'air, aux maladies infectieuses et aux maladies liées à la chaleur.

En épidémiologie, les KG peuvent être utilisés pour modéliser et représenter des relations complexes entre les maladies, les facteurs de risque et d'autres entités pertinentes [18]. [19] propose une approche basée sur le KG pour la surveillance de la santé publique qui intègre des données provenant de plusieurs sources pour identifier et suivre les épidémies. [20] décrit un graphe de connaissances qui intègre des informations sur le COVID-19 provenant de plusieurs sources et démontre son utilité pour la recherche et la découverte d'informations. [21] présente

une étude de cas sur l'utilisation du KGs pour améliorer les enquêtes épidémiologiques sur les épidémies de maladies d'origine alimentaire. [22] proposent un cadre pour la construction et l'analyse des KG en épidémiologie à l'aide d'ontologies. [23] décrit une approche de construction du KG, qui implique l'intégration de données provenant de plusieurs sources, y compris la littérature scientifique, les essais cliniques et les médias sociaux. Il discute également de l'utilisation de techniques de traitement du langage naturel pour extraire des informations à partir de sources de données non structurées. [24] utilise les données épidémiologiques COVID-19 de Wikidata et vérifie leur cohérence par rapport aux faits et règles épidémiologiques connus. Il identifie les données aberrantes, les tendances et les modèles dans les données, et aide à améliorer - des décisions éclairées pour lutter contre la pandémie. [25] construit un domaine KG basé sur 900 articles de recherche publiés dans le domaine de WBE et utilise la technologie informatique pour extraire des connaissances plus riches à partir de textes de littérature scientifique, puis utilise le KG pour analyser des sujets de recherche et, y compris l'utilisation de l'ML et de l'IA pour analyser les données basées sur les eaux usées. [26] vise à retracer les relations de contact potentiellement liées aux infections, à découvrir activement les personnes infectées et à empêcher la propagation de la pandémie.

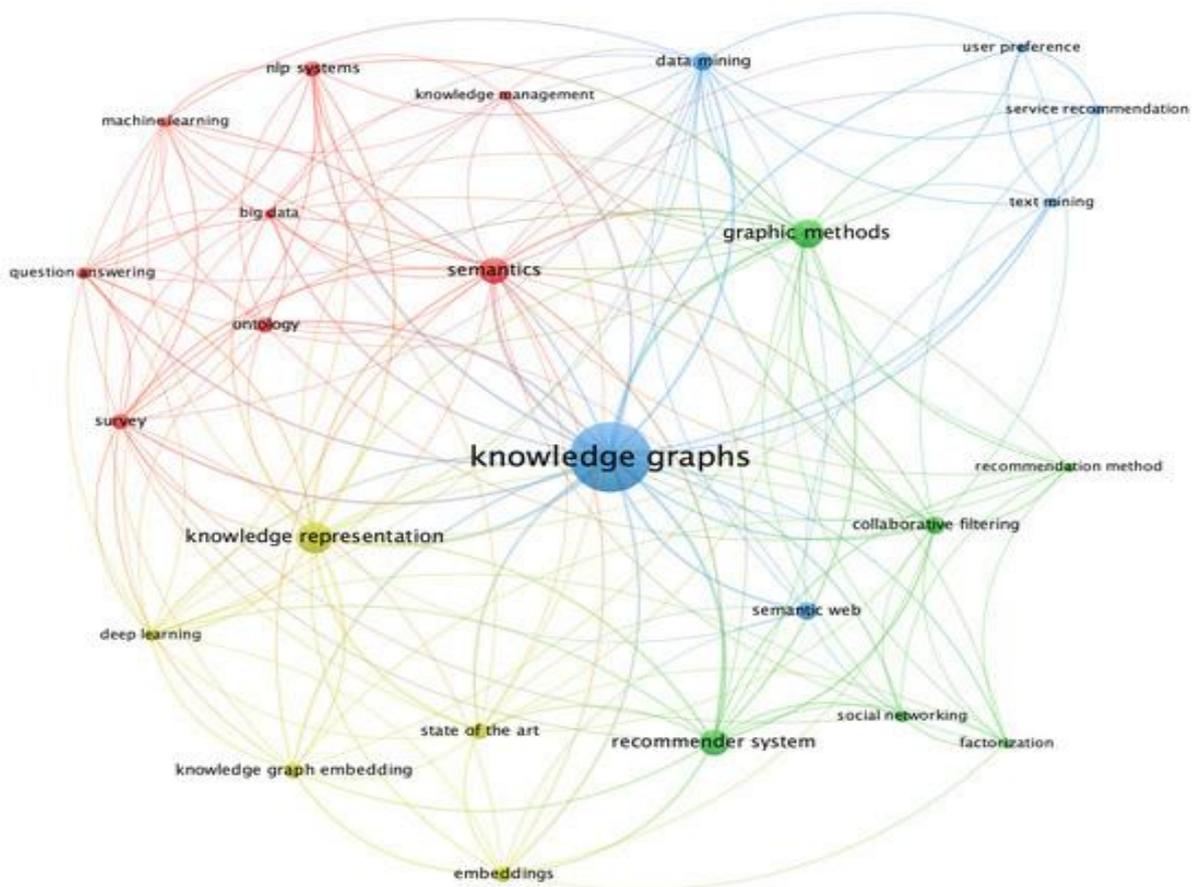


Figure 8 : Graphes de connaissances pour la recherche et la recommandation

1.5.1.1 Les approches de graphes de connaissance :

Les avancées récentes dans le domaine des algorithmes de graphes de connaissances comprennent :

1.5.1.2 Apprentissage par transfert pour les graphes de connaissances :

Des techniques d'apprentissage par transfert ont été appliquées aux représentations de graphes de connaissances, permettant un ajustement efficace de grands modèles préformés pour des tâches spécifiques [27].

1.5.1.3 Achèvement des graphes de connaissances :

Les progrès des algorithmes d'achèvement des graphes de connaissances ont permis d'améliorer la précision et l'efficacité de la prédiction des entités et des relations manquantes dans les graphes de connaissances [27].

1.5.1.4 IA explicable pour les graphes de connaissances :

Les chercheurs ont développé des méthodes pour rendre les prédictions et les décisions des modèles de graphes de connaissances plus interprétables et transparentes [28].

1.5.1.5 Graphes de connaissances fédérés :

Il y a eu un intérêt croissant pour le développement de méthodes d'intégration et de partage d'informations sur plusieurs graphes de connaissances, ce qui peut améliorer la qualité et la couverture des données sous-jacentes [29].

1.5.1.6 Graphes de connaissances multimodaux :

La recherche a progressé dans le développement de graphes de connaissances multimodaux qui peuvent incorporer des données structurées et non structurées, telles que du texte et des images, pour une représentation plus complète des connaissances [30].

1.5.1.7 Raisonnement des graphes de connaissances :

Les progrès des algorithmes de raisonnement des graphes de connaissances ont amélioré la capacité d'effectuer des inférences et de faire des prédictions basées sur les relations entre les entités d'un graphe de connaissances [31].

Dans l'ensemble, le domaine des travaux liés aux graphes de connaissances continue d'évoluer, les chercheurs explorant de nouvelles techniques et applications pour améliorer la représentation, le stockage et l'analyse des connaissances.

1.5.2 Application épidémiologie :

L'application des graphes de connaissances en épidémiologie est en plein essor. Les graphes de connaissances peuvent être utilisés pour modéliser les relations entre différentes entités telles que les maladies, les symptômes, les facteurs de risque, les traitements et les résultats. Ils présentent également des exemples concrets d'utilisation des graphes de connaissances pour surveiller la propagation de la maladie et pour identifier les groupes à haut risque [32]

En outre, les graphes de connaissances peuvent être utilisés pour améliorer la surveillance et la prévention des maladies en modélisant les schémas de transmission des maladies et en identifiant les groupes à haut risque de propagation de la maladie. En utilisant les graphes de connaissances pour analyser les données de surveillance épidémiologique, il est possible de détecter les épidémies émergentes plus tôt et de mettre en place des interventions de santé publique plus efficaces.

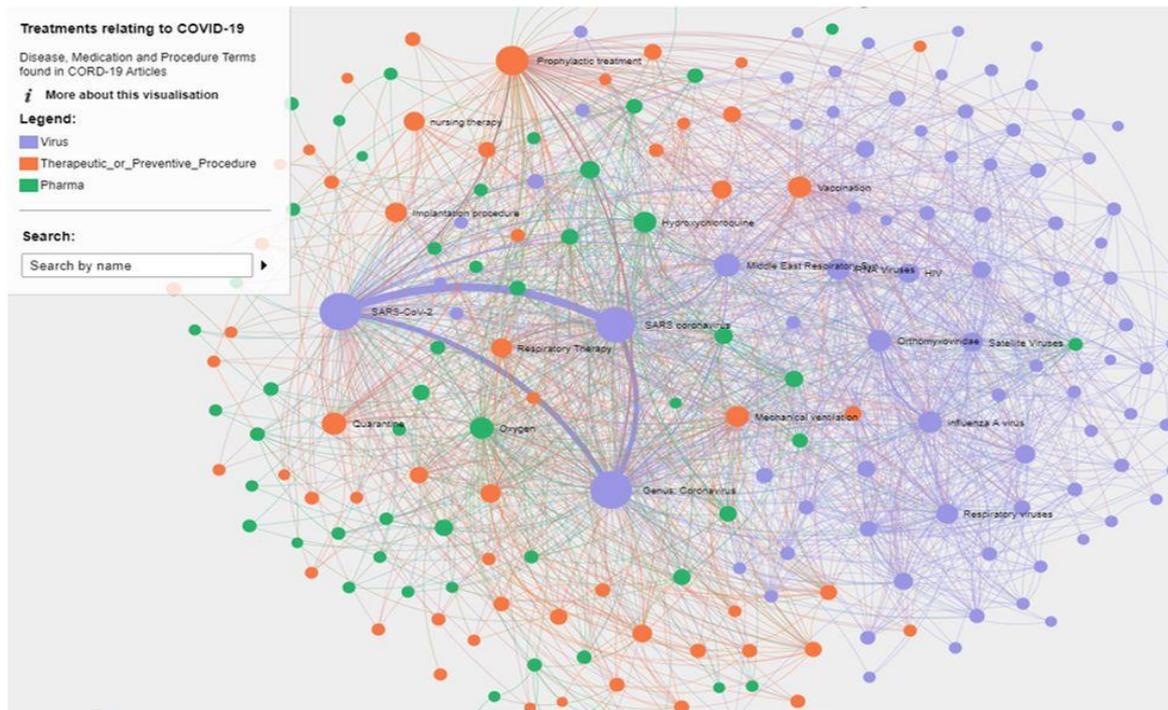


Figure 9: Constructed Knowledge graph connecting losartan (candidate drug in COVID-19)

En somme, les graphes de connaissances ont de nombreuses applications potentielles en épidémiologie et peuvent contribuer à la compréhension, la surveillance et la prévention des maladies.

