République Algérienne Démocratique et Populaire
Ministére de l'enseignement supérieur et de la recherche scientifique
UNIVERSITÉ LARBI TÉBESSI - TÉBESSA
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département des Mathématiques et Informatique

THÈSE

En vue de l'obtention du diplôme

DOCTORAT LMD

En Informatique

Spécialité : *Système d'information coopératif*

Présentée et soutenue publiquement le *28 / 07 /2019* **par :**

Sadek MENACEUR

L'entreposage de données et le Business Intelligence sur le Cloud Computing

---

JURY

Hakim BENDJENNA	Professeur, Université de Tébessa	Président
Nacira GHOUALMI	Professeur, Université de Annaba	Examinateur
Med Ridda LAOUAR	Professeur, Université de Tébessa	Examinateur
Djamel BENMERZOUG	MCA, Université de Constantine 2	Examinateur

---

Directeur de Thèse :
Dr. Makhlouf DERDOUR -Université de Tébessa-
Co-Directeur de Thèse :
Dr. Abdelkrim BOURAMOUL -Université de Constantine 2-
Laboratoire de Recherche :
Laboratory of Mathematics, Informatics and Systems (LAMIS)

L'entreposage de données et le Business Intelligence sur le Cloud Computing

Sadek MENACEUR
université larbi tébessi, tébessa
Laboratory of Mathematics, Informatics and Systems (LAMIS)

Juillet 2019

Remerciements

En tout premier lieu, je remercie le bon dieu, tout puissant, de m'avoir donné le courage, la force morale et physique, pour achever ce travail.

Cette thèse est le fruit de la participation d'un ensemble de personnes qui ont permis de près ou de loin que ces travaux de recherche aboutissent. Je souhaiterais adresser tous mes remerciements à tout ceux qui m'ont aidé pour réaliser ce mémoire.

Tout d'abord, je tiens à exprimer ma profonde gratitude à mon directeur de thèse Mr. Makhlouf DERDOUR pour toute l'aide qu'il a su m'apporter tout au long de mes quatre ans de recherche, ses précises conseils et orientations m'ont permis de mener à bien ce travail.

Je tiens à exprimer aussi ma profonde gratitude à mon co-directeur de thèse Mr. Abdelkrim BOURAMOUL pour son encadrement, ses conseils, sa grande disponibilité, qui ont été pour moi une grande source de motivation.

J'exprime ma profonde reconnaissance à Mr. Hakim BENDJENNA de me faire l'honneur de présider le jury et mes plus sincères remerciements à l'égard de Mme. Nacira GHOUALMI, Mr. Med Ridda LAOUAR et Mr. Djamel BENMERZOUG pour l'intérêt qu'ils ont bien voulu porter à ce travail en acceptant de l'examiner et d'en être rapporteurs.

Je n'oublie pas Mr. Nacif LABED Professeur à l'université de constantine, qui m'a porté beaucoup d'aide dans la révision et la finalisation de cette thèse. Je ne le remercierai jamais assez et j'en serai toujours reconnaissant. Je remercie tous les anciens et actuels doctorants de laboratoire LAMIS pour leur complicité et les bons moments partagés.

J'adresse tous mes remerciements à ma famille : mon père, mes sœurs, mon frère et particulièrement à ma mère qui a toujours cru en moi.

Je remercie de tout Cœur ma chère femme pour son soutien quotidien, sans oublier ma belle-famille, et bien sûr mes plus belles pensées vont à ma petite fille Assil que j'aime autant.

ملخص

يندرج العمل المنجز في هذه الرسالة ضمن ثلاث محاور بحث في مجال البيانات الكبيرة و هي: "تخزين المعطيات"، "استعمال السياق" و "ذكاء الأعمال". و يهدف إلى تقديم مساهمات على محورين متكاملين: أولاً، استغلال ملف تعريف المستخدم في إعادة صياغة الاستعلامات، ثم تخصيص التحليل في البيانات الكبيرة باستخدام الاستعلامات المعدلة والتصفية القائمة على المحتوى. في الواقع، إن عصر البيانات الكبيرة يجلب قيمًا وفوائد هائلة تتعلق بحياتنا اليومية، مثل التجارة الإلكترونية والسياحة والنقل، ... إلخ. ومع ذلك، يبقى تحليل هذه البيانات في ضل المناهج التقليدية لذكاء الأعمال محدودًا للغاية في مواجهة مشكلة ثلاثية الابعاد تشمل: حجم كبير من البيانات المراد معالجتها، تنوع في منابع البيانات (منظمة أو غير منظمة)، و مستوى معين من السرعة التي يتم بها إنشاء البيانات ومعالجتها. للتغلب على الاشكاليات الثلاث المذكورة آنفا، ولتسهيل صياغة متطلبات المستخدم، ولجعل المعلومات المحددة مفهومة وقابلة للاستغلال من قبل المستخدم، أدرجنا آليتين في مساهماتنا هما: أولا، ادراج السياق المتعلق بالمستخدم و المخزن في بنية تسمى "ملف تعريف المستخدم" و الذي يعتبر أساسيا في أي نظام يعتمد على تقنيات تخصيص البيانات. ثانيا، استعمال سياق البحث المستخلص من مصطلحات استعلام المستخدم. الغرض من الآليتين المذكورتين أعلاه هو زيادة انتقائية النتائج في مكعب OLAP المخصص باستخدام تقنية تصفية تستند إلى المحتوى. في الأخير، أجريت مجموعة من التجارب في هذه الرسالة، كان الهدف منها: أولاً، إثبات قابلية تطبيق مختلف الطرق المقترحة، ثانيا، مقارنة كل من مساهماتنا واختبارها والتحقق من صحتها.

الكلمات المفتاحية: تخزين المعطيات، سياق المستخدم، ذكاء الأعمال، البيانات الكبيرة.

Résumé

Cette thèse s'inscrit dans les domaines de l'entreposage, du contexte, et du Business Intelligence dans l'ère du Big Data, elle vise à apporter des contributions sur deux axes complémentaires : d'abord l'exploitation du profil utilisateur dans la reformulation de requêtes, puis la personnalisation de l'analyse dans le Big Data par l'utilisation de requêtes reformulées et le filtrage basé sur le contenu. En effet, l'ère du Big Data apporte des valeurs et des avantages énormes liés à notre vie quotidienne, tels que le commerce électronique, le tourisme, et les transports, etc. Cependant, l'analyse de ces données par les approches traditionnelles de Business Intelligence reste très limitée face à une triple problématique qui couvre : un volume de données important à traiter, une grande variété d'informations (structurées ou non-structurées), et un certain niveau de vélocité à atteindre. Pour pallier à cette triple problématique, et afin de faciliter l'expression du besoin utilisateur, et de rendre l'information sélectionnée intelligible et exploitable par l'utilisateur, nous utilisons deux mécanismes dans nos contributions : d'une part, le contexte relatif à l'utilisateur stocké dans une structure nommée "profil utilisateur" qui est considéré central dans tout système basé sur les techniques de personnalisation, d'une autre part, le contexte de la recherche portée par les termes de la requête utilisateur. Le but des deux mécanismes cités précédemment est d'augmenter la sélectivité des résultats dans le cube OLAP personnalisé par l'utilisation d'une technique de filtrage à base de contenu. Enfin, un ensemble d?expérimentations ont été réalisées dans cette thèse. L'objectif de ces expérimentations était double : d'abord, prouver l'applicabilité des différentes approches proposées, puis comparer, tester et valider chacune de nos contributions.

Mots-clés: Entreposage, contexte utilisateur, Business Intelligence, Big Data

Abstract

This dissertation is part of the fields of warehousing, context and Business Intelligence in the era of Big Data. It aims at making contributions on two complementary axes: first, the exploitation of the user profile in the reformulating queries, then personalizing analysis in Big Data by using reformulated queries and content-based filtering technique. Indeed, the era of Big Data brings remarkable values and benefits related to our daily life such as e-commerce, tourism, transportation, etc. However, the analysis of these data by traditional Business Intelligence approaches remains very limited in the face of a triple problem that covers: a large volume of data to be processed, a wide variety of information (structured or unstructured) and a certain level of velocity to reach. To solve this triple problem, and in order to facilitate the expression of the user's need, and to make the selected information intelligible and ready for use by the user, we use two mechanisms in our contributions : on the one hand, it is the context relative to the user stored in a structure called ?user profile? which is considered central in any system based on personalization techniques. On the other hand, it is the context of the search carried by the terms of the user request. The purpose of the two mechanisms mentioned above is to increase the selectivity of results in the custom OLAP cube by using a content-based filtering technique. Finally, a set of experiments were carried out in this thesis. The aim of these experiments was twofold: first, to prove the applicability of the different proposed approaches, then to compare, test and validate each of our contributions.

Keywords: warehousing, user context, Business Intelligence, Big Data

Table des Matières

Liste des figures

Liste des tableaux

Introduction générale

1 Contexte

Nous vivons à l'ère du déluge de données, où de grandes quantités de données sont créées tous les jours à partir des données utilisateur générées automatiquement sur Internet telles que réseaux sociaux, appareils mobiles, messagerie électronique, blogs, vidéos, transactions bancaires, etc. Tout cela amène à l'établissant d'une nouvelle dimension appelée Big Data [Oussous et al., 2018].

Aujourd'hui, l'ère du Big Data apporte des valeurs et des avantages énormes, non seulement dans les domaines scientifiques tels que l'astronomie ou la biologie, mais également dans des domaines étroitement liés à notre vie quotidienne, tels que le commerce électronique, le tourisme, et les transports, par exemple. Le Big Data introduit une nouvelle culture dans les entreprises, il propose des innovations dans les techniques de stockage et de traitement des données, cela est dû au volume et à la nature des données manipulées qui sont collectées depuis de différentes sources [Lee, 2017].

Par ailleurs, les entreprises utilisent ces données massives pour appuyer leurs décisions, découvrir les besoins de leurs clients et créer de nouveaux modèles commerciaux. Cependant, le traitement de ces données par les approches traditionnelles reste très limité face à une triple problématique qui couvre : un volume de données important à traiter, une grande variété d'informations (structurées ou non structurées), et un certain niveau de vélocité à atteindre [Kaur and Bharti, 2019].

À l'ère technologique actuelle, effectuer des tâches analytiques sur le Big Data nécessite des méthodes spécifiques de stockage, de filtrage, de transformation et de récupération des données. Les systèmes de Business Intelligence font référence aux technologies et aux outils chargés de la collecte, du stockage et de l'analyse des données pour améliorer la prise de décision. Dans les systèmes de Business Intelligence, les utilisateurs interagissent avec l'entrepôt de données en formulant et en lançant des séquences de requêtes visant à explorer des cubes de données multidimensionnels [Lara Pahins et al., 2019].

Cependant, les volumes de données stockés dans un entrepôt de données peuvent être très importants et diversifiés. Ainsi, une grande quantité d'informations non pertinentes renvoyées sous forme de résultats à l'utilisateur pourrait rendre le processus d'exploration des données inefficace. Il devient de plus en plus difficile pour les utilisateurs de retrouver

précisément ce qu'ils recherchent dans ces grandes masses de données. C'est pourquoi il est nécessaire d'aider l'utilisateur en le guidant dans son exploration et analyse de données.

En revanche, certaines approches dans l'analyse et l'exploration de données utilisent la personnalisation et la recommandation de requêtes comme des mécanismes pour augmenter la performance et la pertinence des résultats obtenus après une session d'analyse [Gorrab et al., 2019]. En effet, les systèmes de recommandation de requêtes jouent un rôle majeur dans la réduction des efforts des décideurs pour trouver les informations les plus intéressantes. La reformulation ou l'expansion de requête est l'une des techniques de la personnalisation des données [Gorrab et al., 2019], elle se base sur le principe que l'utilisateur n'est souvent pas capable de formuler ses besoins en informations et consiste à l'aider dans sa formulation de requête en ajoutant des termes de la requête à partir des autres sources telles que WordNet, par exemple. Une autre techniques développées est le filtrage d'informations. Cette technique consiste à concevoir des mécanismes qui permettent de faciliter la tâche de l'analyse à l'utilisateur, en lui faisant parvenir continuellement l'information qui l'intéresse selon son profil. Cependant, le problème de capture des paramètres du profil reste encore posé. L'utilisateur devrait être représenté par un modèle général regroupant l'ensemble des dimensions informationnelles le caractérisant.

Le travail présenté dans cette thèse est orienté vers un rapprochement technologique, qui consiste à intégrer les techniques de la personnalisation et le filtrage des informations dans l'analyse de Big Data, afin d'offrir une solution de Business Intelligence plus performante par une analyse multidimensionnelle OLAP personnalisée, basée sur le profil de l'utilisateur et le contexte de sa requête de recherche. Les utilisateurs dans ce cas-là ont la possibilité d'accéder rapidement aux données les plus pertinentes pour lui.