1.6 Conclusion

L'étude des réseaux complexes et leur modélisation soient en constante évolution, il existe un défi majeur qui est la compréhension et la représentation des KGs. Dans ce chapitre, nous avons exploré différents types de réseaux complexes et leur modélisation.

Nous avons également examiné le concept de KG, en particulier dans le contexte des réseaux complexes de contacts. Dans le chapitre qui suit nous allons aborder l'état de l'art sur les nouvelles méthodes pour améliorer la détection des communautés.

Chapitre 2

Conception

2.1.Introduction

Ce chapitre aborde la spécification de la conception de la solution retenue. Nous commençons par l'architecture de l'application pour avoir une vue globale des niveaux d'abstraction ainsi que le partitionnement des traitements entre ces différents modules. Nous présentons les stores NoSQL et nous montrons leur construction et la construction de réseau temporel sous forme graphe RDF.

Nous spécifions les stratégies de propagation/immunisation sous forme de communautés à détecter depuis le graphe RDF (réseau temporel). A chaque étape ci-dessus, nous proposons la spécification par une modélisation UML, tout en se restreignant aux cas d'utilisation, aux diagrammes des séquences et aux diagrammes des classes. Nous terminons par une conclusion.

2.2.Architecture de l'application

Afin d'aboutir à l'architecture de l'application, nous commençons par soulever les parties majeures à expliciter. Tout d'abord, dans une zone donnée, telle qu'une cité, un quartier ou une zone limitée par un périmètre. La population de cette zone est dynamique, avec un nombre d'individus qui peut augmenter ou diminuer en fonction des entrées et des sorties. Les personnes venant d'autres zones (exogènes) peuvent se rendre dans cette zone pour faire des courses, tandis que les habitants de la zone (endogènes) peuvent aller travailler ailleurs.

Pour suivre le comportement de la zone, le concept de smart city est utilisé. La propagation de l'épidémie se fait par des contacts entre individus infectés (capables d'infecter) et individus sains (susceptibles d'être infectés), ce qui entraîne une augmentation du nombre d'infectés. Ces contacts sont modélisés par un réseau temporel de contacts, qui enregistre les points de contact et le moment où ils ont eu lieu. Un point de contact est défini comme la position (longitude, latitude) où deux individus se trouvent face à face à une distance d'1 à 2 mètres pendant au moins 1 seconde. Ainsi, nous insistons sur trois parties. La 1ère relève de la smart city, la 2ème de la propagation et la 3ème de l'immunisation.

L'architecture doit montrer les niveaux et les services qu'il faut mettre en oeuvre. Notre application utilise des stores NoSQL qui seront interrogés par des requêtes SPARQL vus que les stores utilisent des graphes RDF. Les stores sont déployés sur des machines virtuelles (VM). La manipulation de graphes RDF a besoin d'API spécifique telles que Jena. L'interaction entre

l'épidémiologiste et l'application se fait par des services web Restful qu'il faut déployer sur un/des serveur(s) tels que Flask qui sont populaires et pratiques. La communication entre le(s) serveur(s) (Flask) et le(s) store(s) nécessite des Gateways qu'il faut mettre en oeuvre. La communication entre le client (épidémiologiste) et le serveur (Flask) se fait par des requêtes http qu'il faudra spécifier. Certains outils/technologies ont été retenus dû à leur usage simple et à leur performance. Comme Flask est en Python et que le store serait implémenté sous Oracle NoSQL Database, le Gateway sera une passerelle qui traduit du Python depuis/vers Java vu que Jena est en Java. Nous devons également nous aligner aux travaux antérieurs de l'équipe AIR et faire usage des technologies spécifiées dans le cahier de charges.

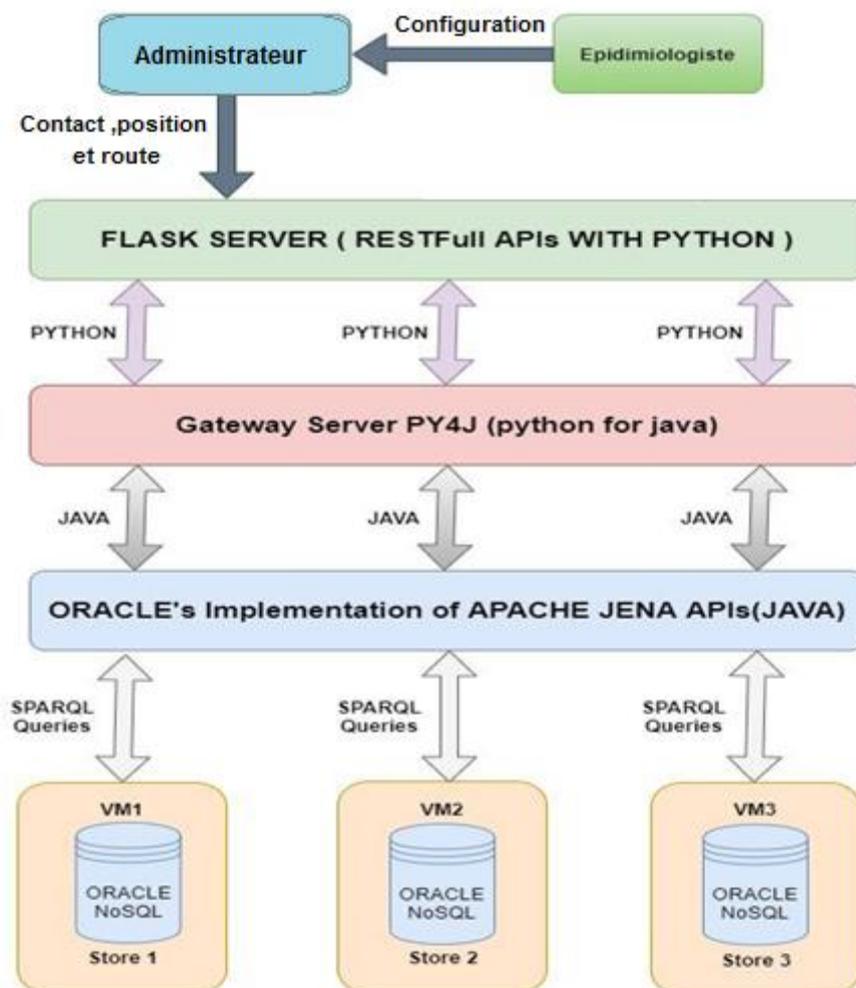


Figure 10 : Architecture de l'application [33]

2.3.Rôle de l'architecture des données dans les stores NoSQL

2.3.1. Définition des stores NoSQL

NoSQL est une approche de conception de stores de données pouvant s'adapter à une grande variété de modèles de données, y compris les formats avec des clés, des documents, des colonnes et des graphes (Graph Database). NoSQL regroupe de nombreuses bases d'actuellement son exploitation chez Google, Facebook, Amazon, Oracle et bien d'autres [36].

2.3.2. Composants des Stores NOSQL

Il y a quatre composants principaux [36]:

- **Langage de modélisation:** il définit la structure de la base de données et la façon dont les données sont organisées. Les données sont stockées sous forme de lignes et de colonnes utilisant des formats XML, et chaque donnée est associée à une clé unique. Le modèle est conçu pour permettre un accès rapide aux données.
- **Structure de la base de données:** chaque base de données utilise ses propres structures de données et stocke les données dans un dispositif de stockage permanent.
- **Langage de requête de base de données:** toutes les opérations sur la base de données, telles que la création, la mise à jour, la lecture et la suppression (CRUD), sont effectuées à l'aide d'un langage de requête spécifique à la base de données.
- **Transactions:** pendant une transaction, des erreurs ou des pannes peuvent se produire, mais le système continue de fonctionner.

2.3.3. Les types des stores NoSQL

Les bases de données NoSQL sont classées sur la base de la façon dont les données sont :

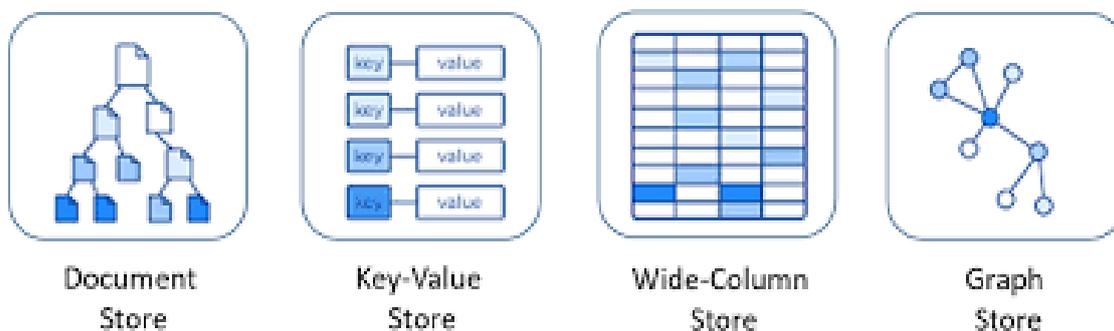


Figure 11 : Les types des stores NoSQL

- **Clé/valeur(Key-Value Store) :**

Dans ce modèle on ne dispose généralement que des quatre opérations Create Read Update Delete (CRUD) [37]:

- ✓ **Create:** créer un nouvel objet avec sa clé → create (key, value)
- ✓ **Read:** lit un objet à partir de sa clé → read (key)
- ✓ **Update:** met à jour la valeur d'un objet à partir de sa clé → update(key, value)
- ✓ **Delete:** supprime un objet à partir de sa clé → delete(key).

Les solutions les plus connues sont Redis, Riak, Voldemort créé par LinkedIn et Oracle NoSQL Database.

➤ **Orienté Colonne (Column-Oriented Store) :**

Ce modèle ressemble à première vue à une table dans un modèle relationnel. Dans les bases de données des colonnes, chaque clé est associée à un ou plusieurs attributs (colonnes). Le nombre de colonnes est fixé dès la création du schéma de la table et ce nombre reste le même pour tous les enregistrements dans cette table. Par contre, avec ce modèle, le nombre de colonnes peut varier d'un enregistrement à un autre ce qui évite de retrouver des colonnes ayant des valeurs nulles [34].

➤ **Orienté Document (Document Store) :**

Ce type de base de données stocke des documents non structurés (texte) ou semi-structurés (XML) qui sont généralement de nature hiérarchique. Ici, chaque document se compose d'un ensemble de clés et de valeurs qui sont presque les mêmes que dans les bases de données Key Value. Chaque base de données résidant dans les magasins de documents pointe vers ses champs à l'aide de pointeurs car elle utilise la technique de hachage [36]. Pour ce modèle, les implémentations les plus populaires sont CouchDB d'Apache, MongoDB et Oracle NoSQL Database [34].

➤ **Orienté Graphe (Graph Store) :**

Les bases de données de graphes sont basées sur la théorie des graphes. En général, nous voyons que le graphe se compose généralement de nœuds, de propriétés et d'arêtes. La base de données NoSQL Graph se compose de [36] :

- Les nœuds représentent des entités.

- Les propriétés représentent des attributs.
- Les bords représentent des relations.

Ce type de modèle a été utilisé dans la réalisation de la détection de communautés. La principale solution est Neo4J et Oracle NoSQL Database. La structure de la base de données graphique est comme indiqué ci-dessous :

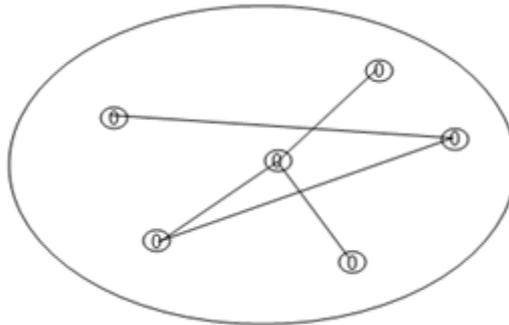


Figure 12 : Graph store

2.3.4. Caractéristiques de NoSQL

Voici quelques caractéristiques de NoSQL [37]:

- NoSQL stocke et gère de vastes volumes de données.
- Il est conçu pour fonctionner dans des environnements distribués, où les données sont réparties sur plusieurs machines, assurant ainsi une cohérence sans faille.
- Les pannes ou les défauts sur une machine n'interrompent pas le fonctionnement de l'ensemble du système.
- NoSQL est une base de données open source, ce qui signifie que son code source est accessible à tous et qu'il peut être utilisé gratuitement, sans frais de licence.
- Contrairement aux bases de données relationnelles, NoSQL ne repose pas sur le concept ACID (Atomicité, Cohérence, Isolation, Durabilité) pour garantir la cohérence des données.
- NoSQL est conçu pour être évolutif horizontalement, ce qui signifie que l'ajout de nouvelles machines permet d'obtenir de meilleures performances de manière linéaire.
- Il offre une structure plus flexible, permettant de stocker différents types de données de manière non structurée ou semi-structurée.

2.3.5. Graphes NOSQL orienté

Les bases de données graphes représentent une catégorie particulière de bases de données NoSQL. La principale caractéristique des graphes NoSQL orientés est leur capacité à représenter des relations complexes et interconnectées entre les données. Ils sont particulièrement adaptés pour modéliser des réseaux.

Nous avons retenu ce type de modèle pour la création de nos graphes de contact et de localisation vu qu'il permet la manipulation de triplets et graphes RDF utiles à notre application.

2.3.5.1. Réseau temporel en graphe RDF

Nous considérons le réseau typique de la figure 4 (exemple) comprenant des noeuds et des liens, les noeuds représentant différents individus d'un système réel et chaque lien représentant la connexion entre deux noeuds. Étant donné que la mobilité des individus peut jouer un rôle important dans le processus de propagation de l'épidémie.

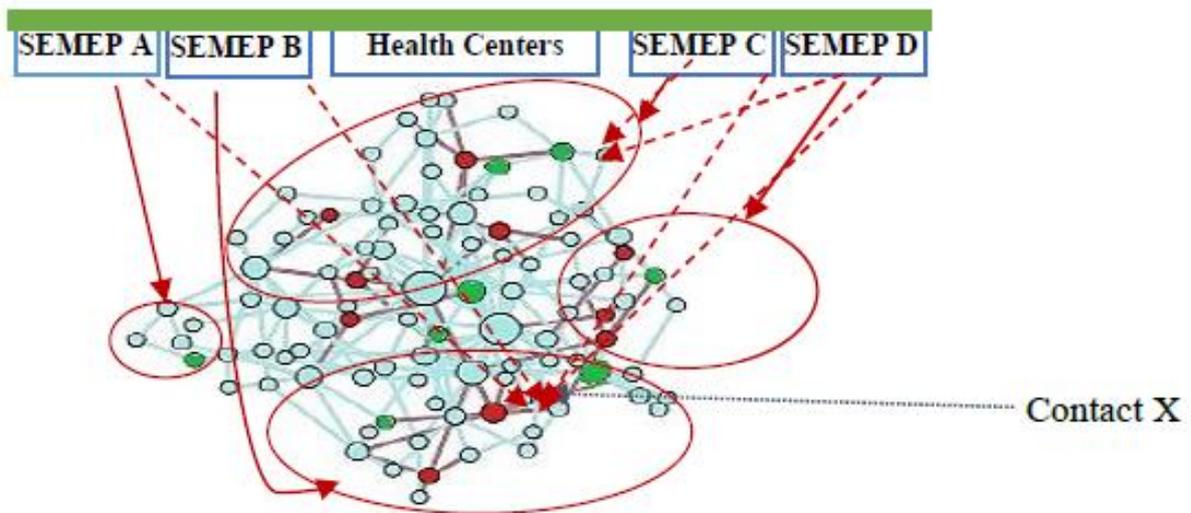


Figure 13 : Expéditrice de réseau global [33]

La figure 13 montre la relation entre les contacts et leurs stores respectifs. Certains contacts ont leurs informations stockées dans 4 stores (SEMEP A, SEMEP B, SEMEP C, SEMEP D). Une même ville peut avoir plusieurs SEMEP. Par exemple, un contact X est né dans une ville A où il a reçu certains vaccins à SEMEP A et SEMEP B de cette ville et vit maintenant dans une ville B où il a reçu d'autres vaccins en SEMEP C. Il a été, aussi, admis dans un hôpital public de la ville D. Pour rassembler toutes les informations de X, nous devons nous connecter

à ces stores distants et récupérer chaque information associée. Ces informations sont récupérées sous forme de triples RDF via une invocation de divers services web Restful [33].

2.3.6. Graphe RDF

Le RDF est une norme d'échange de données basée sur des graphes qui permet de représenter des données hautement interconnectées. Chaque instruction RDF se compose de trois parties, qui sont des ressources identifiées par des URI. La représentation des données en RDF facilite l'identification, la clarification et l'interconnexion des informations par les systèmes d'IA. RDF permet de faire des déclarations sur les ressources, et ces déclarations suivent une structure simple.

```

<u:Birth_27000100> <u:hasChromosome> <u:Chromosome_Birth_27000100>
<u:Birth_27000100> <u:hasFirstName> <Belarbi>
<u:Birth_27000100> <u:hasLastName> <Imene>
<u:Birth_27000100> <u:hasBirthDate> <1995-05-05>
<u:Birth_27000100> <u:hasMail> <Imene@hotmail.fr>
<u:Birth_27000100> <u:hasState> <S>
<u:Birth_27000100> <u:hasPhone> <0798341899>
<u:Birth_27000100> <u:hasAddressIndex> <45_Rue_Ahmed_Blaha>
<u:Birth_27000100> <u:hasCommuneIndex> <Stidia>
<u:Birth_27000100> <u:hasSex> <F>
<u:Birth_27000100> <u:hasFatherIndex> <Belarbi_Mohamed>
<u:Birth_27000100> <u:hasMotherIndex> <Mahdi_Meriam>

<u:Birth_27000100> <u:hasPosition> <u:position_2>
<u:Birth_27000100> <u:hasPosition> <u:position_1>
<u:Birth_27000100> <u:hasRoute> <u:route_12>
<u:Birth_27000100> <u:hasPosition> <u:position_3>

```

Figure 14 : Exemple de triples RDF

La figure 14 est un exemple de triples RDF décrivant le contact 'Belarbi Imane' et certains de ses mouvements. L'ensemble de tous les triples RDF d'un contact définit son graphe RDF (son ontologie). Cette ontologie couvre ses vaccinations avec les dates, les infirmières, les centres, les virus associés aux vaccinations, le poids, la tailles, le type de domicile, y compris l'humidité, le climat, la température, la pression, etc. Toutes ces données sont nécessaires pour calculer les degrés et les seuils de la maladie. Certaines de ces données sont temporelles. Le réseau temporel de données de contacts définit le réseau temporel intelligent complet sur lequel s'appliquent les stratégies.

2.3.7. KVStore

Le KVStore est un ensemble de nœuds de stockage (Storage Nodes) qui contiennent des nœuds de réplication. Les données sont réparties entre ces nœuds de réplication. Dans une architecture Web traditionnelle à trois niveaux, le KVStore remplace ou fonctionne aux côtés de la base de données principale (Back-end Database).

Le store comprend plusieurs nœuds de stockage, qui sont des machines physiques ou virtuelles dotées de leur propre stockage local. Chaque nœud de stockage peut être différent des autres en termes de matériel, bien qu'il puisse également être identique [38].