2 Problématique

Le rapprochement entre les techniques dont nous avons parlé précédemment permet d'obtenir un concept personnalisable, facilitant la création d'une analyse multidimensionnelle personnalisée basée sur le profil de l'utilisateur et le contexte de sa requête de recherche. Il s'agit donc d'un traitement Big Data où on intègre les techniques de personnalisation et de recommandation de requêtes afin de construire des cubes OLAP personnalisés. Les technologies des systèmes Business Intelligence permettent de rendre accessible l'analyse multidimensionnelle OLAP de données et d'accélèrer l'accès aux données pertinentes.

Dans ce contexte, plusieurs questions se posent au sujet de l'amélioration du processus d'analyse OLAP dans le contexte de Big Data, et de la manière dont les résultats retournés

sont personnalisés selon le contexte de l'utilisateur. Les problématiques auxquelles nous cherchons à trouver des solutions dans le cadre de cette thèse sont :

1. Comment peut-on intégrer le profil de l'utilisateur et le contexte de recherche pour la reformulation des requêtes utilisateur ?
2. Comment peut-on améliorer le processus de l'analyse en ligne OLAP par la prise en compte des requêtes reformulées pour la personnalisation afin de réduire les efforts des décideurs pour trouver les informations les plus intéressantes depuis un ensemble de données massive (Big Data).

C'est sur la base de ces interrogations que nous avons construit les contributions de cette thèse. Ceci est fait à partir d'un état de l'art des usages en cours, des travaux académiques sur la reformulation et l'enrichissement des requêtes utilisateurs par l'utilisation du contexte de son profil ainsi que le contexte de sa requête de recherche, et la personnalisation de l'analyse en ligne OLAP dans un contexte Big Data.

3 Contribution

Afin d'assurer une continuité dans l'enchaînement de nos contributions et procurer ainsi une meilleure couverture des objectifs de cette thèse, nous avons réparti nos propositions sur deux axes :

Un premier axe relatif à la reformulation et l'enrichissement des requêtes utilisateur en se basant sur l'utilisation du profil et le contexte de recherche afin de personnaliser l'analyse OLAP, une contribution est proposée dans ce sens Menaceur et al. [2017a], et qui consiste à proposer une nouvelle approche, qui met en œuvre une technique d'analyse et de personnalisation qui soit capable d'effectuer des opérations analytiques multidimensionnelles rentables sur des données volumineuses structurées. L'architecture du système est basée principalement sur les techniques de personnalisation citées dans le deuxième chapitre, en intégrant le contexte de la requête de recherche utilisateur et les éléments contextuels stockés dans son profil. Un espace de données réduit a été construit et orienté vers la génération des cubes OLAP personnalisés relatifs au besoin contextuel et au profil de l'utilisateur.

Un deuxième axe concerne cette fois la personnalisation et l'analyse multidimensionnelle des données volumineuses non structurées telles que; les recommandations des clients en ligne. La contribution proposée repose principalement sur deux éléments complémentaires Menaceur et al. [2019] : (i) les techniques d'expansion de requêtes et (ii) le filtrage

des informations. Dans un premier temps, nous utilisons les éléments de préférence de l'utilisateur stockés dans son profil pour réduire l'espace de recherche dans le cube OLAP afin d'extraire les meilleur données "Good Data". Dans la deuxième étape, nous incorporons des mots équivalents (synonymes) pour tout ou partie des mots de requête d'origine d'un utilisateur afin d'enrichir la requête, ce qui permet de personnaliser la recherche dans le cube OLAP réduit afin d'extraire les données les plus utiles "useful data".

Enfin, un ensemble d'expérimentations a été réalisé dans cette thèse. L'objectif de ces expérimentations était double : d'abord, prouver l'applicabilité des différentes approches proposées, puis comparer, tester chacune de nos contributions.

4 Organisation de la thèse

Les chapitres composant cette thèse sont organisés en deux grandes parties; La première partie est un état de l'art présentant respectivement : les domaines dans lesquels la problématique de thèse est posée (Business Intelligence et Big Data), les outils que nous avons utilisé comme support d'analyse multidimensionnelle dans nos contribution (OLAP et filtrage à base de contenus) et enfin, les techniques de personnalisation selon lesquelles les données massives des cubes OLAP sont personnalisées. Dans la deuxième partie, nous présentons en détail nos contributions.

Plus précisément les chapitres de cette thèse se présentent comme suit : Dans la première partie nous avons abordé l'état de l'art, cette partie comprend deux chapitres:

Le premier chapitre présente les concepts de base sur la Business Intelligence et le Big Data. Nous commençons dans la section une par donner une définition de la Business Intelligence, ensuite nous discutons ses différents objectifs. Nous décrivons par la suite les différents composants clés d'une architecture Business Intelligence traditionnelle, ainsi que l'impact du Big Data dans le processus de Business Intelligence. Nous finalisons cette section par la présentation d'OLAP comme outil de Business Intelligence. La deuxième section introduira le concept Big Data en détail; sa définition, ses caractéristiques, ses défis et ses technologies. Nous discutons dans la section qui suit le rapprochement Business Intelligence et le concept Big Data. Nous exposons dans la cinquième section l'entreposage de données dans l'ère du Big Data et les différentes approches utilisées dans la littérature pour une analyse multidimensionnelle OLAP. Nous finalisons ce chapitre par une sixième section consacrée à synthétiser les différents types de Business Analytics utilisés pour les systèmes Business Intelligence et les tendances de la recherche Big Data et Business Intelligence.

Le deuxième chapitre aborde le concept de la profilisation et l'accès personnalisé aux données massives. Nous y présentons en premier lieu les concepts clés de la profilisation, la contextualisation et la personnalisation des données massives en utilisant le profil utilisateur, ainsi que les techniques d'expansion des requêtes. Dans un deuxième lieu nous exposons les techniques de filtrage d'informations dans les systèmes de recommandation, où un état de l'art présente les différentes approches disponibles dans la littérature.

Dans une deuxième partie, nous avons mis en relief nos contributions. Cette partie comprend deux chapitres :

Le troisième chapitre présente notre contribution Menaceur et al. [2017a], qui consiste à proposer une nouvelle approche qui permet à mettre en œuvre une technique d'analyse et de personnalisation capable d'effectuer des opérations analytiques multidimensionnelles rentables sur des données volumineuses. L'architecture du système est basée principalement sur les techniques de personnalisation citées dans le deuxième chapitre en intégrant le contexte de la requête de recherche utilisateur et les éléments contextuels stockés dans son profil. Un espace de données réduit a été construit et orienté vers la génération des cubes OLAP personnalisés relatifs au besoin contextuel et au profil de l'utilisateur.

Le quatrième chapitre présente une deuxième contribution qui traite le problème de la personnalisation dynamique dans un contexte de Big Data, en utilisant une approche d'analyse multidimensionnelle. Cette approche utilise la technique d'expansion de requêtes et le filtrage basé sur le contenu pour personnaliser et filtrer les informations. Notre proposition consiste en première étape à traiter les requêtes initiales des utilisateurs par une technique d'enrichissement, afin d'intégrer les éléments contextuels du profil et contexte de recherche dans le contexte de la requête initiale de l'utilisateur, dans le but de réduire l'espace de recherche dans le cube multidimensionnel OLAP, ensuite d'utiliser la technique d'expansion des requêtes comme deuxième étape pour étendre la portée de l'analyse dans le cube multidimensionnel. Les résultats obtenus seront : "aussi pertinents que possible" par rapport à la demande initiale de l'utilisateur. Par ailleurs, nous utilisons les techniques de filtrage d'informations telles que le filtrage basé sur le contenu pour personnaliser l'analyse dans le cube de données réduites en fonction de la fréquence des termes et de la similarité des cosinus. Une étude expérimentale selon une étude de cas est présentée pour évaluer et valider notre contribution.

Une conclusion générale du travail fait durant cette thèse est présentée à la fin du manuscrit. Elle résume les points essentiels du travail réalisé et présente quelques perspectives de recherche suggérées par le bilan de ce travail.

Part I

Etat de l'Art

Chapter 1

Big Data et Business Intelligence

1 Introduction

L'entreposage des données (Data Warehouse) et les systèmes d'analyse en ligne OLAP sont des technologies de l'informatique décisionnelle ou Business Intelligence (BI en anglais) destinées à faciliter le processus de prise de décision au moyen d'une analyse en ligne et multidimensionnelle dans de grands volumes de données (Big Data). La révolution technologique conduit les entreprises à des confrontations vis-à-vis le choix des environnements, qui ont de plus en plus complexes et compétitifs, et dans lesquels le pilotage implique des choix qui doivent être fait dans des temps très courts, tout en prenant en compte un volume de données toujours plus important.

La notion de Business Intelligence englobe des solutions informatiques dont le but est de consolider les données disponibles au sein d'un entrepôt de données de l'entreprise, et de les analyser d'une manière significative [Kimble and Milolidakis, 2015] pour prendre des décisions éclairées. Selene Xia and Gong [2014], Kowalczyk and Buxmann [2014] ont vu que le Business Intelligence pourrait jouer un rôle important dans l'amélioration de la performance organisationnelle, en identifiant de nouvelles opportunités, en soulignant les menaces potentielles, en révélant de nouvelles perspectives commerciales et en améliorant les processus de prise de décision.

Aujourd'hui, l'avènement des nouvelles technologies informatiques et Internet conduit à une explosion dans le volume de données (2,5 trillions d'octets de données générés chaque jour), il n'y aurait que 20% de données structurées en entreprise, et les 80% restant constituent des données hétérogènes complexes et simples provenant de sources multiples, ceci pose de nouveaux défis et ouvre de nouvelles opportunités pour le Business Intelligence.

2 Business Intelligence

Malgré l'existence de nombreux systèmes modernes d'analyse et de traitement de données à grande échelle, la Business Intelligence reste le processus le plus utile pour l'analyse et la visualisation des données.

2.1 Définitions

Dans les références académiques, plusieurs définition ont été affectées au terme BI y compris : Gupta et al. [2015] et Hu et al. [2014] ont décrit la BI comme un terme générique pour un ensemble d'applications, de technologies et de processus utilisées pour la collecte, le stockage, la récupération et l'analyse de données afin d'aider les utilisateurs professionnels à prendre des décisions éclairées. Marjanovic [2013, 2015] dans ces deux références incluait des architectures et des méthodologies dans leurs définitions et décrivait la Business Intelligence comme un ensemble d'applications, de technologies, d'architectures, de processus et de méthodologies utilisés pour la collecte, le stockage, la récupération et l'analyse de données. Par contre Debortoli et al. [2014] ont appelé Business Intelligence comme un ensemble de fonctionnalités et ont donné comme exemples de ses fonctionnalités l'extraction, la transformation, le chargement (ETL), le traitement analytique en ligne (OLAP[1]) et l'entreposage de données. Fekete and Vossen [2015] ont décrit la BI comme l'intégration des composants de données, de stratégie, de processus et d'analyse d'une organisation pour soutenir la prise de décision.

À la lumière de ce qui précède, nous voyons que la BI est un ensemble de stratégies intégrées, d'applications, de technologies, d'architectures, de processus et de méthodologies utilisés pour collecter, stocker, récupérer et analyser des données afin de faciliter la prise de décision.

2.2 Objectifs de Business Intelligence

La Business Intelligence n'est plus perçue comme un outil d'aide au système d'aide à la décision, mais bien comme un outil permettant d'améliorer divers processus tels que le service client ou la fabrication. Le développement de la Business Intelligence a fourni aux organisations et aux décideurs la capacité d'accéder à des données pertinentes et de les analyser pour prendre des décisions éclairées et réalistes [Elbashir et al., 2008].

La mise en œuvre de la Business Intelligence a plusieurs objectifs, tels que l'amélioration du processus de prise de décision, l'efficacité des opérations, l'augmentation de la satisfac-

[1]Online Analytical Processing.

tion des clients et la création d'un avantage concurrentiel sur le marché actuel. Récemment, un autre objectif majeur des outils de Business Intelligence est celui de contrôler les énormes flux de données provenant de sources multiples en les rassemblant dans un modèle condensé uniforme pour identifier les lacunes et les opportunités pour les organisations.