L'illustration suivante représente l'architecture typique utilisée par une application qui utilise Oracle NoSQL Database :

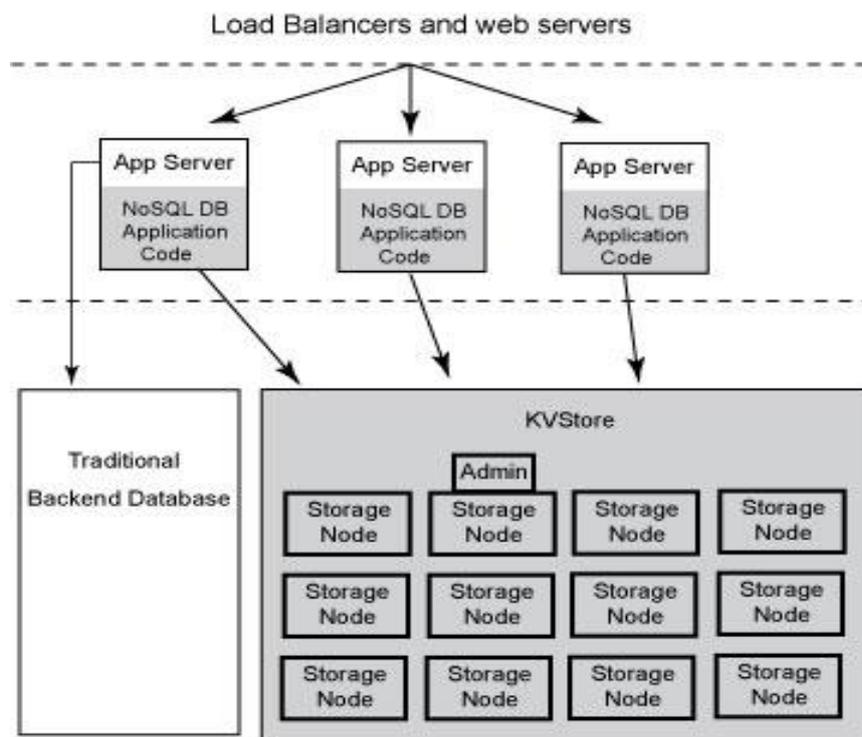


Figure 15 : Architecture de KV-Store [38]

2.3.7.1. Les nœuds de réplication et les shards

Les nœuds de réplication sont regroupés en fragments. Chaque fragment comprend un nœud maître responsable des opérations d'écriture, ainsi que des répliques en lecture seule. Le nœud

maître enregistre toutes les écritures et les réplique aux répliques pour les opérations de lecture. Bien qu'il ne puisse y avoir qu'un seul nœud maître à la fois, n'importe quel membre du fragment peut être promu en tant que nœud maître en cas de besoin. En d'autres termes, chaque fragment utilise une stratégie unique de nœud maître avec plusieurs répliques pour améliorer les performances de lecture et la disponibilité.

Les KVStores de production doivent contenir plusieurs fragments. Au moment de l'installation, vous fournissez des informations qui permettent à Oracle NoSQL Database de décider automatiquement du nombre de fragments que le magasin doit contenir. Plus votre magasin contient de fragments, meilleures sont vos performances d'écriture, car le magasin contient plus de nœuds responsables du traitement des demandes d'écriture [38].

L'illustration suivante montre comment le KVStore est divisé en shards :

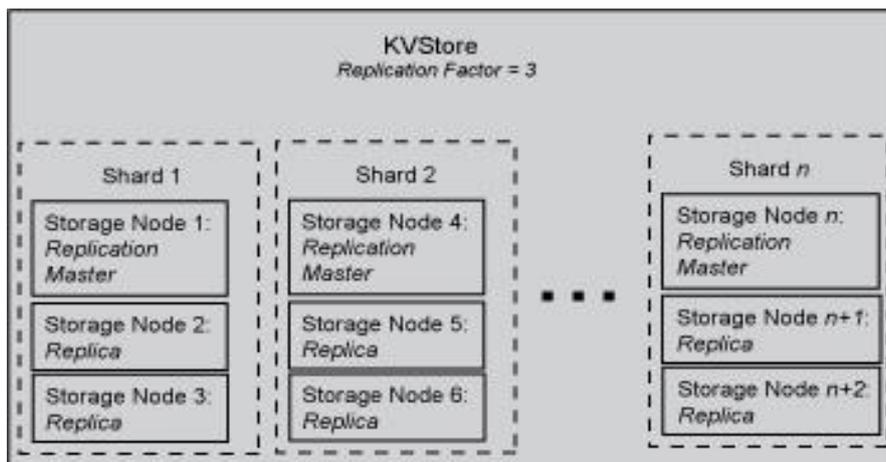


Figure 16 : Le sharding [37]

2.3.7.2. Facture de réplication

Le nombre de nœuds appartenant à un fragment est appelé son facteur de réplication. Plus le facteur de réplication d'une partition est élevé, plus son débit de lecture est rapide (car il y a plus de machines pour traiter les demandes de lecture), mais plus ses performances d'écriture sont lentes (car il y a plus de machines sur lesquelles les écritures doivent être copiées) [38].

2.3.7.3. Partitions

Les partitions de shard sont utilisées pour répartir les données sur différents nœuds de réplication dans un système distribué. Chaque shard peut avoir son propre nœud maître

responsable de l'exécution des écritures dans le shard, ainsi que des répliques en lecture seule pour traiter les opérations de lecture. Les partitions de shard garantissent que les données sont réparties de manière équilibrée entre les nœuds de réplication, ce qui permet d'optimiser les performances en parallélisant les opérations de lecture et d'écriture [38].

2.3.7.4. Zone

Une zone est un emplacement physique qui offre une connectivité réseau fiable entre les nœuds de stockage et présente une certaine séparation physique. Elle est dotée de redondance électrique, de communications de données redondantes, de contrôles environnementaux et de dispositifs de sécurité. Les zones permettent d'isoler les pannes et d'augmenter la disponibilité des données. Il existe deux types de zones : les zones principales qui contiennent des nœuds pouvant être des maîtres ou des répliques, et les zones secondaires qui contiennent uniquement des répliques [38].

2.3.7.5. Topologie

La topologie d'un magasin de bases de données NoSQL fait référence à sa configuration, comprenant les zones, les nœuds de stockage, les nœuds de réplication et les services d'administration. Elle est conçue pour éviter les points de défaillance uniques, en répartissant les nœuds de réplication sur différents fragments.

Cette approche améliore la disponibilité du magasin même en cas de panne matérielle. Les topologies peuvent être ajustées en fonction des besoins de performance ou des changements dans les nœuds de stockage. Le processus de modification et de déploiement de la topologie est itératif et peut être effectué via une interface en ligne de commande [38].

2.4.SPARQL

SPARQL est un langage de requête utilisé pour interroger des données structurées au format RDF. Il permet d'extraire des informations spécifiques à partir d'un graphe RDF en utilisant des motifs triples. Avec SPARQL, il est possible de rechercher, filtrer et combiner les données RDF de manière flexible et puissante. Ce langage offre des fonctionnalités avancées telles que les jointures, les filtres et les regroupements, ce qui en fait un outil essentiel pour l'exploration et

l'analyse de données liées. SPARQL joue un rôle important dans le domaine du Web sémantique et facilite l'interopérabilité des données ainsi que la découverte de connaissances.

Nous donnons ici un exemple de requête SPARQL de sélection de données :

```
select ?g ?s ?p ?o where { graph <http://localhost:5000/ContactGraph> { ?s ?p ?o } }
```

Cette requête permet d'afficher toutes les données sous forme de triples RDF (s:subject, p:predicate, o:object) enregistrées dans le graphe ContactGraph.

La requête suivante permet de supprimer le graphe ContactGraph :

```
drop graph <http://localhost:5000/ContactGraph>
```

On peut filtrer sur les autres prédicats, par exemple. pour trouver l'URI correspondante exactement au latitude « 0.17019 »

```
select ? subject where { ?subject <u :has latitude> 0.17019 }
```

2.5. Gateway Server pour Py4J

Le Gateway Server pour Py4J est un composant essentiel qui facilite la communication entre Python et Java. En tant que serveur exécuté côté Java, il agit comme un pont entre les deux langages, permettant aux développeurs Python d'accéder et d'invoquer des objets et des méthodes Java. Grâce à cette fonctionnalité, les développeurs peuvent intégrer des bibliothèques, des Framework ou des applications Java existantes dans leurs projets Python, ou vice versa, sans rencontrer de problèmes de compatibilité. Le Gateway Server offre une interopérabilité transparente entre Python et Java, permettant un partage facile de données et de fonctionnalités entre les deux environnements de programmation.

2.6. RESTful API avec Flask et Python

Une RESTful API est une interface de programmation d'application qui suit les principes de l'architecture REST (Representational State Transfer). Il permet aux clients (tels que les applications Web ou mobiles) d'interagir avec un serveur en envoyant des requêtes et en recevant des réponses de manière standardisée [39]. Flask est un framework Web léger écrit en Python qui est couramment utilisé pour créer des applications Web, y compris des API RESTful. Il offre un moyen simple et flexible. Cependant, de nombreuses extensions permettent d'ajouter facilement ces fonctionnalités. Il se base sur deux modules Werkzeug et Jinja2 pour offrir plusieurs des fonctions suivantes [39]:

- Serveur de développement et debugger

- Simplifie l'écriture de tests unitaires
- C'est un moteur de Template pour le rendu HTML
- Documentation complète
- Se base sur l'Unicode
- Déploiement aisé sur plusieurs hébergeurs
- Ajout de fonctionnalités via les extensions

2.7.Réseau temporel et graphe RDF

On doit créer un réseau temporel en graphe RDF d'une façon statique pour valider le déroulement et les traitements faits par l'application. Le graphe de la figure 2 se compose de 33 individus, dont 8 individus infectés en rouge et 15 individus susceptibles en mauve et 10 individus rétablis en vert. La figure 26 montre la population initiale et la figure 27 montre le graphe après une propagation.

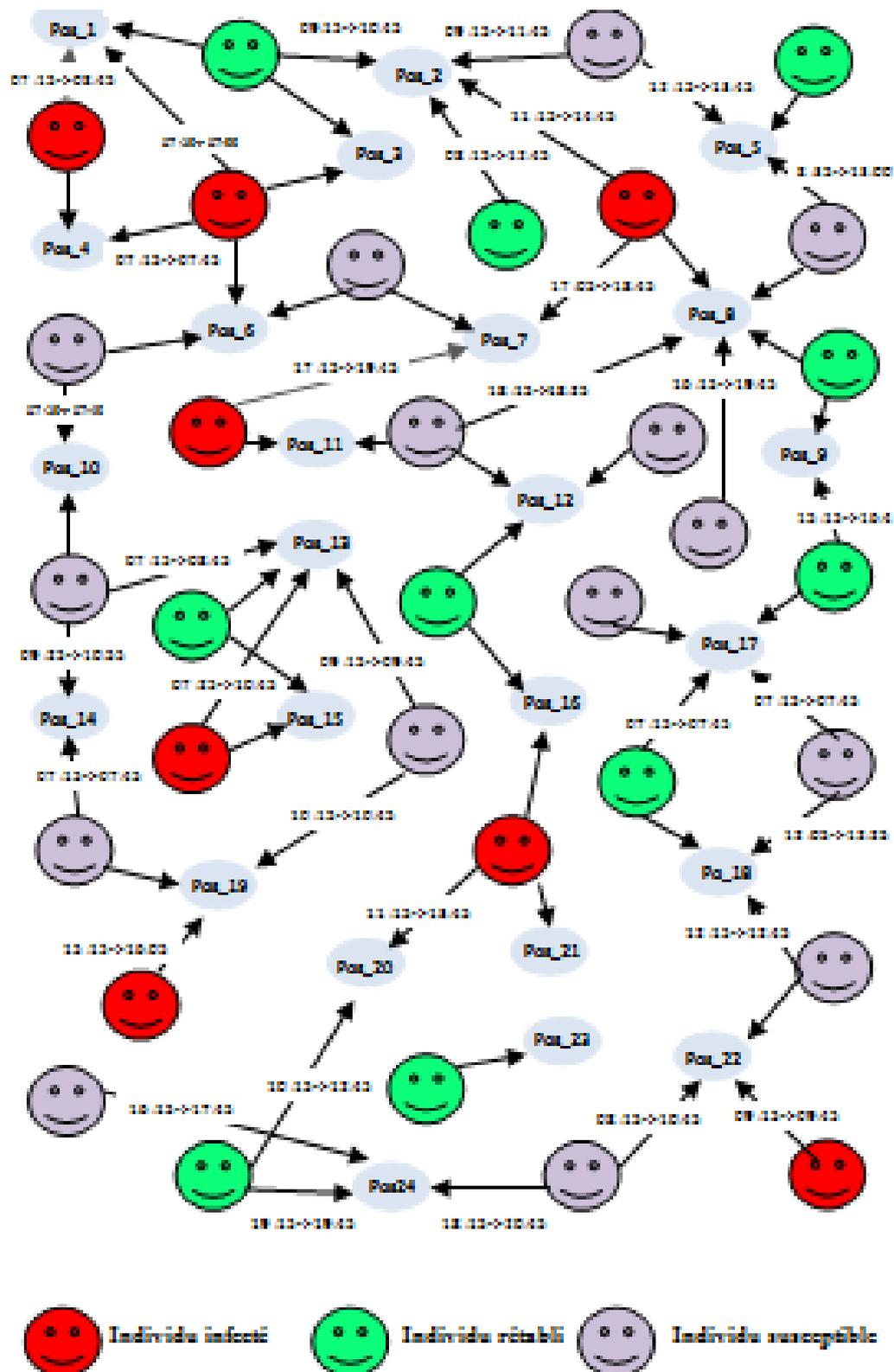


Figure 17 : Réseau temporel en graph RDF de la population initiale

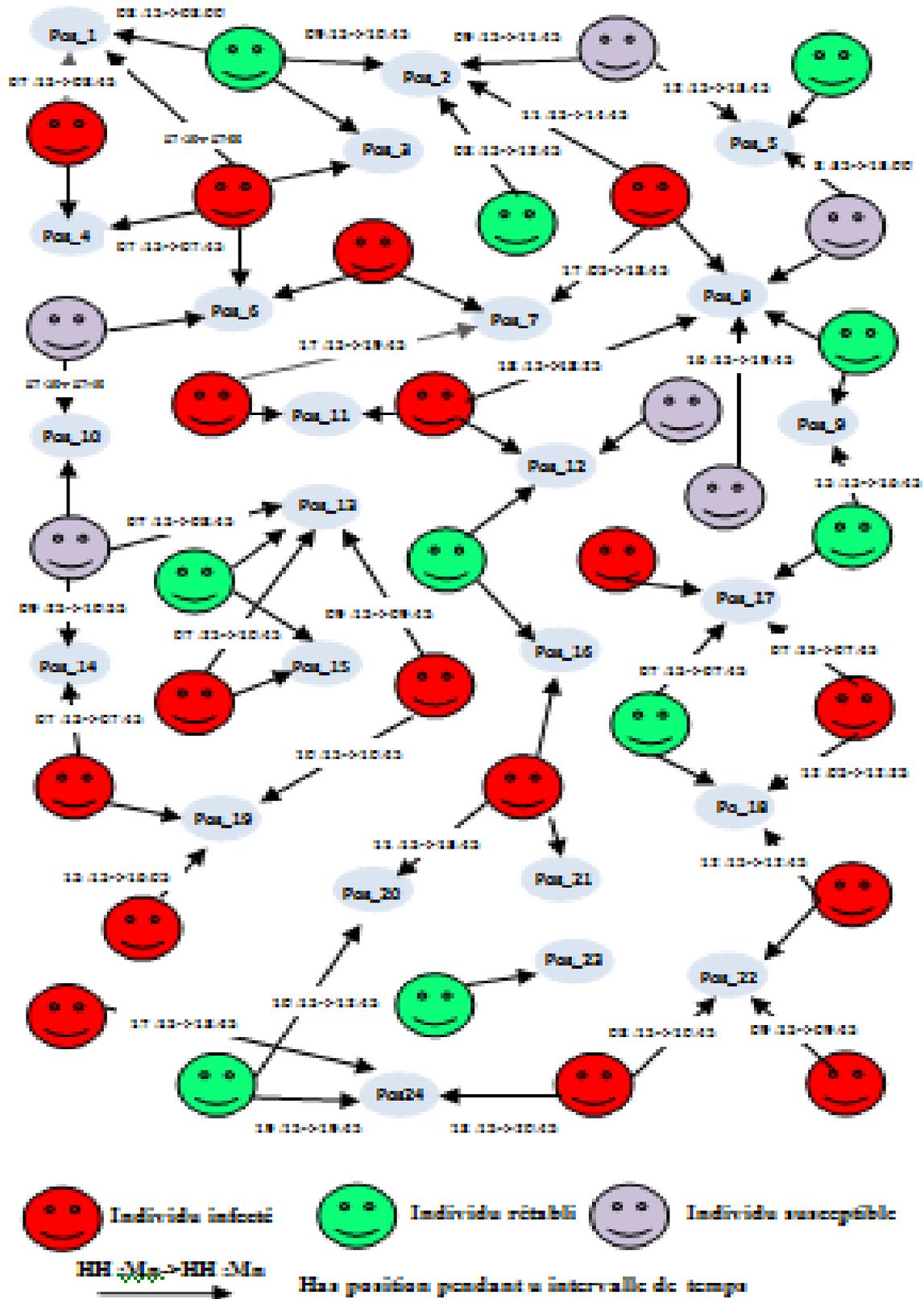


Figure 18 : Réseau temporel en graph RDF après la propagation

2.8.Modélisation UML

2.8.1. Smart city et génération de la population par simulation

2.8.1.1. Diagramme de cas d'utilisation

L'épidémiologiste crée une population comprenant plusieurs contacts, qu'il ajoute individuellement au graphe NoSQL ContactGraph. Chaque contact généré contient des itinéraires, qui sont également créés au fur et à mesure de la génération des contacts. Étant donné que chaque itinéraire est une collection de positions, celles-ci sont également générées dans un cercle spécifié par une position et un rayon. Les itinéraires et leurs positions sont enregistrés dans le graphe PositionGraph (Figure19). Pour construire le graphe temporel des contacts, l'épidémiologiste explore les graphes ContactGraph et PositionGraph à la recherche de tous les contacts dont les itinéraires se croisent avec un barycentre donné. Ces contacts sont regroupés dans une liste.