2.3 Architecture traditionnelle de Business Intelligence

Afin d'offrir une vision transversale de l'activité de l'entreprise, les systèmes d'aide à la décision collectent et stockent des données en provenance des bases de données des différents métiers de l'entreprise ou de sources externes (sites web, emails, etc.). Ces entreprises commencent à adopter la BI, une tâche très importante qui consiste à s'assurer qu'elles suivent un bon plan architectural de la BI dans leur processus de mise en œuvre, afin de déterminer le succès de leur investissement en BI. Selon, Jarke et al. [2011] une architecture est une structure générale représentant un système qui gère l'organisation de ses différents composants et surveille les relations entre ses composants.

De nombreuses organisations utilisent des systèmes de BI pour améliorer leurs processus de prise de décision. Mais bien que ces nombreuses organisations aient adopté des systèmes de BI, toutes ces implémentations n'ont pas abouti. Par ailleurs, chaque architecture d'un système BI est conçue pour décrire les détails d'un système précis afin de répondre à des besoins bien déclarés. D'une autre part, chaque entreprise étant un cas en soi, il existe des cas dans lesquels une configuration de BI traditionnelle peut suffire pour de nombreuses années. Rob et al. [2008] définissent une architecture BI comme un cadre décrivant différents composants de la BI (données, personnes, processus, technologie et gestion) et indique comment ces composants doivent être combinés pour assurer le bon fonctionnement d'un système BI.

La BI traditionnelle est une couche de reporting et d'analyse mise en œuvre au-dessus d'un entrepôt de données pour la génération de rapports historiques et l'analyse de tendances. Une architecture traditionnelle de BI est subdivisée en quatre phases :

1. **Phase de collecte ou d'alimentation :** utilise un outil ETL[2] qui charge périodiquement les données provenant de sources de données opérationnelles et les transforme et les réorganise en un modèle de données BI constitué de tables de dimensions et de faits, adapté à la production de rapports efficaces.

2. **Phase d'intégration :** c'est une phase de prétraitement qui permet de traiter les données transformées par l'outil ETL dans l'entrepôt de données (Data Warehouse).

[2]Extract Transform Load.

3. **Phase de distribution :** mit à la disposition des utilisateurs les données stockées dans l'entrepôt de données. Elle permet de segmenter à un contexte cohérent les données sous forme de *data marts* qui seront exploités par la modélisation des *cubes*, afin de supporter efficacement les analyses multidimensionnelles.

4. **Phase de restitution :** des outils d'interrogation, et de visualisation sont disponibles et utilisés pour des traitements analytiques et l'analyse avancée. L'analyse OLAP fournit des vues résumées et multidimensionnelles des données. Ils sont utilisés pour des fins d'analyse, de modélisation, de planification pour optimiser l'activité et créer des rapports aux utilisateurs professionnels de l'entreprise.

Comme le montre la Figure. 1.1, cette architecture traditionnelle de BI est utilisée depuis des décennies et reste l'architecture la plus dominante en matière d'analyse d'entreprise.

Figure 1.1: Architecture traditionnelle de Business Intelligence.

Avoir une architecture de BI solide est essentiel. Si l'architecture sous-jacente n'est pas conçue correctement, des incohérences entre les différents composants peuvent entraîner des problèmes, tels que, l'impossibilité de partager des informations entre les composants, l'incapacité de répondre aux exigences de l'entreprise et les performances médiocres de l'entreprise. Dans le pire des cas, une architecture de BI incorrecte peut entraîner le cas où des informations erronées sont transmises à la mauvaise personne au mauvais moment. Même dans le cas où les systèmes de BI sont fonctionnels malgré une architecture médiocre, les entreprises ne pourront pas maximiser la valeur qu'elles devraient tirer de leurs investissements en BI. Jusqu'à aujourd'hui, l'importance d'une bonne architecture de BI est un facteur essentielle, plusieurs revues de la littérature nous ont permis de constater qu'il existe plusieurs architectures BI. Ces architectures sont différentes dans leurs structures, telles que, les couches, les composants, les processus et les relations, et ce afin de guider les efforts de mise en œuvre d'un système BI. D'un point de vus technique, une architecture BI utilise une variation de techniques et d'outils selon le type de données et le besoin de l'entreprise.

2.4 Big Data : un nouveau défi pour la Business Intelligence

Dans les deux dernières décennies, il est presque quasi impossible de trouver une entreprise prospère qui n'exploite pas la technologie BI, car elle occupe un espace très concurrentielle entre les organisations. Les décideurs des organisations s'accélèrent entre eux pour acquérir ou pour adapter ces outils afin assurer une prise de décisions adéquate [Chaudhuri et al., 2011]. En conséquence, la technologie BI est intégrée totalement dans la vie professionnelle des entreprises quelque soit le domaine de son activité. Par exemple, la technologie BI est utilisée dans les entreprises pour la gestion des commandes et la gestion de la clientèle ; dans les services financiers pour l'analyse des sinistres et la détection de fraude, dans le transport pour la gestion des flottes, dans les télécommunications pour identifier les raisons qui poussent les clients à se désabonner, ainsi que dans le domaine des soins de santé pour l'exploitation des résultats des analyses médicales [Sangupamba Mwilu, 2018].

En revanche, cette diversification dans les domaines d'activité des entreprises, et la croissance d'Internet au cours des deux dernières décennies a fourni aux entreprises une multitude de nouvelles données pouvant être utilisées pour la vie stratégique et décisionnelle des entreprises [Glogowski, 2014]. Par exemple, seul le moteur de recherche Google a indexé plus de 58 milliards de sites Web [worldwidewebsize.com, 2018]. Facebook affirme avoir plus de 936 millions d'utilisateurs actifs par jour [Facebook.com, 2018]. Twitter publie plus de 600 millions de tweet par jour [internetlivestats.com, 2018], tandis que YouTube revendique plus de 1.9 milliards de visiteurs uniques chaque mois [YouTube.com, 2018]. Ce volume énorme de données a donné naissance au concept Big Data [Sharma et al., 2014]. La valeur de ces données est déterminée uniquement lorsque les utilisateurs et les applications accèdent à ces données et les utilisent pour prendre des décisions. En revanche, le Big Data est la tendance technologique basée sur l'analyse massive de données qui ne peuvent pas être traitées ou analysées à l'aide d'outils classiques, elles doivent généralement être gérées par une plateforme spécifique, telle que, la Business Intelligence, mais l'hétérogénéité et la complexité dans cette énorme quantité de données pose des nouveaux défis et des nouvelles opportunités dans l'architecture traditionnelle de Business Intelligence.

2.5 L'analyse en ligne: un outil de Business Intelligence

L'analyse en ligne OLAP consiste à exploiter intuitivement les entrepôts de données, il permet une analyse dynamique requise pour créer, animer, manipuler, et synthétiser l'information des modèles d'analyse de données [Codd et al., 1993]. elle est considérée

comme un support pour la prise de décision, elle peut être associée à un processus, à un type de système, à un type d'analyse, ou à un type de donnée [Jerbi, 2012]. Un système OLAP, peut être considéré comme un serveur d'applications pour un traitement directe des données, et il peut être vu comme un outil d'exploration des données grâce à une navigation interactive.

Dans la littérature, la composition d'un système OLAP est subdivisée en trois éléments : la base de données multidimensionnelle qui correspond à un entrepôts de données et ses magasins de données, un serveur OLAP, qui analyse et traduit les requêtes OLAP en requêtes pour la base de données, puis organise le résultat de la requête fournie par le système de gestion de base de données selon un format multidimensionnel pour l'afficher à l'utilisateur [Jerbi, 2012]. Le client OLAP permet aux utilisateurs d'effectuer les différentes analyses via une interface spécialisée et des opérateurs adaptés [Proulx, 2004].

Selon Aubay [2015], analyser consiste à déterminer les corrélations entre les données dans le but d'extraire des valeurs utiles, des suggestions ou des décisions qui reflètent les besoins réels de l'utilisateur. Un utilisateur dans un processus d'analyse en ligne OLAP fait recoure à l'utilisation d'une structure particulière appelée cube de données ou hypercube lorsque le nombre de dimensions est supérieur à trois [Gray et al., 1997]. C'est le concept central pour l'analyse OLAP. Un modèle d'analyse OLAP fournit des opérateurs pour la visualisation des informations contenues dans le cube de données. Ces opérateurs de navigation sont généralement décomposés en trois catégories : opérateurs de structuration (Rotate, Switch, Push, Pull), de sélection (Slice, Dice) et d'agrégation (Roll-up, Drill-down). Dans le contexte de notre travail, nous nous intéressons particulièrement à la dernière catégorie d'opérateurs.

Abdelhédi [2014] a présenté un exemple de cube qui permet l'analyse des ventes de matériel informatique. L'analyse des montants de ventes s'effectue en fonction de trois dimensions : les magasins où ont été effectuées les ventes, les dates de ventes et les produits vendus. Chacune de ces dimensions est associée à des paramètres de granularité différente (pour la dimension Magasin : ville, pays et continent). Ces niveaux hiérarchiques permettent d'obtenir des visions plus ou moins synthétiques lors des analyses OLAP (voir Figure 1.2). Torlone [2003] voit que la modélisation d'un entrepôt sous la forme d'un cube s'avère très limitée puisqu'elle se limite à trois dimensions. Il est indispensable de définir d'autre type de structures plus avancées pour améliorer l'élaboration des schémas multidimensionnels. Ces derniers permettent la modélisation des sujets d'analyse appelés faits, et d'axes d'analyse appelés dimensions [Kimball et al., 1996]. Les faits sont des regroupements d'indicateurs d'analyse appelés mesures. Les dimensions sont composées d'attributs, ap-

pelés paramètres, agencés de manière hiérarchique et qui modélisent les différents niveaux de détails des axes d'analyse. Un fait et ses dimensions associées composent un schéma en étoile [Kimball et al., 1996]. Les données des mesures sont appelées données factuelles car elles représentent un événement. Elles correspondent aux données des cellules du cube qui seront analysées en fonction des axes d'analyse. Le schéma multidimensionnel, associé à l'exemple de la Figure 1.2, est présenté dans la Figure 1.3.

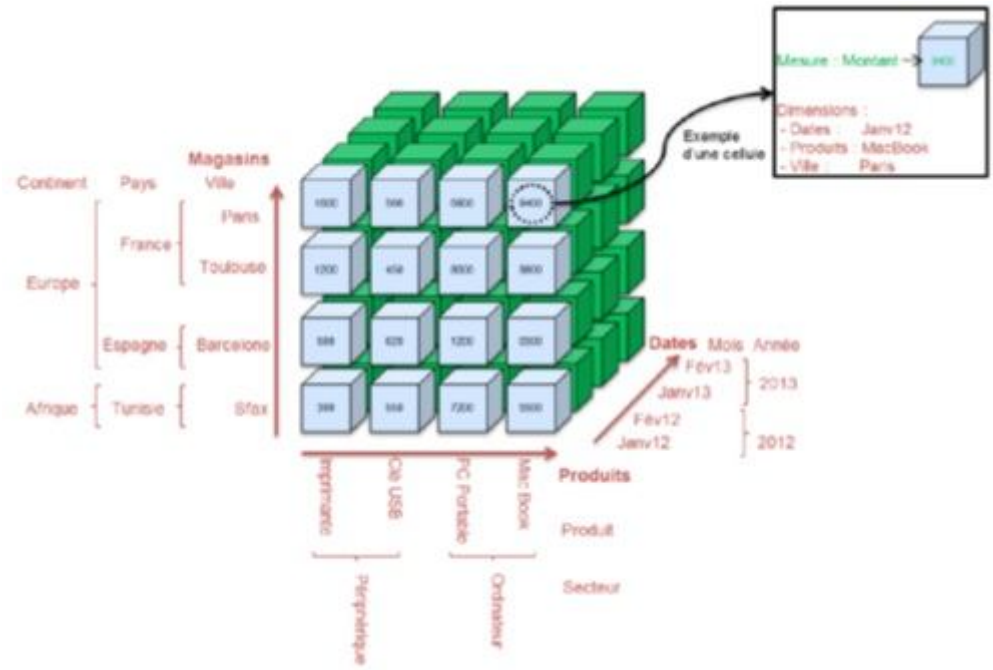


Figure 1.2: Exemple d'un cube représentant les ventes de matériel informatique.

3 Big Data

Au cours des cinq dernières années, la tendance du Big data a été émergé et est devenue un élément central de la recherche en Business Intelligence. Plusieurs nouvelles technologies et outils sont apparus afin de répondre à un triple problématique qui couvre : un Volume de données important à traiter, une grande Variété d'informations (structurées ou non structurées), et un certain niveau de Vélocité à atteindre [Khan et al., 2018].