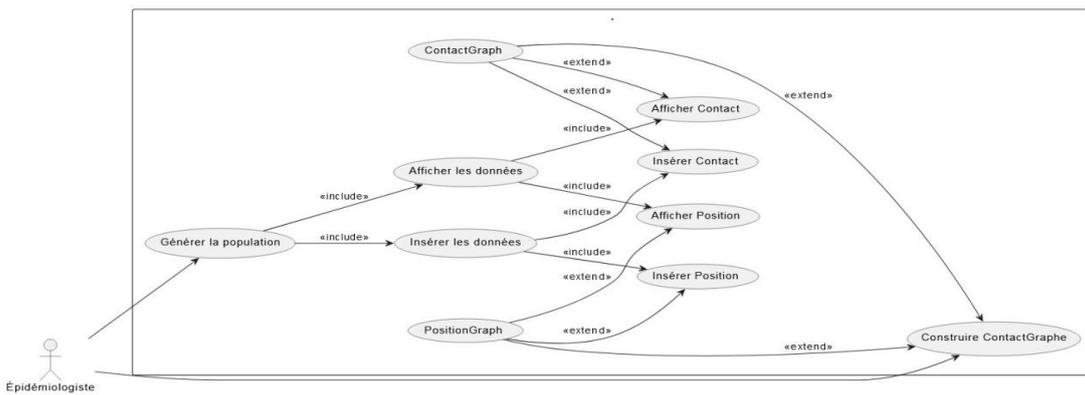


Figure 19 : Diagramme de case d'utilisation d'épidémiologiste

2.8.1.2. Diagramme de séquences

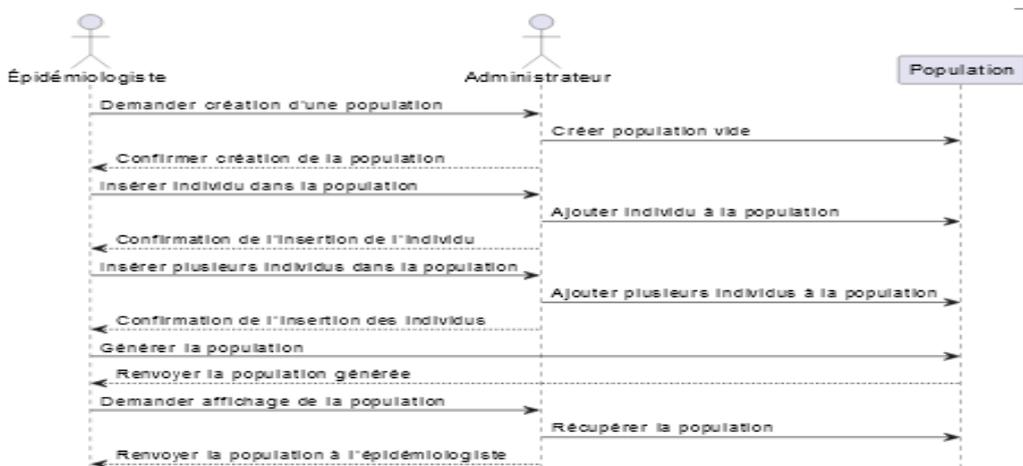


Figure 20 : Diagramme de séquences de génération de la population

2.8.1.3. Diagramme de classe

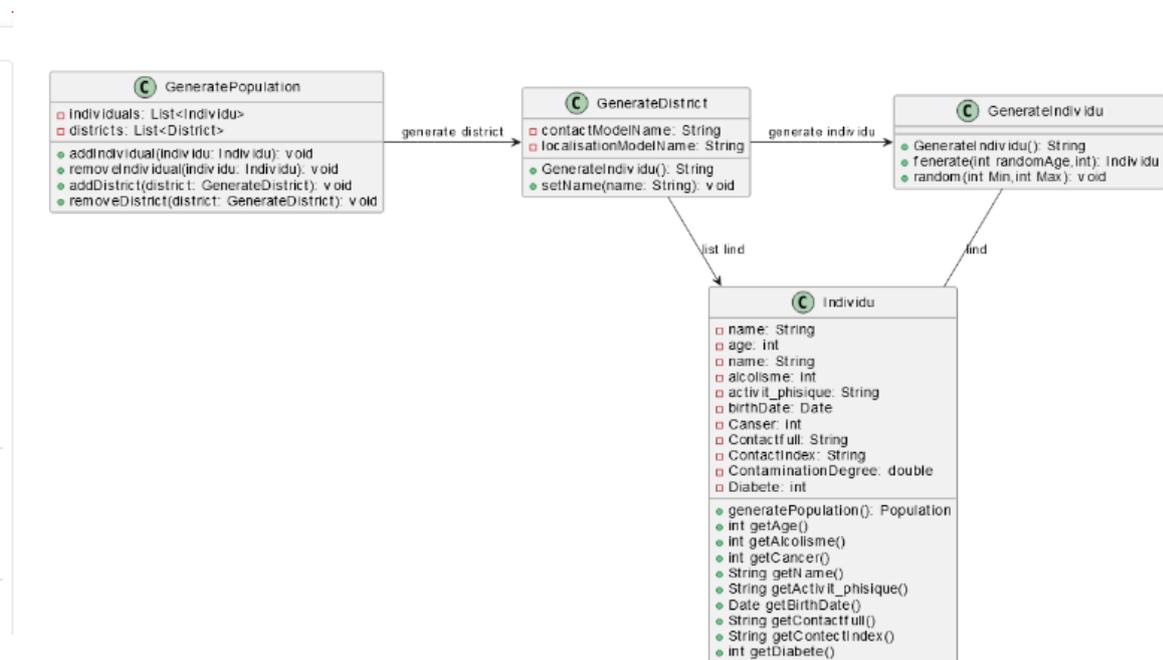


Figure 21 : Diagramme de classe

2.9. Conclusion

Ce chapitre a été dédié à la conception et la modélisation de la solution qu'on a proposée, les diagrammes UML permet de bien comprendre les fonctionnalités du système d'information de notre projet. La phase de l'implémentation concrétisera ce qui a été présenté et fera l'objet du chapitre prochain.

Chapitre 3:

Implémentation

3.1.Introduction

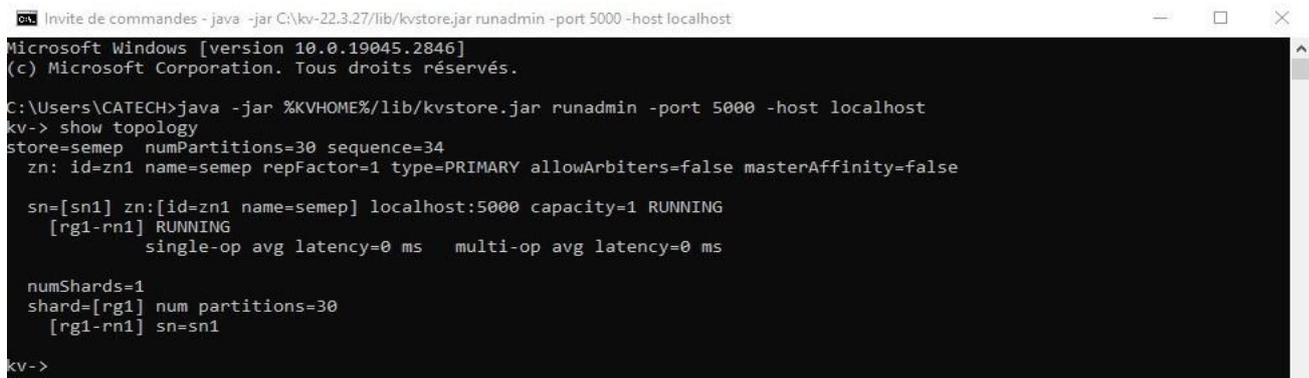
Ce chapitre présente la partie applicative de notre projet. Nous allons démontrer l'implémentation de notre projet dont nous allons présenter les outils employés et leur choix.

Par conséquent, ce chapitre donne les détails de la configuration des stores NoSQL en vue d'héberger les graphes RDF. Ensuite, il montre l'implémentation du modèle DL retenu comme stratégie de propagation/immunisation.

3.2.Les langages de programmation utilisés

Dans notre application, nous avons décidé d'utiliser deux langages de programmation : Java et Python. En utilisant ces langages dans notre application, nous bénéficions des avantages spécifiques offerts par ces deux langages, Java en termes de traitement des requêtes SPARQL, de manipulation des graphes RDF, de développement web et d'interactions avec les stores NoSQL. Dans notre cas spécifique, Python est choisi en raison de la popularité de Flask et de sa facilité d'utilisation, tandis que Java est utilisée pour Jena, qui est l'API spécifique requise pour la manipulation de graphes RDF. La passerelle entre Python et Java est nécessaire pour faciliter la communication entre ces deux parties de notre architecture d'application.

3.3.La configuration des stores NOSQL



```
Invite de commandes - java -jar C:\kv-22.3.27\lib\kvstore.jar runadmin -port 5000 -host localhost
Microsoft Windows [version 10.0.19045.2846]
(c) Microsoft Corporation. Tous droits réservés.

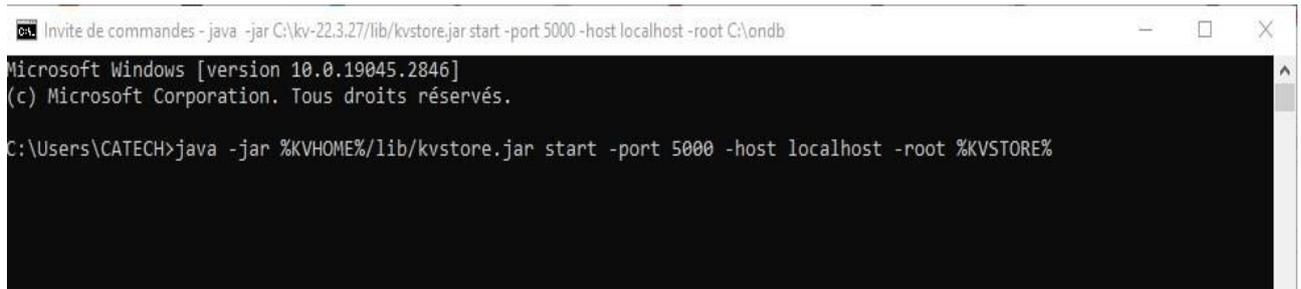
C:\Users\CATECH>java -jar %KVHOME%/lib/kvstore.jar runadmin -port 5000 -host localhost
kv-> show topology
store=semep numPartitions=30 sequence=34
zn: id=zn1 name=semep repFactor=1 type=PRIMARY allowArbiters=false masterAffinity=false

sn=[sn1] zn:[id=zn1 name=semep] localhost:5000 capacity=1 RUNNING
[rg1-rn1] RUNNING
single-op avg latency=0 ms multi-op avg latency=0 ms

numShards=1
shard=[rg1] num partitions=30
[rg1-rn1] sn=sn1

kv->
```

Figure 22 : Topologie déployer après configuration de store



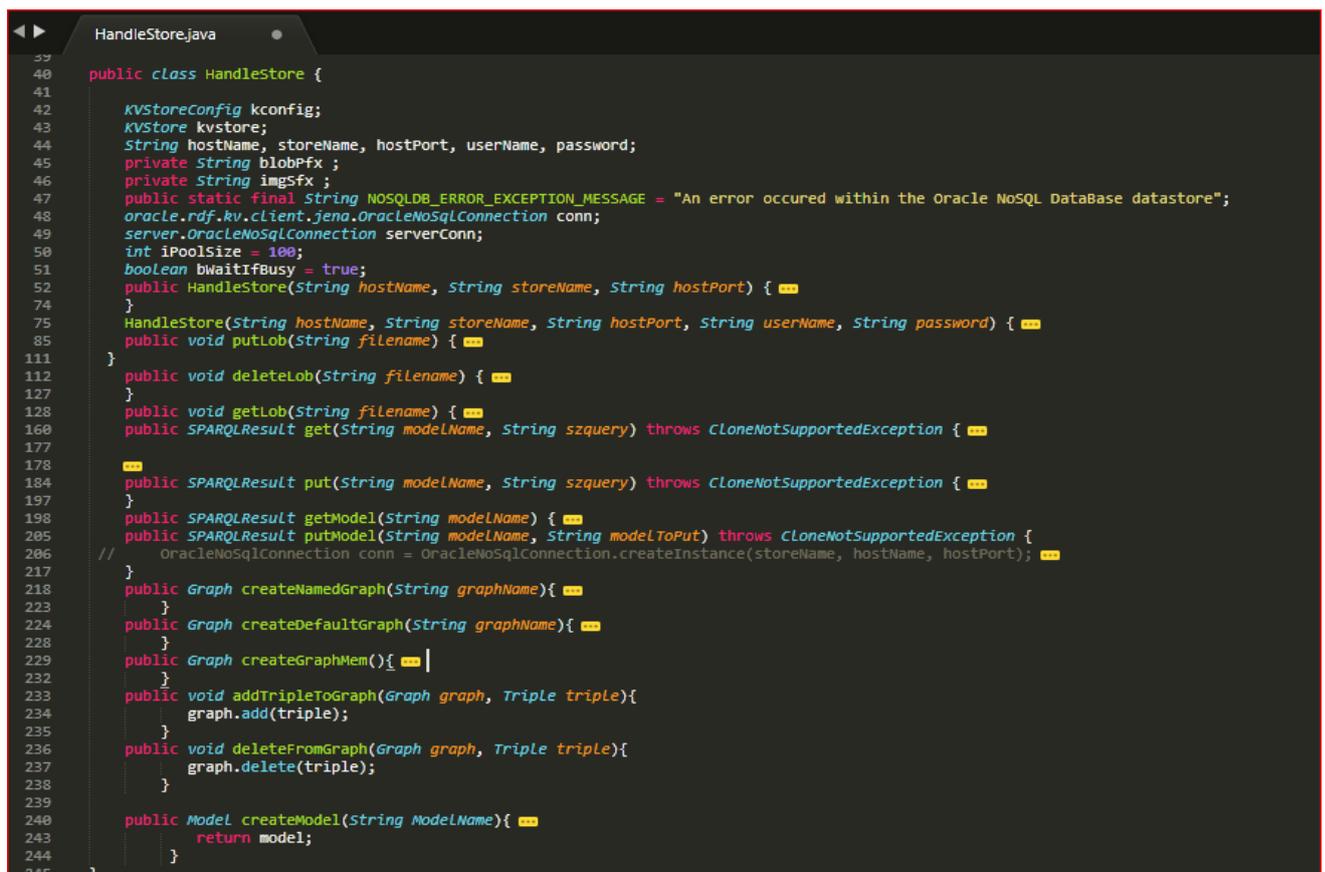
```
CL Invite de commandes - java -jar C:\kv-22.3.27\lib\kvstore.jar start -port 5000 -host localhost -root C:\ondb
Microsoft Windows [version 10.0.19045.2846]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\CATECH>java -jar %KVHOME%/lib/kvstore.jar start -port 5000 -host localhost -root %KVSTORE%
```

Figure 23 : Lancement du store

3.4.L'accès au store NoSQL via Apache Jena API (Java)

L'API Jena nous permettrons de créer des requêtes dynamiques, d'exécuter des opérations complexes sur les données RDF et de manipuler les résultats de manière flexible. On a créé la classe "HandleStore" pour responsabilité principale la gestion des opérations sur les graphes RDF, y compris la création du graphe, l'insertion et la suppression de triples RDF, ainsi que l'exploration du graphe par sujet, par prédicat et par objet.



```
HandleStore.java
39
40 public class HandleStore {
41
42     KVStoreConfig kconfig;
43     KVStore kvstore;
44     String hostName, storeName, hostPort, userName, password;
45     private String blobPfx ;
46     private String imgSfx ;
47     public static final String NOSQLDB_ERROR_EXCEPTION_MESSAGE = "An error ocured within the Oracle NoSQL DataBase datastore";
48     oracle.rdf.kv.client.jena.OracleNoSqlConnection conn;
49     server.OracleNoSqlConnection serverConn;
50     int iPoolSize = 100;
51     boolean bWaitIfBusy = true;
52     public HandleStore(String hostName, String storeName, String hostPort) { ...
74     }
75     HandleStore(String hostName, String storeName, String hostPort, String userName, String password) { ...
85     public void putLob(String filename) { ...
111     }
112     public void deleteLob(String filename) { ...
127     }
128     public void getLob(String filename) { ...
160     public SPARQLResult get(String modelName, String szquery) throws CloneNotSupportedException { ...
177     }
178     ...
184     public SPARQLResult put(String modelName, String szquery) throws CloneNotSupportedException { ...
197     }
198     public SPARQLResult getModel(String modelName) { ...
205     public SPARQLResult putModel(String modelName, String modelToPut) throws CloneNotSupportedException {
206     // OracleNoSqlConnection conn = OracleNoSqlConnection.createInstance(storeName, hostName, hostPort); ...
217     }
218     public Graph createNamedGraph(String graphName){ ...
223     }
224     public Graph createDefaultGraph(String graphName){ ...
228     }
229     public Graph createGraphMem(){ ... |
232     }
233     public void addTripleToGraph(Graph graph, Triple triple){
234     graph.add(triple);
235     }
236     public void deleteFromGraph(Graph graph, Triple triple){
237     graph.delete(triple);
238     }
239     }
240     public Model createModel(String modelName){ ...
243     return model;
244     }
245 }
```

Figure 24 : La classe "HandleStore"

3.5. Gateway Server

```
C:\HandleStoreByPython\Server\deploy>
C:\HandleStoreByPython\Server\deploy>java -jar GatewayServer.jar
Host Name :
localhost
Host Port :
5000
endpoint :localhost:5000
Store Name :
semep
semep
entry point of store localhost at semep:5000
semep
localhost:5000
accessing gateway
Gateway Server Started
```

Figure 25 : Lancement du Gateway Server

La figure 25 ci-dessus présente le lancement du Gateway Server.

3.6. Flask server

```
Invite de commandes - python FlaskServer.py
Microsoft Windows [version 10.0.19045.2846]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\CATECH>cd C:\Application\PFE_Python\WebNode

C:\Application\PFE_Python\WebNode>python FlaskServer.py
* Serving Flask app 'FlaskServer'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:8000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 165-371-026
127.0.0.1 - - [05/Jun/2023 22:06:13] "GET /semep/contacts/ HTTP/1.1" 200 -
127.0.0.1 - - [05/Jun/2023 22:06:14] "GET /favicon.ico HTTP/1.1" 404 -
```

Figure 26 : Lancement du Flask Server

La figure 26 ci-dessus présente le lancement du Flask Server. On situe ci-dessous l’invocation sur le graphe d’un individu en utilisant son index dans le lien.

```
test=====u:Amini
  size = 24
u:hasImmunityDegree ---->u:0.7
u:hasStress_chronique ---->u:1
u:hasLastName ---->u:Amini
u:hasContaminationDegree ---->u:0.2
u:hasSex ---->u:M
u:hasAge ---->u:27
u:hasFirstName ---->u:Amina
u:hasDiabete ---->u:0
u:hasBirthDate ---->u:1996-02-01
u:hasNiveau_Social ---->u:0.2
u:hasState ---->u:S
u:hasAddress ---->u:rue Amine Amini n°1 oran
u:hasPhone ---->u:041424225
u:hasPoids ---->u:65
u:hasCancer ---->u:0
u:hasMalnutrition ---->u:0.9
u:hasItinerance ---->u:0
u:hasTabagisme ---->u:1
u:hasTrait_corticoide ---->u:0
u:hasActivite_physique ---->u:1
u:hasAlcoolisme ---->u:1
u:hasMail ---->u:amina.amini@gmail.com
u:hasTuber_preced ---->u:1
u:hasUDI ---->u:1
```

Figure 27 : Information d'un individu

```
subject of findGraphBySubject PositionIndex

size of produced graph : 3
<u:position_2
<u:position_1
<u:position_3
/***** list of contact index position *****/
  trying to find All index by subject and Predicate from collectorJava Birth_270000004 and hasPosition
<u:position_2
<u:position_1
<u:position_3
/***** list of position by contact index*****/
```

Figure 28 : Les positions d'un contact

La figure 28 ci-dessous présente les positions d'un contact.

```
subject of findGraphBySubject position_2

size of produced graph : 5
size = 24
u:hasEndTime ---->u:2021-01-1 9:00:00
u:hasStartTime ---->u:2021-01-1 08:00:00
u:hasLatitude ---->u:0.9
u:hasLongitude ---->u:35.2
u:hasAddress ---->u:ici
Process exited with exit code 0.
```

Figure 29 : Information sur une position « 2 » visitée par un individu

La figure 29 ci-dessous présente les détails de la visite d'un individu à une position donnée.

3.7. Conclusion

Ce chapitre constitue la mise en œuvre principale que nous avons réalisée. Nous avons offert une vision détaillée de la configuration des stores NoSQL afin d'héberger les graphes RDF, nous avons renforcé notre travail en incluant des captures d'écran illustrant l'exécution de l'application.

Conclusion Générale

Ce projet nous a permis de développer une application de modélisation de réseaux complexes, en particulier un réseau épidémiologique pour une population importante. Pour cela, nous avons utilisé des ressources matérielles et logicielles essentielles, nécessitant une distribution multi-tiers sur le web et des serveurs d'application puissants.

Nous avons proposé une stratégie d'immunisation basée sur la construction d'un KG épidémiologique. Nous avons implémenté le KG Epidémiologique.

En ce qui concerne les données, nous avons utilisé le store NoSQL d'Oracle, en déployant différents clusters et nœuds de stockage sur plusieurs machines virtuelles. Nous avons choisi le modèle de données de graphe, qui est le plus adapté pour modéliser les réseaux complexes. Nous avons émis des requêtes sur des graphes RDF, qui sont maintenus dynamiquement en tant que réseaux sémantiques temporels.