3.1 Définitions

Plusieurs recherches approfondies sur la définition du Big Data ont été menées, mais aucune définition unique du Big Data n'est généralement connue. En revanche, la notion de "V" est souvent utilisée dans la définition du concept Big data dans la plupart des

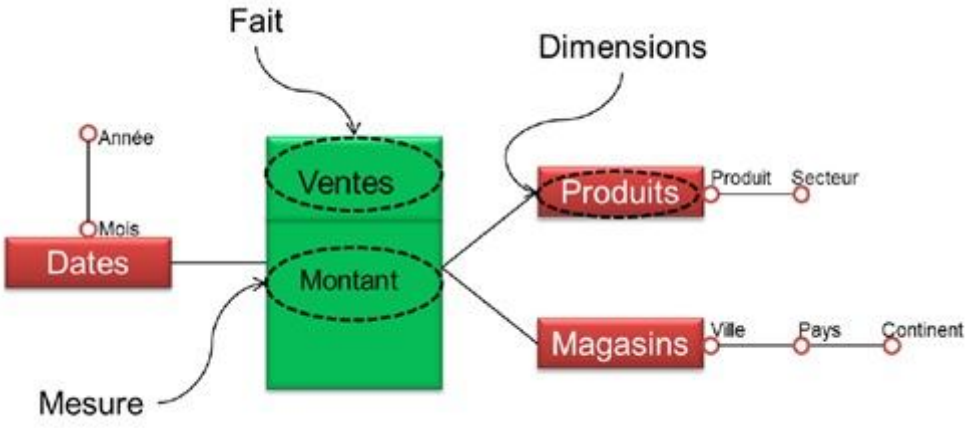


Figure 1.3: Exemple d'un schéma multidimensionnel [Abdelhédi, 2014].

références de recherche, Mashingaidze and Backhouse [2017] ont résumé dans leur travail les différentes définitions du concept Big Data et les thèmes de ces définitions, ils affirment que si les dimensions mentionnées (le volume, la vélocité et la variété) sont combinées avec les thèmes de la littérature scientifique, alors, le Big Data peut être défini comme suit:

"Big Data is data that is high in volume, is obtained from a variety of sources and is generated and analyzed at high velocity. Big Data is too large and complex for conventional technologies to manage and requires advanced technologies and techniques to store and analyze."

3.2 Caractéristiques du Big Data

Le Big Data est une question cruciale qui nécessite une attention sérieuse [Ularu et al., 2012]. L'énorme explosion de la quantité de données a généré des gros problèmes dans la vie des entreprises [Mervis, 2012] où les utilisateurs sont confrontés à de nombreux défis. En 2014, l'étude [Chen and Zhang, 2014] a expliqué que le Big Data ne pouvait pas être traité par les technologies et les méthodes existantes, mais la communauté des chercheurs a proposé des solutions selon des perspectives différentes. Par exemple, le Cloud Computing[3] utilisé pour répondre aux exigences en matière de matériel d'infrastructures, le NoSQL[4] utilisé pour pallier les lacunes du stockage des systèmes traditionnels, etc.

Dans des récentes études, le concept Big data peut être décrit sous plusieurs modèles. Laney [2001] caractérise le concept Big data par un modèle en 3Vs, il note qu'en raison de la forte augmentation des activités de commerce électronique, les données sont augmentées

[3]https://www.lebigdata.fr/definition-cloud-computing
[4]http://nosql-database.org/

selon trois dimensions à savoir :

1. **Volume :** c'est la taille de l'ensemble des données qui sont estimées de l'ordre de zettabytes de nos jours, et qui sont en croissance d'environ 40% chaque année. D'ici 2020, le volume cumulé de Big Data passera de 4,4 zettaoctets à environ 44 trillions de Go.

2. **Vélocité :** représente la fréquence ou la vitesse à laquelle les données sont générées, capturées et partagées. Cette vitesse augmente avec le temps. Les données doivent être analysées en temps réel pour répondre aux besoins des clients. Gartner [2015] prévoyait que 6,4 milliards d'appareils connectés seraient utilisés dans le monde en 2016 et qu'ils atteindraient 20,8 milliards d'ici 2020. En 2016, on estimait que 5,5 millions de nouveaux appareils étaient connectés chaque jour pour collecter, analyser et partager des données.

3. **Variété :** La variété fait référence au nombre de types de données. Les progrès technologiques permettent aux organisations de générer divers types de données structurées, semi-structurées et non structurées. Le texte, la photo, l'audio, la vidéo, les données par clic et les données de capteur sont des exemples de données non structurées, dépourvues de la structure normalisée requise pour un calcul efficace. Les données semi-structurées ne sont pas conformes aux spécifications de la base de données relationnelles, mais peuvent être spécifiées pour répondre à certains besoins structurels des applications. Le langage XBRL[5], développé pour l'échange de données financières entre organisations et agences gouvernementales, est un exemple de données semi-structurées. Les données structurées sont prédéfinies et peuvent être trouvées dans de nombreux types de bases de données traditionnelles. À mesure que de nouvelles techniques d'analyse sont développées, les données non structurées sont générées à un rythme beaucoup plus rapide que les données structurées et le type de données devient un obstacle moins important pour l'analyse [Lee, 2017].

Parfois, le modèle de [Laney, 2001] est étendu vers d'autres Vs en fonction des besoins particuliers. IBM [2012], a ajouté la *véracité* en tant que quatrième dimension, qui représente le manque de fiabilité et l'incertitude latente dans les sources de données. Des outils et techniques statistiques ont été développés pour traiter l'incertitude et le manque de fiabilité des Big Data avec des niveaux de confiance ou des intervalles spécifiés.

SAS [2013] a ajouté deux dimensions supplémentaires au Big Data : la *variabilité* et la *complexité*. La variabilité fait référence à la variation des débits de données, en plus

[5]Extensible Business Reporting Language

de la vitesse et de la variété des données, les flux de données peuvent fluctuer avec des pics et des creux imprévisibles. Les données de pointe déclenchées par des événements imprévisibles sont difficiles à gérer avec des ressources informatiques limitées, d'autre part, l'investissement dans les ressources pour répondre à la demande informatique de pointe sera coûteux en raison de la sous-utilisation globale des ressources. La complexité fait référence au nombre de sources de données. Le Big Data est collecté à partir de nombreuses sources de données. La complexité rend difficile la collecte, le nettoyage, le stockage et le traitement de données hétérogènes. Il est nécessaire de réduire la complexité avec les sources ouvertes, la plateforme standard et le traitement en temps réel des données.

Oracle a introduit la valeur en tant que dimension supplémentaire du Big Data. Les entreprises doivent comprendre qu'il est important d'utiliser le Big Data pour augmenter leurs revenus, réduire leurs coûts opérationnels et mieux servir leurs clients, tout en prenant en compte le coût d'investissement d'un projet Big Data.

3.3 Défis de Big Data

D'autre part, il sera difficile de relever les défis du Big Data, mais ces difficultés doivent être surmontées [Menaceur et al., 2016]. En revanche, SAS [2013] note que les entreprises seront confrontées à des problèmes de rapidité, traitement, interprétation, qualité, visualisation et du traitement des exceptions de données volumineuses.
En 2014, Jean-Michel Franco (Franco), a présenté une étude qui contient une classification en cinq défis dans le domaine de Big Data. Ces défis résident dans les moyens d'*acquisition*, *stockage*, *recherche*, *partage*, *analyse* et de la *visualisation* de données. Jean-Michel Franco a souligné que la maîtrise de ces cinq défis, alliée à un respect méthodologique du projet de Big Data permettra une mise en œuvre sûre et réussie.

Récemment, une nouvelle étude effectuée par Oussous et al. [2018] a donné plus que les défis cités dans [Franco, 2014]. Il existe de nouveaux défis qui concernent la sécurité et la confidentialité, en particulier dans les applications distribuées basées sur les données. Par ailleurs, l'étude [Oussous et al., 2018] a présenté aussi une synthèse sur des travaux de recherches qui discutent également les techniques et les méthodologies utilisées dans l'environnement Big Data et de la façon dont elles peuvent aider à améliorer les performances, l'évolutivité et la précision des résultats. La plupart des travaux de recherches cités ont porté uniquement sur les opportunités, les applications, les défis et les problèmes liés au Big Data (Par exemple Chen and Zhang [2014]; Benjelloun et al. [2015]; Wang et al. [2016]), d'autres ont préféré étudier les algorithmes et les techniques utilisés dans un tel contexte (c'est-à-dire l'exploration de données et l'apprentissage automatique) Radha and

Rao [2016]. Seuls quelques papiers traitent des technologies Big Data en ce qui concerne les aspects et les couches qui constituent un système Big Data du monde réel [Acharjya and Ahmed, 2016]. En fait, la plupart du temps, ces travaux sont axés sur les technologies Big Data et traitent celles-ci sous un angle particulier (analyse Big Data, exploration Big Data, stockage Big Data, traitement Big Data ou visualisation Big Data) [Acharjya and Ahmed, 2016, Lee, 2017, Siddiqa et al., 2017].

3.4 Technologies du Big Data

À l'ère technologique actuelle, effectuer des tâches analytiques sur le Big Data nécessite des méthodes spécifiques de stockage, de filtrage, de transformation et de récupération des données. Un aspect important pour la performance des applications de Big Data analytique est la localisation des données. Le Big Data étant caractérisé entre autres par son volume élevé, une grande variété d'informations (structurées ou non structurées), et un certain niveau de vélocité sont à atteindre. Différents Framework sont disponibles pour traiter ces Big Data. Hadoop et Spark sont tous deux un Framework open source permettant de traiter le Big Data. Hadoop fournit un traitement par lots, tandis que Spark prend en charge les traitements par lots et par flux, c'est-à-dire qu'il s'agit d'un cadre de traitement hybride. Les deux cadres ont leurs propres avantages et inconvénients [Talan et al., 2019].

1. **Ecosystème Hadoop :** Hadoop est un framework open source écrit en Java et destiné aux applications distribuées et à la gestion intensive de données. Il permet aux applications de travailler avec des milliers de nœuds et de pétabytes de données. Il a été conçu pour répondre aux besoins du Big Data. Hadoop est un projet Apache amélioré et soutenu par des entreprises telles que Cloudera, il est dérivé des solutions MapReduce, GoogleFS et Google BigTable.

   Généralement, lorsque nous parlons de Hadoop, nous faisons référence à deux produits open source HDFS (système de fichiers distribués Hadoop) et MapReduce. Le système de fichiers distribués Hadoop serait inspiré du système de fichiers (GFS) de Google [Ghemawat et al., 2003]. HDFS fournit un stockage évolutif, efficace et basé sur des répliques de données sur divers nœuds faisant partie d'un cluster. Hadoop fournit la flexibilité, ses performances évoluent presque linéairement avec le nombre de machines du cluster. La particularité de Hadoop est qu'il peut gérer différents types de données stockées dans n'importe quel type d'infrastructure. Nous pouvons ensuite mélanger des données structurées, semi-structurées et non structurées.

   Pour bien comprendre Hadoop, Samadi et al. [2018] illustrent l'utilité et l'efficacité de MapReduce et HDFS dans le traitement du Big Data en mode parallèle et distribué

sur des grands clusters. Kaur and Bharti [2019] ont aussi présenté les avantages et les désavantages de l'utilisation de l'écosystème Hadoop via quatre modules complémentaire.

2. **Apache Spark :** Spark est un framework qui suit Hadoop et son écosystème entourant le monde du Big Data, il a été développé à l'origine en 2009 chez l'AMPLab de l'Université de Californie, et devient open source comme projet Apache en 2010 [Kaur and Bharti, 2019]. L'objectif de Spark est de permettre aux développeurs de créer des applications Big Data. De plus, Spark vise à faciliter l'écriture et à accélérer l'exécution d'applications Big Data qui réutilisent les données à plusieurs reprises (algorithmes interactifs ou itératifs). Il est environ 100 fois plus rapide que son homologue Hadoop. Les données peuvent être mises en cache en mémoire. La mise en mémoire cache des données intermédiaires dans des algorithmes itératifs fournit une vitesse de traitement incroyablement rapide [Gu and Li, 2013].

4 Business intelligence et Big Data

Dans l'architecture traditionnelle de Business Intelligence présentée dans la section précédente (Figure 1.1), le processus de l'entreposage de données constitue un support essentiel dans la prise de décisions. La méthodologie BI traditionnelle fonctionne sur le principe de regrouper toutes les données de l'entreprise dans un serveur central (Data Warehouse). Les données dans ce cas-là sont généralement analysées en mode déconnecté, et structurées dans des SGBDR[6] avec un très peu de données non structurées [EDUCBA, 2019].

Une solution Big Data, est différente d'une BI traditionnelle dans les aspects suivants [Sawant and Shah, 2013]:

- Les données sont conservées dans un système de fichiers distribué et scalable plutôt que sur un serveur central;
- Les données sont de formats différents, à la fois structurées ainsi que non structurées;
- Les données sont analysées en temps réel ;
- La technologie Big Data s'appuie sur un traitement massivement parallèle (concept MPP[7]).

[6]Système de Gestion de Bases de Données Relationnelles
[7]https://www.sciencedirect.com/topics/computer-science/massively-parallel-processing

En revanche, Le Big Data peut gérer certaines problématiques dépassant le domaine de la BI. Aujourd'hui, le Data Warehouse atteint un niveau de maturité plus élevé, les compétences nécessaires ont été acquises, et de nouveaux outils très puissants ont pu être mis à la disposition des utilisateurs tel que Hadoop et Apache Spark. En conséquence, une nouvelle architecture BI est devenue disponible pour le traitement de Big Data. Cette architecture va nous permettre d'aller encore plus loin lors de l'analyse des données.

5 Entreposage de données dans le Big Data

5.1 Entreposage de données traditionnelles

Bill Inmon en 2001 définit l'entreposage de données ou Data Warehouse en anglais, dans son livre considéré comme étant la référence dans le domaine "Building the Data Warehouse" comme suit : "*Le Data Warehouse est une collection de données orientées sujet, intégrées, non volatiles et évolutives dans le temps, organisées pour le support d'un processus d'aide à la décision*".

- **Orienté sujet :** c'est l'une des caractéristiques les plus solides qui distinguent le Data Warehouse. Il est fondé sur des sujets majeurs de l'entreprise tels que : clientèle, ventes, produits, etc., contrairement à l'approche transactionnelle utilisée dans les systèmes opérationnels, qui sont conçus autour d'applications et de fonctions telles que : cartes bancaires, solvabilité client, etc. Dans un système opérationnel, les données sont essentiellement destinées à satisfaire un processus fonctionnel et obéissent à des règles de gestion, alors que, les données d'un Data Warehouse sont destinées à un processus analytique.

- **Intégré :** les données des sources de données sont importées dans le Data Warehouse via le processus appelé ETL qui désigne l'extraction, la transformation et le chargement. L'infrastructure ETL extrait les données utiles sur le plan analytique à partir des sources de données choisies, transforme les données extraites afin qu'elles se conforment avec la structure de l'entrepôt de données (tout en garantissant la qualité des données transformées), puis charge les données dans l'entrepôt de données.

- **Evolutives dans le temps :** dans un système décisionnel, il est important de conserver les différentes valeurs d'une donnée, cela permet les comparaisons et le suivi de l'évolution des valeurs dans le temps, alors que dans un système opérationnel la valeur d'une donnée est simplement mise à jour. Dans un Data Warehouse chaque valeur est associée à un moment

- **Non volatiles :** c'est ce qui est, en quelque sorte, la conséquence de l'historisation décrite précédemment. Une donnée dans un environnement opérationnel peut être mise à jour ou supprimée, de telles opérations n'existent pas dans un environnement Data Warehouse.

Par ailleurs, dans la littérature [Di Tria et al., 2018] les entrepôts de données traditionnels adoptent généralement une architecture à deux niveaux, où le niveau analytique est opposé à l'entrepôt de données. Cependant, dans l'architecture, trois couches physiques peuvent être observées :

- La couche source de données, qui contient des bases de données hétérogènes, pouvant être internes ou externes au système d'information ;
- La couche d'entrepôt de données qui stocke des données synthétiques et agrégées ;
- La couche analytique qui exécute les applications utilisées pour créer et déployer des rapports et des graphiques en appliquant des opérateurs OLAP ;

Dans ce type d'architecture, une phase ETL est nécessaire pour charger périodiquement des données dans l'entrepôt de données conformément à une stratégie de rafraîchissement [Kimball and Caserta, 2011]. Cependant, il existe une variante de cette architecture, dans laquelle le processus ETL n'alimente pas directement l'entrepôt de données, mais une base de données globale et réconciliée qui s'occupe de l'alimentation de l'entrepôt de données (architecture à trois niveaux).

5.2 Les démarches d'élaboration des Data Warehouses

Dans le cadre de l'entreposage de données et selon [Abdelhédi, 2014], le processus pour construire un entrepôt de données comporte plusieurs étapes successives : la conception d'un schéma multidimensionnel, la création de l'entrepôt conforme au schéma et le chargement de l'entrepôt depuis les sources. La conception d'un schéma multidimensionnel peut être effectuée selon l'une des 3 démarches suivantes:

- La démarche ascendante, utilise uniquement le schéma des sources pour générer des schémas multidimensionnels candidats sans prendre en compte, dans un premier temps, les besoins des décideurs. Ceux-ci choisissent ensuite le schéma le plus adapté à leurs besoins.

- La démarche descendante, prend uniquement en compte les besoins des décideurs, elle se base sur la spécification de ces besoins pour définir les sujets et les axes d'analyse. A l'issue du processus d'élaboration du schéma multidimensionnel, la correspondance entre le schéma résultat et la source de données est établie.

- La démarche mixte, combine les deux démarches précédentes. En effet, cette démarche construit d'une part des schémas candidats à partir des sources de données (démarche ascendante) et d'autre part des schémas multidimensionnels à partir des besoins d'analyse (démarche descendante). L'informaticien doit confronter ces deux types de schémas pour obtenir un schéma multidimensionnel cohérent et répondant aux besoins des décideurs.

5.3 Du Data Warehouse traditionnel vers le Big Data Warehouse

Comme nous l'avons indiqué dans la section précédente, plusieurs problèmes doivent être résolus dans les environnements traditionnels de l'entreposage de données, les entrepôts traditionnels ne peuvent pas prendre en charge les défis présentés par le modèle 5V du Big Data. Il est nécessaire de migrer les entrepôts traditionnels vers un environnement de Big Data. L'auteur dans [Krishnan, 2013] note qu'avec l'avènement du Big Data, le défi pour les entrepôts de données est de réfléchir à une approche complémentaire avec le Big Data, d'où on pourrait concevoir un modèle hybride. Dans ce modèle, les restes de données optimisées opérationnelles très structurées seront stockées et analysées dans l'entrepôt de données, tandis que les données qui sont fortement distribuées et non structurées seront contrôlées par des nouvelles technologies de Big Data [Singh et al., 2018] susceptibles d'améliorer d'une manière décisive les besoins en performances de l'entrepôt de données actuelles et de fournir une plate-forme globale permettant de répondre aux exigences étendues des nouvelles données et des utilisateurs associés. Par conséquent, les concepteurs et les gestionnaires doivent examiner en parallèle et avec soin si une architecture Big Data constitue la bonne solution, en évaluant et en comparant les coûts et les avantages dans un contexte commercial donné.

Ces dernières années, de nombreux progrès et développements architecturaux et technologiques ont adopté le terme Big Data Warehouse (BDW) comme référence à un entrepôt de données caractérisé par le 5 Vs. La solution proposée dans la littérature pour la création d'un BDW est l'adoption du framework Hadoop, qui est généralement utilisé conjointement avec un entrepôt de données traditionnel [Mohanty et al., 2013]. De nombreux auteurs discutent de ce besoin et proposent des travaux principalement guidés par des approches basées sur des cas d'utilisation, dans lesquelles des solutions spécifiques sont recommandées

et testées, donnant principalement des directives non structurées sur la manière de concevoir des Big Data Warehouse et aussi, une révision des techniques traditionnelles de modélisation.

Krishnan [2013] dans son travail a discuté l'intégration du Big Data dans le processus de l'entreposage de données, et les différentes techniques et les pièges possibles à éviter dans certain type de technologie. L'auteur s'est concentré aussi sur les problèmes de la complexité et l'hétérogénéité des technologies Big Data; quelles seront les performances et les scalabilités de chaque technologie, et comment pouvoir maintenir les performances pour les nouveaux environnements.

Das and Mohapatro [2014] dans leur papier, ont présenté les technologies Big Data et les limites d'un système d'entreposage traditionnel basé sur SQL. Pour relever tous les défis du traitement des données volumineuses en terme de traitement rapide et d'adaptation à des données volumineuses extrêmement variées, les auteurs intègrent l'entrepôt de données a un moteur Hadoop afin de pouvoir exploiter Map Reduce en parallèle puissance de calcul. Les auteurs ont présenté aussi une interface commune permettant de créer un entrepôt de données sur un moteur Hadoop et l'intégrer avec des solutions BI.

Les travaux présentés ci-dessus illustrent parfaitement l'interfaçage de Big Data avec le Data Warehouse. Les données hétérogènes et non structurées peuvent être collectées dans un HDFS qui représente l'endroit intermédiaire avant que ces données ne soient transformées et chargées à l'aide d'outils spécifiques dans le Data Warehouse et les outils traditionnels de BI [Sawant and Shah, 2013].

Di Tria et al. [2018] propose une architecture améliorée de Big Data Warehouse, qui consiste à réduire le temps nécessaire à l'intégration de nouvelles sources de données et à l'inclusion immédiate des nouvelles exigences de l'entreprise. Les données sont immédiatement disponibles pour les analyses, car l'architecture sous-jacente est basée sur un entrepôt virtuel de données qui ne nécessite pas la phase d'importation. Des exemples d'application ont été présentés dans ce travail afin de montrer la validité de cette approche par rapport à une approche traditionnelle. Les différences entre un entrepôt traditionnel de données et un BDW sont résumées dans la Table 1.1.

5.4 Analyse multidimensionnelle sur le Big Data

Les systèmes OLAP font désormais partie des solutions prometteuses pour améliorer le processus de prise de décision. Ils ne sont pas adaptés aux besoins et aux contextes d'analyse des décideurs suite à la diversification des particularités des utilisateurs et l'énorme

		Data Warehouse	Big Data Warehouse
Architecture	Niveaux	Deux, Trois	Un
	ETL	Oui	Non
Modèle logique		ROLAP, MOLAP, HOLAP	Non-relationnel
Méthodologie	Stratégie	Basée sur les données, Basée sur les exigences, Hybride	Hybride
	Automatique	Optionnelle	Obligatoire

Table 1.1: Data Warehouse et Big Data Warehouse

augmentation du volume de données. En conséquence, les travaux de recherche sont orientés vers les défis des grandes architectures de données, où de nouvelles techniques ont été utilisées pour le stockage et le calcul des cubes de données OLAP sur le Big Data.

Aujourd'hui, nous parlons du Big Data Warehouse où, les entrepôts n'utilisent pas les modèles de données classiques (Entité-association et Relationnel), mais des modèles multidimensionnels où, les données sont organisées en termes de faits et de dimensions. Les modèles conceptuels multidimensionnels (Etoile, Constellation et Flocon) sont adaptés aux analyses OLAP effectuées par les décideurs des entreprises.

Les systèmes du Big Data modernes, tels que Hadoop, sont utilisés comme solution de remplacement pour l'entreposage de données [Hollingsworth, 2012, Kuldeep and Bhimappa, 2014]. Ils collectent et stocke des flux de données complexes par nature en raison du volume, de la vitesse, de la valeur, de la variété, de la variabilité et de la véracité [Russom et al., 2011].

Pour les raisons sus-citées, la mise en œuvre d'un OLAP traditionnel, à savoir le système ROLAP basé sur un SGBDR, semble inadéquate, car les nouvelles architectures de données et les outils d'analyse volumineux vont au-delà des entrepôts SQL et des moteurs OLAP [Song et al., 2015]. D'autre part, les travaux [Cuzzocrea et al., 2013] mettent en évidence les problèmes et les tendances de recherche soulevés dans le domaine de l'entreposage de données et de l'analyse multidimensionnelle OLAP sur le Big Data.

Xia [2008] a présenté une approche d'analyse OLAP basée sur le modèle de programmation parallèle simplifié MapReduce. Les entrées du système se présentent sous la forme

de SMS[8], exploités par les deux fonctions *Map* et *Reduce* qui produisent de très petits messages populaires avec un taux de couverture d'envoi élevé et répondent aux besoins réels.

Wang et al. [2014] ont présenté un modèle de cube de données pour les documents XML. Ils ont proposé un algorithme optimisé implémenté par *MapReduce* sur Hadoop et répondant à la demande croissante d'analyse massive de données XML dans OLAP.

Le travail de Song et al. [2015] présente la conception, la mise en œuvre et l'évaluation de HaoLap, c'est un système OLAP pour le Big Data, basé sur des modèles et des algorithmes utilisant une architecture bien spécifiée de Hadoop . HaoLap a été testé sur plusieurs grands ensembles de données et applications OLAP et a comparé ces performances avec d'autres outils tels que : Hive, HadoopDB, HBaseLattice et Olap4Cloud.