En résumé, ce projet nous a offert l'opportunité de développer une application de modélisation de réseaux complexes en utilisant une combinaison de technologies telles que Python, Flask, PY4J, Jena, SPARQL et les stores NoSQL. Nous avons également mis en évidence l'importance de la visualisation graphique pour comprendre ces réseaux multicouches.

Bibliographie

- [1] Reuven Cohen, Bar Ilan and Shlomo Havlin, (2010), « Complex Networks: Structure, Robustness and Function », United States of America by Cambridge University Press, New York.
- [2] Tiffany J. Callahan, Ignacio J. Tripodi, Harrison Pielke-Lombardo and Lawrence E. Hunter, (2020), « Knowledge-Based Biomedical Data Science », Article (PMC), National Center for Biotechnology Information.
- [3] Vincent Levorato, (2008), « Contributions à la Modélisation des Réseaux Complexes: Prétopologie et Applications », Université Paris VIII Vincennes-Saint Denis.
- [4] Newman, M. E. J, (2003), « The structure and function of complex networks », Article Society for Industrial and Applied Mathematics, pp.1095-7200.
- [5] Nicholas A. Christakis and James H. Fowler, (2013), « Social Contagion Theory: Examining Dynamic Social Networks and Human Behavior », Statistics in Medicine 32(4), 10.1002/sim.5408.
- [6] Vincent Picard, (2010), « Graphes aléatoires ».
- [7] E. N. Gilbert, (1959), « Random Graphs », Mathematical Statistics, Vol 30, pp. 1141-1144.
- [8] Duncan J. Watts and Steven H. Strogatz, (1998), « Collective dynamics of 'small-world' networks », Article, Volume: 393, Nature Macmillan Publishers Ltd.
- [9] Dieter Fensel, Umutcan Şimşek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich and Alexander Wahler, (2020), « Introduction: What Is a Knowledge Graph », Knowledge Graphs pp 1–10.
- [10] Natasha Noy, Yuqing GAO, Anshu Jain, Anant Narayanan, Alan Patterson and Jamie Taylor, (2019), « Industry-scale Knowledge Graphs: Lessons and Challenges », Communications of the ACM, vol. 62 (8) (2019), pp. 36-43.
- [11] Vinay K. Chaudhri, Chaitanya Baru, Naren Chittar, Xin Luna Dong, Michael Genesereth, James Hendler, Aditya Kalyanpur, Douglas B. Lenat, Juan Sequeda, Denny Vrandečić and Kuansan Wang, (2022), « Knowledge graphs: Introduction, history, and perspectives », NSF CONVERGENCE ACCELERATOR pages: 17-29.
- [12] Jan Portisch, Michael Hladik and Heiko Paulheim, (2021), « FinMatcher at FinSim-2: Hypernym Detection in the Financial Services Domain using Knowledge Graphs ».

Bibliographie

- [13] Wei Meng, Dongning Zhang, Tengxuan Guo, Zhenguo Zong, Yijuan Liu, Yanmei Wang, Jing Li and Weiyi Zhu, (2021), « Design and Implementation of Knowledge Graph Platform of Power Marketing », International Conference on Computer Engineering and Application (ICCEA 2021).
- [14] Sören Auer EMAIL logo , Allard Oelen , Muhammad Haris , Markus Stocker , Jennifer D'Souza , Kheir Eddine Farfar , Lars Vogt , Manuel Prinz , Vitalis Wiens and Mohamad Yaser Jaradeh,(2020), « Improving Access to Scientific Literature with Knowledge Graphs »,Journal Bibliothek Forschung und Praxis.
- [15] Lei Wang, Huimin Xie, Wentao Han, Xiao Yang, Lili Shi, Jiancheng Dong, Kui Jiang and Huiqun Wu, (2020), « Construction of a knowledge graph for diabetes complications from expert-reviewed clinical evidences », Article Computer Assisted Surgery (IF 2.357), Department of Medical Informatics, Medical School of Nantong University.
- [16] Payal Chandak, Kexin Huang and Marinka Zitnik, (2023), « Building a knowledge graph to enable precision medicine », Scientific Data, volume 10, Article number: 67, data descriptors.
- [17] Yating Zhao, Jingjing Guo, Chao Bao, Changyong Liang and Hemant K Jain, (2020), «Knowledge Graph Analysis of Human Health Research Related to Climate Change », Article PMC7599824.
- [18] Avishek Chatterjee, Cosimo Nardi, Cary Oberije and Philippe Lambin, (2021), « Knowledge Graphs for COVID-19: An Exploratory Review of the Current Landscape », Article (MDPI), Basel, Switzerland.
- [19] Enayat Rajabi and Somayeh Kafaie, (2022), « Knowledge Graphs and Explainable AI in Healthcare », pp [MDPI].13(10), 459.
- [20] Chongyan Chen, Islam Akef Ebeid, Yi Bu and Ying Ding, (2020), « Coronavirus Knowledge Graph: A Case Study ». In KDD2020: ACM Knowledge Discovery in Databases. ACM, New York, NY, USA.
- [21] Ying Shen, Yonghong Liu, Xiaokang Jiao, Yuxin Cai, Xiang Xu, Hui Yao and Xiaoli Wang (2021), « Knowledge Graph: Applications in Tracing the Source of Large-Scale Outbreak », Article : Methods and Applications, China CDC Weekly.
- [22] Marta Contreiras Silva, Patrícia Eugénio, Daniel Faria and Catia Pesquita, (2022), « Ontologies and Knowledge Graphs in Oncology Research », PMC9029532, Article (MDPI), Basel, Switzerland.

Bibliographie

[23] Hualing Liu, Yi Sun and Shijie Cao, (2022), « Building a COVID-19 Literature Knowledge Graph Based on PubMed », International Conference on Medical Imaging and Computer-Aided Diagnosis (MICAD 2021) (pp.364-373).

[24] Houcemeddine Turki, Dariusz Jemielniak, Mohamed A. Hadj Taieb, Jose E. Labra Gayo, Mohamed Ben Aouicha, Mus'ab Banat, Thomas Shafee, Eric Prud'hommeaux, Tiago Lubiana, Diptanshu Das and Daniel Mietchen, (2022), « Using logical constraints to validate statistical information about disease outbreaks in collaborative knowledge graphs: the case of COVID-19 epidemiology in Wikidata », Article PMC9575845, National Center for Biotechnology information.

[25] Zhihan Gao, Min Gao, Chun-hua Chen, Yifan Zhou, Zhi-Hui Zhan and Yuan Ren, (2023), « Knowledge graph of wastewater-based epidemiology development »: A data-driven analysis based on research topics and trends, Nature Public Health Emergency Collection, PMC9867605.

[26] Tian Chen, Yimu Zhang, Xinwu Qian and Jian Lia, (2022), « A knowledge graph-based method for epidemic contact tracing in public transportation », Elsevier Public Health Emergency Collection, PMC8818383.

[27] Maximilian Nickel, Kevin Murphy, Volker Tresp and Evgeniy Gabrilovich, (2015), « Knowledge Graph Completion: A Survey of Approaches and Evaluation Methods », cs.LG, arXiv: 1503.00759 [stat.ML].

[28] Qingyu Guo, Fuzhen Zhuang, Chuan Qin and Hengshu Zhu, (2020), « A Survey on Knowledge Graph-Based Recommender Systems », IEEE Transactions on Knowledge and Data Engineering, pp (99):1-1, 10.1109/TKDE.3028705.

[29] Mingyang Chen, Wen Zhang, Zonggang Yuan, Yantao Jia and Huajun Chen, (2022), « Federated knowledge graph completion via embedding-contrastive learning », Article, Knowledge-Based Systems, Volume 252, 109459.

[30] Xiangru Zhu, Zhixu Li, Xiaodan Wang, Xueyao Jiang, Penglei Sun, Xuwu Wang, Yanghua Xiao and Nicholas Jing Yuan, (2022), « Multi-Modal Knowledge Graph Construction and Application: A Survey », arXiv:2202.05786 [cs.AI].

[31] Ling Tian, Xue Zhou, Yan-Ping Wu, Wang-Tao Zhou, Jin-Hao Zhang and Tian-Shu Zhang, (2022), « Knowledge graph and knowledge reasoning: A systematic review », Article, Journal of Electronic Science and Technology, Volume 20, Issue 2, pp.100159.

Bibliographie

- [32] Haofen Wang, Huifang Du, Guilin Qi, Huajun Chen, Wei Hu and Zhuo Chen, (2022), « Construction of a Linked Data Set of COVID-19 Knowledge Graphs: Development and Applications », Article: PMC9109781, 10.2196/37215, Originally published in JMIR Medical Informatics.
- [33] Mostefa Mokaddem, (2019), « Convergence des issues de simulation et de Data Mining en épidémiologie », Thèse de doctorat, ASJP. Informatique. Université d'Oran 1 Ahmed Ben Bella, Algérie.
- [34] TOBAL Ali and Hamza Amar Setti, (2021), « OUTILS DE DATA MINING POUR LA DETECTION DE COMMUNAUTES EN EPIDEMIOLOGIE », Mémoire de Master, Département d'informatique, Faculté des sciences exactes et appliquées, Université d'Oran 1 Ahmed Ben Bella, Algérie.
- [35] Mostefa Mokaddem, Ilhem Idris Khodja, Hamza Amar Setti, Baghdad Atmani and Chihab Eddine Mokaddem, (2022), « Communities Detection in Epidemiology »: Evolutionary Algorithms Based Approaches Visualization, International Symposium on Modelling and Implementation of Complex Systems (MISC 2022).
- [36] Ameya Nayak, Anil Poriya, Dikshay Poojary, (2013), « International Journal of Applied Information Systems (IJAIS) »: ISSN : 2249-0868 Volume 5– No.4, March 2013 : Type of NOSQL Databases and its Comparison with Relational Databases
- [37] Vatika Sharma, Meenu Dave, (2012), «SQL and NoSQL Databases »: International Journal of Advanced Research in Computer Science and Software Engineering, volume 2, issue 8.
- [38] Oracle NoSQL Database Administrator's Guide, « <https://docs.oracle.com/database/nosql-12.1.3.2/AdminGuide/index.html> », Consulter le 10-04-2023.
- [39] Kunal Relan, (2019), « Building REST APIs with Flask ».