Les auteurs de [Ranawade et al., 2017] ont proposé un système principalement destiné à la construction de cubes OLAP sur l'écosystème Hadoop. L'architecture interne du système se présente sous la forme de trois niveaux, où, chaque niveau fournit des services au niveau supérieur. Les cubes OLAP ont été créés par Apache Kylin [Apache, 2015].

Chen et al. [2017] ont présenté un système OLAP distribué selon une architecture a quatre modules. Le module d'acquisition de données est chargé d'obtenir des données à partir de sources de données structurées et non structurées. Le module de stockage de données conserve les données source dans HDFS, le magasin de valeurs-clés et la base de données relationnelle. Le module d'analyse OLAP effectue un calcul de cube OLAP et une exécution de requête de type SQL. Le module de visualisation des données permet de calculer la définition des cubes OLAP, la visualisation des résultats de la requête et la spécification des privilèges de l'utilisateur.

Santos et al. [2017] ont étudié le problème de la migration d'un entrepôt traditionnel de données vers un entrepôt de données sur Big Data. Les auteurs proposent une architecture de stockage de données volumineuses dans laquelle les données des sources de données sont obtenues par les outils ETL et initialement accumulées dans la zone de stockage intermédiaire HDFS, puis les données sont transformées et chargées dans le magasin de données volumineuses implémenté dans Hive.

[8]Short Message Service

Références	Travail réalisé	Type analyse	Type de données	Source de données	Modèle de programmation	Technologie	Contexte utilisateur	Entrer du système/approche
[Xia, 2008]	Approche d'analyse OLAP	Multi	non structurer	database	MapReduce	Hadoop	Non	SMS
[Wang et al., 2014]	Modèle du cube OLAP	Multi	semi-structurer	database	MapReduce	Hadoop	Non	/
[Song et al., 2015]	Système Haolap	Multi	non structurer	Large datasets	MapReduce	Hadoop	Non	/
[Ranawade et al., 2017]	Modèle du cube OLAP	Multi	non structurer	Large datasets	Kylin	Hadoop	Non	Requête HQL
[Chen et al., 2017]	Système OLAP distribué	Multi	structurer	database	Kylin/Impala	Hadoop	Non	Requête SQL

Table 1.2: Synthèse sur l'analyse multidimensionnelle OLAP sur le Big Data

6 Big Data analytique pour le Business Intelligence

Avec les défis commerciaux constants face à l'explosion de volume de données, les organisations doivent gérer tout ce volume d'informations, structurées et non structurées, issues de différentes sources, afin d'améliorer la prise de décision. La voluminosité et la complexité de ces nouvelles sources de données ont engendré également des nouvelles questions auxquelles il n'est pas possible de répondre efficacement avec les méthodes d'analyse traditionnelles. Pour surmonter ces problèmes, des nouvelles méthodologies et techniques de traitement ont été développées, ouvrant ainsi une nouvelle ère dans la prise de décision dans les entreprises, appelées Business Analytics [Mortenson et al., 2015]. La Figure 1.4 illustre l'évolution des techniques d'analyse et de la terminologie associée au cours des dernières décennies.

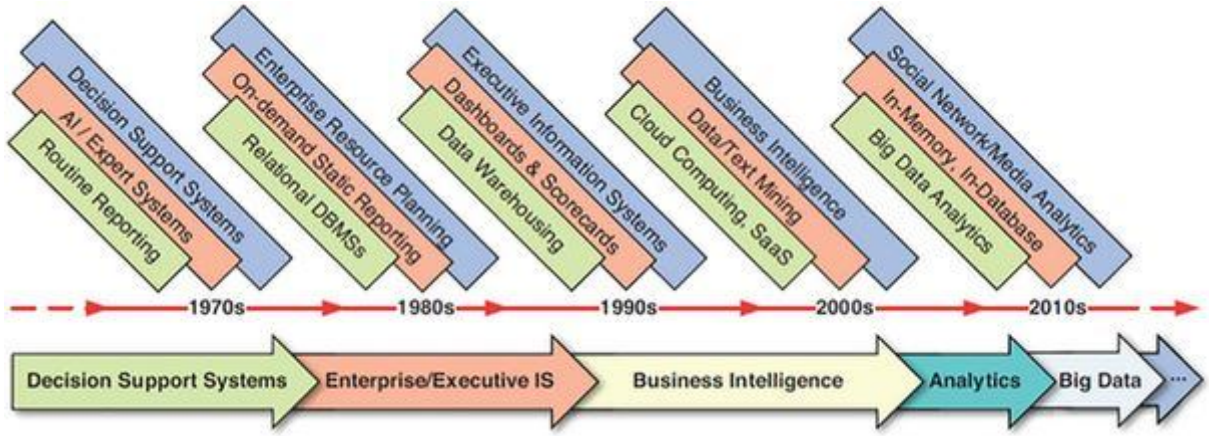


Figure 1.4: Vue longitudinale de l'évolution de l'analytique [Dursun and Hamed, 2018].

En revanche, Mashingaidze and Backhouse [2017] définissent Business Analytics comme l'ensemble de compétences, d'applications, de technologies, d'architectures, de processus et de méthodologies utilisés pour collecter, stocker et récupérer des données à des fins d'analyse, afin de faciliter la prise de décision, d'informer la stratégie commerciale et, à terme, d'améliorer les performances.

Les organisations font de plus en plus de la BI sur le Big Data, d'où l'appellation du Big Data Analytics qui se réfère à l'application des techniques analytiques sur ces gros volumes de données. Chen et al. [2012] définissent le business intelligence and Big Data Analytics comme les techniques, technologies, systèmes, pratiques, méthodologies et applications qui analysent les données critiques de l'entreprise afin de mieux l'aider à comprendre ses affaires et son marché et à prendre les bonnes décisions au bon moment. En revanche, le Big Data et la BI ont attiré une attention particulière dans la vie des chercheurs du domaine dans

les dernières années, d'où, la plupart des recherches existantes portent sur les méthodes, les problèmes techniques et les solutions possibles pour l'utilisation de Data Analytics et la BI.

Phillips-Wren et al. [2015] présentent dans un papier une architecture de la BI adaptée dans le cadre du Big Data Analytics. Cette architecture comprend tous les éléments relatifs à une architecture type de BI, elle tient aussi compte des nouvelles architectures de données et outils analytiques parallèles. Par ailleurs Sangupamba Mwilu [2018] a apporté des modifications sur l'architecture proposée par Phillips-Wren et al. [2015], où ils restructurent l'architecture en trois composants : la collection et la consolidation des données, la modélisation et le stockage, et l'analytique.

Une forte importance est donnée au composant de collection et de consolidation des données, il assure toutes les activités de collecte, transfère et chargement depuis des différentes sources de données jusqu'à la mise à jour de l'entrepôt de données. Des méthodes, des techniques et des outils sont intégrés dans le composant de modélisation et de stockage afin de gérer les grands volumes de différents types de données. Ces derrières, représentées selon une vue multidimensionnelle, avec l'utilisation des trois concepts fondamentaux de la modélisation multidimensionnelle : le fait, la dimension et la mesure [Al-Aqrabi et al., 2012]. Le dernier composant c'est l'analytique et qui représente la dernière étape de la BI. Elle comprend tous les types d'analyse ainsi que la transmission des données aux utilisateurs finaux. Les éléments essentiels de cette étape sont des outils de reporting, de data mining et d'OLAP.

Lustig et al. [2010] présentent une taxonomie en trois dimensions des techniques analytiques : descriptif, prédictif et prescriptif. Une révision un peu récente a été effectuée par les auteurs dans [Banerjee et al., 2013] qui divise l'analyse en ce qui concerne son orientation en quatre dimensions, ou en ajoutant un composant de diagnostic. La Figure 1.5 illustre une vue taxonomique simple de l'analyse.

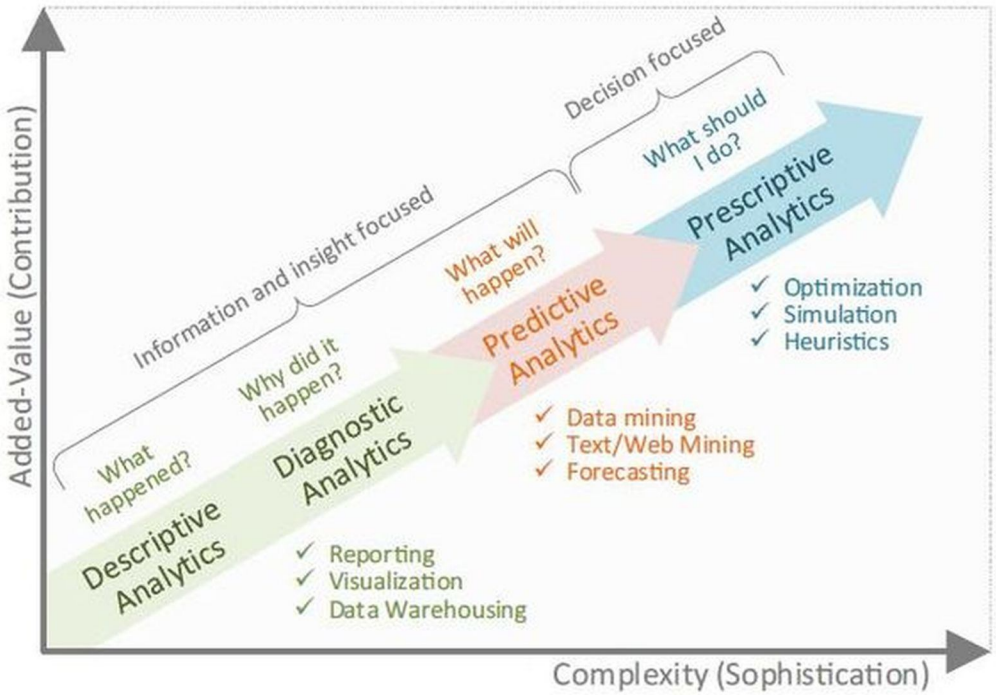


Figure 1.5: Différents types de Business Analytics [Dursun and Hamed, 2018].

Récemment, plusieurs groupes de recherche sont orientés vers des recherches sur les deux paradigmes l'analyse de Big Data et la BI. Ting-Peng and Yu-Hsi [2018] ont présenté dans leur papier un état de synthèse sur la tendance temporelle des publications dans le domaine du Big Data et celle du BI. Ils examinent également 141 articles qui incluent simultanément le Big Data et la BI comme mots clés. La figure 1.6 montre l'évolution temporelle des publications de Big Data et BI. Le nombre de ces publications a considérablement augmenté pour atteindre 32 en 2015 et a continué d'augmenter, mais il n'est toujours pas comparable à celui des papiers Big Data. La raison du faible nombre de publications pourrait être que, les applications de BI et de Big Data se chevauchent généralement, mais la plupart des papiers peuvent choisir de montrer leur principale orientation technique ou managériale. Une autre possibilité est que le Big Data est un mot à la mode qui a été largement utilisé dans les points de vente commerciaux, tandis que la Business Intelligence est plus restreinte à certains domaines d'activité. Par conséquent, la plupart des journaux préfèrent le Big Data au BI.

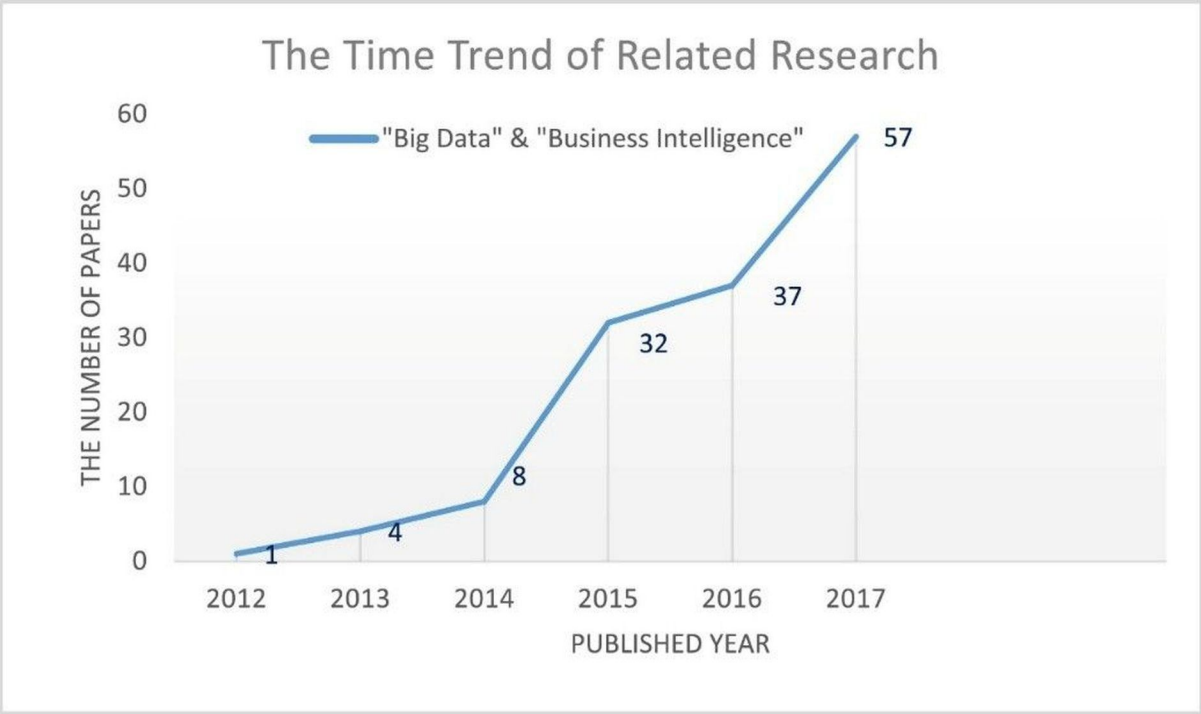


Figure 1.6: Tendance de la recherche Big Data et BI [Ting-Peng and Yu-Hsi, 2018]

7 Conclusion

Dans ce chapitre, nous avons abordé les principales notions et concepts de la Business Intelligence et Big Data. A travers les différentes sections que nous avons présenté, nous concluons que la Business Intelligence et le Big Data occupent un grand intérêt dans la vie des chercheurs du domaine. De nouvelles technologies et techniques permettent de combiner les deux concepts afin de répondre aux besoins des utilisateurs finaux de l'entreprise. Le but est de permettre aux utilisateurs de retrouver les informations dont le contenu répond à leur besoin en information, il s'agit donc de retourner l'ensemble de données pertinentes afin de mieux aider les décideurs de l'entreprise à comprendre leurs affaires et leurs marchés et à prendre les bonnes décisions au bon moment.

Chapter 2

Personnalisation pour le filtrage collaboratif dans le Big Data

1 Introduction

Ces dernières années l'émergence des mégadonnées (ou Big Data) a incité les experts du domaine de développement des nouvelles approches et des technologies pour la gestion de ces grandes masses de données. Des grandes plateformes ont été investies dans les techniques de la personnalisation et du filtrage des données comme solution pour les intégrer dans ces systèmes de BI tel que l'analyse multidimensionnelle OLAP. Les utilisateurs avaient la possibilité d'accéder rapidement aux données les plus pertinentes pour eux.

L'objectif de ce chapitre est de présenter un état de l'art sur les techniques de personnalisation des entrepôts de données à base de profil utilisateur, ainsi que, les techniques d'expansion des requêtes, par l'utilisation d'une approche linguistique, afin d'élargir l'espace de recherche et de récupérer les informations les plus pertinentes au besoin de l'utilisateur. Un filtrage collaboratif à base de contenu est utilisé pour améliorer le processus décisionnel.

2 Profilisation

Ces dernières années, le sujet de profilisation dans le processus d'analyse en ligne (OLAP) occupe une grande part dans le domaine de l'informatique décisionnelle (Business Intelligence) [Menaceur et al., 2017b], d'autant plus que le monde aujourd'hui vit une croissance massive dans le volume des données (Big Data), ces dernières sont très souvent décrites comme des données qui dépassent les capacités de l'organisation à stocker ou à analyser dans le but de prendre une décision précise et opportune [Kulkarni and Inc., 2013]. Comme nous avons cité dans la section 3.2, le concept Big data a été caractérisé par un modèle en 3Vs. Ces dimensions (Vs) présentent en réalité les grands défis lorsqu'il s'agit de l'analyse des données.

Partant de ce constat, nous souhaitons répondre à la problématique suivante : Parmi le volume important des données stockées, comment extraire les données pertinentes qui répondent mieux au contexte et au besoin de l'utilisateur.

2.1 Définition du profil

L'analyse des références académiques nous permet de constater les définitions suivantes:

"Une source de connaissance qui contient des acquisitions sur tous les aspects de l'utilisateur qui peuvent être utiles pour le comportement du système" [Wahlster and Kobsa, 1986].

"Toutes les variations qui caractérisent un utilisateur ou un groupe d'utilisateurs, peuvent se regrouper sous le terme de profil de l'utilisateur" [Bruande and Chevallet, 2003].

Le profil utilisateur dans le contexte des systèmes de personnalisation d'informations, peut être défini comme une structure qui permet de modéliser et stocker des informations relatives à l'utilisateur [Brusilovsky, 1998]. Cette proposition, bien que générale, correspond à nos orientations, il peut contenir :

(a) **Ses données personnelles** telles que son identité (nom, prénom, etc.), ses données démographiques (âge, sexe, adresse, situation familiale, etc.), ses données professionnelles.

(b) **Son historique/ feedbacks** qui regroupe l'ensemble des informations collectées sur son comportement, de façon explicite ou implicite (par exemple, le nombre de clics qu'il a effectué sur le lien d'une page ou le nombre des requêtes qu'il a émis, etc.).

(c) **Les annotations** associées par l'utilisateur aux documents, pouvant être sous différentes formes (par exemple, les annotations textuelles, les signets qui mémorisent les liens vers d'autres documents, les tags, qui sont les références sous forme d'un ensemble de mots-clés choisis librement par l'utilisateur pour identifier le document visité).

(d) **Ses préférences** désignent les caractéristiques de l'utilisateur, en terme de présentations ou d'interactions avec les informations (par exemple, des couleurs et/ou les styles de présentation de pages web préférées, etc.).

(e) **Ses intérêts** expriment ses domaines d'expertise ou son périmètre d'exploration. Ils sont généralement définis par un ensemble de mots clés ou concepts, le plus souvent pondérés.

Par ailleurs, l'évolution du profil utilisateur se fait souvent selon un processus incrémental basé sur l'addition de nouvelles informations dans la représentation du profil [Daoud et al., 2008]. Le contenu du profil utilisateur dépend à de la mise à jour automatique au fil du temps lorsqu'il s'agit des préférences et des intérêts. Cette mise à jour consiste principalement à deux phases, à savoir:

(a) La capture des changements des centres d'intérêt de l'utilisateur;

(b) La propagation de ces changements au niveau de la représentation du profil;

En revanche, le profil est relativement stable dans le temps pour les données personnelles.

2.2 Modélisation du profil utilisateur

Un profil utilisateur a pour objectif de permettre à un système de s'adapter à l'utilisateur, son contenu donc dépend fortement de l'application qui l'exploite. La modélisation de profil utilisateur est un processus à différentes étapes, à savoir (1) la représentation du profil utilisateur, (2) la collecte des informations et (3) la construction du profil utilisateur (Voir Figure 2.1).

1. **Représentation du profil utilisateur :** dans la littérature, la représentation et la structuration des profils utilisateurs prend trois formes d'approches qui peuvent être :

   (a) **Vectorielle :** le profil est constitué d'un ou plusieurs vecteurs définis dans un espace de termes d'indexation [Mc Gowan, 2003].

   (b) **Hiérarchique :** les caractéristiques d'un utilisateur sont organisées dans une structure hiérarchique de concepts représentant les domaines d'intérêt [González et al., 2002].

   (c) **Multidimensionnelle :** le profil est représenté par un modèle structuré de dimensions prédéfinies (données personnelles, domaine d'intérêt, préférences de livraison, etc.) [Kostadinov, 2003].

2. **Collecte des informations :** une fois le choix de la représentation est terminé, la collecte des informations que représentent l'utilisateur est indispensable. Elle s'effectue d'une manière explicite ou implicite [Gauch et al., 2003, Kraft et al., 2005].

3. **Construction du profil utilisateur :** la construction d'un profil utilisateur nécessite d'effectuer des analyses sur les données, en fonction de la modélisation utilisateur

mise en œuvre. En terme de type de modélisation, on peut distinguer deux grands types : la modélisation du comportement et la modélisation des intérêts [On-At, 2017].

(a) **La modélisation de comportement :** consiste à analyser les comportements des utilisateurs via leurs interactions avec le système. Elle est généralement utilisée dans les services Web (ex. historique de navigation, transaction avec le serveur Web). Ce type de modélisation a pour but de prédire ou de déterminer les préférences ou les feedbacks de l'utilisateur (ex. déterminer les parcours de navigation récurrents, valider la pertinence des campagnes marketing).

(b) **La modélisation des intérêts :** consiste à construire une liste à partir d'une analyse des données, représentant un point de vue du système, ce que sont les intérêts de l'utilisateur. Pour extraire les intérêts de l'utilisateur, on peut le faire de façon directe à partir des données explicites de l'utilisateur ou de façon implicite à partir des données collectées..

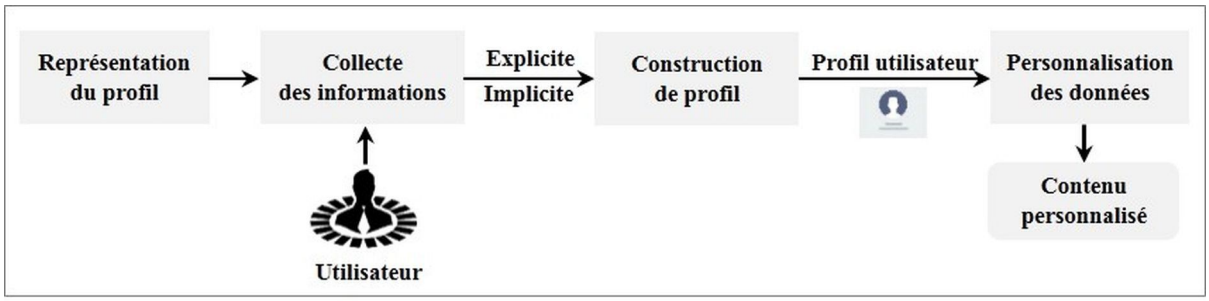


Figure 2.1: Les étapes de construction du profil utilisateur

2.3 Exploitation du profil utilisateur

La plupart des systèmes décisionnels actuels donnent accès à un grand nombre de sources hétérogènes et distribuées. Le volume des données accroît au fur et à mesure avec une vitesse très élevée. L'utilisateur se voit confronté à une surcharge informationnelle dans laquelle il est difficile de distinguer l'information pertinente de l'information secondaire et même du bruit. Afin de surmonter ce type de problème, il est indispensable d'utiliser la personnalisation comme technique pour adapter le processus décisionnel est-ce par l'intégration des préférences et des caractéristiques de l'utilisateur enregistrées dans son profil. Ensuite, faire de la recommandation sur les informations pertinentes. En revanche, dans la littérature on distingue deux techniques d'adaptation, à savoir :

(a) **Filtrage des données ou des résultats :** la requête de l'utilisateur peut être affinée avant son exécution afin d'avoir directement un résultat pertinent ([Koutrika and Ioannidis, 2004, 2005].

(b) **Tri des résultats :** exécuter la requête de l'utilisateur puis réduire le résultat en présentant seulement l'information pertinente [Bradley et al., 2000]. En outre, les résultats peuvent être triés afin de présenter les informations les plus pertinentes en premier lieu [Sun et al., 2008].

La deuxième forme de prise en compte de l'utilisateur dans le processus décisionnel est la recommandation qui consiste à proposer à l'utilisateur des données qui peuvent l'intéresser et surtout utiliser ses préférences pour avoir des recommandations personnalisées.

3 Contextualisation

Définir le profil de l'utilisateur est un critère indispensable, mais il n'est pas assez suffisant pour la personnalisation dans les entrepôts de données, car il est souvent lié à d'autres critères tels que les préférences et le contexte. En ce qui concerne les préférences, elles sont liées fortement au profil utilisateur, et on ne peut en aucun cas séparer les unes des autres, mais leur description peut changer en fonction du contexte. Une préférence peut être associée à un contexte, dans ce cas, elle est dite contextuelle (ou conditionnelle). Le contexte d'une préférence définit sa portée, c'est-à-dire l'environnement dans lequel elle doit être prise en compte. Donc on peut noter qu'une préférence contextuelle est un couple (P, C), où P est une préférence et C est un contexte. La partie contexte spécifie les conditions sous lesquelles la préférence P sera activée, ou P peut être formulée selon une approche quantitative ou qualitative.

Toutefois, l'analyse des définitions présentes dans la littérature [Brown et al., 1997, Schmidt et al., 1999] nous conduit à constater une définition du contexte comme suivante : *"Toute information susceptible de caractériser la situation d'une entité. Une entité est une personne, un lieu ou un objet qui est considéré pertinent pour l'interaction entre l'utilisateur et l'application, incluant l'utilisateur et l'application"* [Dey, 2001].

De ce qui précède, nous constatons qu'il y a une ambiguïté autour des trois concepts : profil, contexte et préférences. Le sens qu'on leur donne change d'une approche à une autre et il arrive souvent que l'un d'entre eux soit utilisé à la place des deux autres ou des trois à la fois. Cette ambiguïté de la terminologie rend difficile l'étude et la compréhension de la problématique liée à la personnalisation.

4 Personnalisation

Les techniques de personnalisation constituent un enjeu majeur dans l'industrie informatique, elles sont initialement abordées dans trois domaines technologiques : l'Interaction Homme-Machine (IHM), la Recherche d'Information (RI), et les Bases de Données (BD) [Kostadinov, 2007]. Plus tard, elles sont élargies pour inclure presque tous les domaines tels que l'entreposage de données et l'analyse de Big Data.

Par ailleurs, pour répondre à la problématique mentionnée dans la section 2.2 et pour pouvoir discriminer les utilisateurs en fonction de leurs besoins spécifiques, certains systèmes proposent des techniques de personnalisation basées sur le profil de l'utilisateur comme solution, pour obtenir les informations pertinentes relatives aux besoins de cet utilisateur [Domshlak and Joachims, 2007]. Le succès ou le rejet de ces techniques de personnalisation reste liés à la description des caractéristiques de l'utilisateur, ces derrières sont modélisées et formulées sous forme de profil et des préférences de recherche. [Kwok, 1998], dans son agenda de recherche affirme que la personnalisation d'un système consiste à définir, puis à exploiter un profil utilisateur qui ne peut être définie de façon standard, il regroupe souvent un ensemble de caractéristiques servant à configurer ou à adapter le système à l'utilisateur (Voir Figure 2.1).

4.1 Personnaliser dans les entrepôts de données

La personnalisation dans les entrepôts des données (ED) a fait l'objet de très nombreux travaux de recherche, et peut se situer à plusieurs niveaux dans les systèmes d'analyse OLAP.
Elle peut porter sur le schéma de l'entrepôt de données, la visualisation des données, l'analyse et/ou sur l'interrogation. Tous ces niveaux sont principalement basés sur le profil de l'utilisateur. Par ailleurs, d'autres chercheurs, voient que la personnalisation dans les ED est utilisée pour expliquer comment recevoir à partir d'une grande quantité d'informations uniquement la partie qui intéresse un individu ou un groupe d'individus [Khemiri, 2015], et ce, par l'intégration de son profil (ses intérêts, ses préférences ou même ses contraintes, ses comportements, etc.) dans le processus de personnalisation (Voir Figure 2.2).

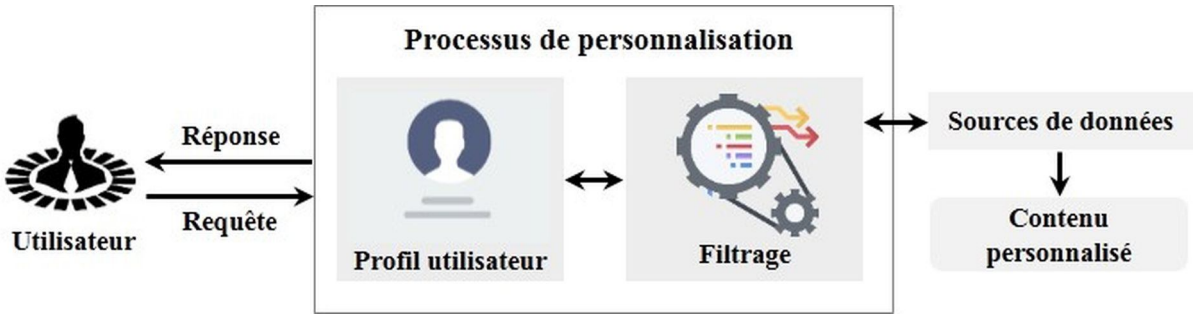


Figure 2.2: Personnalisation à base de profil

En revanche, quelque soit le domaine technologique, la personnalisation peut être exploitée selon deux modes de gestion :

(a) En personnalisation (ou interrogation), consiste à adapter le système OLAP en fonction de préférences utilisateur explicitement ou implicitement collectées;

(b) En recommandation, où, on recommande les requêtes à base des préférences ou de l'historique de navigation de l'utilisateur.

Après une revue des définitions qui existent dans la littérature, nous constatons une certaine ambiguïté quant à la définition de la personnalisation et de la recommandation. Il est alors temps de faire la distinction entre la recommandation et la personnalisation pour mieux comprendre les problèmes posés et les solutions proposées.

4.2 Personnaliser un entrepôt de données à base du profil utilisateur

Le système de personnalisation dans ce cas repose sur les besoins, les préférences et les caractéristiques des utilisateurs [Ioannidis and Koutrika, 2005], et généralement sur des profils d'utilisateurs définis. Il est mentionné précédemment qu'il n'existe pas de consensus pour la définition d'un profil utilisateur, mais un profil comprend généralement un ensemble de fonctionnalités utilisées pour configurer ou adapter le système à l'utilisateur. Ainsi, le système fournit des résultats personnalisés et efficaces [Domshlak and Joachims, 2007] adaptés à un profil utilisateur.

D'autres recherches utilisent les préférences des utilisateurs définies dans leurs profils [Bellatreche et al., 2005, Golfarelli, 2010, Jerbi et al., 2008] pour configurer ou adapter le système de personnalisation. Ces préférences peuvent aussi être liées à leurs contextes définissant les cadres d'application des dites préférences [Garrigós et al., 2009, Jerbi et al., 2008]. Plus tard, Jerbi [2012] donne une classification en trois catégories à base de profil utilisateur pour les mécanismes de personnalisation, comme suit:

1. Personnaliser le schéma des sources de données [Bentayeb et al., 2009, Garrigós et al., 2009], en adaptant les structures de données à des besoins spécifiques des usagers.

2. Personnalisation de la visualisation des requêtes [Bellatreche et al., 2005], ou représentation des requêtes. [Golfarelli, 2010, Jerbi et al., 2008].

3. Recommandation de requêtes OLAP [Giacometti et al., 2009] pour aider à l'exploration des entrepôts des données.

4.3 Personnaliser un entrepôt de données par recommandation

La personnalisation par recommandation est l'axe le plus émergent dans la personnalisation des entrepôts des données, elle est traitée par divers travaux tels que [Bentayeb et al., 2009, Chatzopoulou et al., 2009, Giacometti et al., 2008, 2009]. Les systèmes de recommandations s'appuient fondamentalement sur le filtrage d'informations (Information filtering), qui a pour objectif d'identifier dans un flux documentaire, les documents correspondants aux intérêts d'un utilisateur (profil utilisateur). Dans la littérature on peut distinguer trois catégories des méthodes de recommandation :

1. **Recommandation basée sur le filtrage par contenu :** elle est basée sur la comparaison entre deux profils (documents ou items et l'utilisateur) pour filtrer les documents ou les items dont la similarité sémantique est plus proche à celle présentée par le profil de l'utilisateur. Différentes fonctions de similarité peuvent être appliquées. La fonction la plus utilisée est le cosinus de similarité qui mesure le cosinus de l'angle entre le vecteur représentant le profil de l'utilisateur et le vecteur des documents/items [Adomavicius and Tuzhilin, 2005].

2. **Recommandation basée sur le filtrage collaboratif :** ce type de recommandation est dit collaboratif [Ekstrand et al., 2011], il utilise les évaluations des autres utilisateurs dans le système pour proposer des ressources appropriées pour l'utilisateur. Avec cette technique, seuls les items bien évalués par les utilisateurs peuvent être recommandés.

3. **Recommandation basée sur le filtrage hybride (Hybrid Filtering) :** combine les deux types de filtrage basé sur le contenu et le filtrage collaboratif afin de pallier les limites posées dans les deux techniques [Burke, 2002, Godoy and Amandi, 2008].

4.4 Personnaliser l'analyse OLAP dans un entrepôt de données

Nous avant cité dans la section 5.4 (Chapitre.1) que l'analyse multidimensionnelle OLAP consiste à exploiter intuitivement de gros volumes de données via des requêtes multidimensionnelles, néanmoins l'adaptation du résultat de ces requêtes aux besoins spécifiques de chaque utilisateur, et la restriction du résultat massif aux données les plus pertinentes reste un grand défi. Actuellement, ces systèmes ont peu de connaissances sur les utilisateurs. Cela va conduire indirectement à une dégradation dans la performance du processus décisionnel. De ce fait, l'intégration de l'utilisateur dans l'analyse OLAP a fait l'objet de nombreuses recherches [Bentayeb et al., 2008, Jerbi et al., 2008, Koutrika and Ioannidis, 2004, Stefanidis and Pitoura, 2008]. Ce type d'intégration permet d'afficher le contenu informationnel pertinent vis-à-vis des intérêts de l'utilisateur. Cependant, cette pertinence est définie par des éléments contextuels directement liés à l'utilisateur, tels que ses centres d'intérêts, ses préférences de recherche, etc, l'ensemble de ces éléments est stocké dans une structure appelée profil utilisateur.

Une étude comparative de certains travaux de recherche sur la personnalisation de l'analyse OLAP est présentée dans le tableau ci-dessous. Nous classifions les travaux selon l'objectif de la personnalisation mentionnée déjà dans la section 4.2 (personnalisation du schéma, personnalisation de l'interrogation ou la recommandation de requêtes OLAP) et le type de l'approche de personnalisation.

Ravat et al. [2007], proposent une approche de personnalisation de requête qui se focalise sur les habitudes d'utilisation du système par l'utilisateur. Les auteurs traitent la personnalisation au niveau de la navigation.

Giacometti et al. [2009], proposent de recommander des requêtes lors d'une analyse. Le processus d'analyse est basé uniquement sur une succession de requêtes. L'approche de personnalisation est centrée sur l'historique. Elle permet de recommander une requête MDX[1] suivante en se basant sur l'analyse en cours et l'historique des requêtes effectuées par les utilisateurs. Cette approche ne prend pas en compte les utilisateurs lors de la recommandation.

Garrigós et al. [2009], proposent une approche de personnalisation orientée sur les intérêts de l'utilisateur. Elle aide à la définition d'une requête en personnalisant le schéma de l'ED. En effet, la particularité de cette approche est que le mécanisme de personnalisation débute à la phase de conception.

[1]MultiDimensional eXpressions, http://msdn.microsoft.com/fr-fr/site/aa216767

Golfarelli [2010], présente une approche de personnalisation des tuples d'une requête OLAP. La personnalisation est effectuée en fonction des préférences de l'utilisateur. Il définit au départ les préférences souhaitées sous forme d'une requête dite personnalisées, cette dernière sera exécutée pour retourner les résultats adéquats.

Jerbi [2012], Jerbi et al. [2010], proposent une démarche de personnalisation générique basée sur les préférences de l'utilisateur. L'approche de personnalisation couvre plusieurs étapes de l'analyse OLAP. Elle permet de personnaliser les requêtes définies par l'utilisateur pour ensuite recommander les ressources par anticipation.

Sarraj et al. [2014], proposent une approche de personnalisation sémantique pour l'exploitation d'un ED. La personnalisation est effectuée en fonction du besoin, du profil de l'utilisateur et des concepts de la base de connaissances. Cette approche permet de fournir des ressources personnalisées. Une expansion de la recherche personnalisée effectuée afin de recommander les ressources susceptibles de l'utilisateur.

		[Ravat et al., 2007]	[Giacometti et al., 2009]	[Garrigós et al., 2009]	[Golfarelli, 2010]	[Jerbi, 2012, Jerbi et al., 2010]	[Sarraj et al., 2014]
Architecture	Personnalisation de requêtes	⋆			⋆	⋆	⋆
	Recommandation de requêtes		⋆			⋆	⋆
	Personnalisation du schéma de l'ED			⋆			
Type d'approche	Orientée Utilisateur / Historique	Utilisateur	Utilisateur	Utilisateur	Utilisateur	Utilisateur	Utilisateur

Table 2.1: Synthèse "Approche personnalisation"

Le tableau ci-dessus présente une synthèse sur les travaux de personnalisation dans l'analyse OLAP, nous constatons que les approches se basent majoritairement sur les util-

isateurs. Dans notre travail de thèse, on s'intéresse à la personnalisation et à la recommandation des requêtes.

5 Expansion de requête

De nos jours, les méthodes d'expansion de requêtes (Query Expansion (QE) en anglais) [Azad and Deepak, 2019] sont devenues un domaine de recherche plus attractif. Les spécialistes du Big Data considèrent qu'il s'agit d'une première étape pour améliorer l'efficacité de l'analyse des données et réduire considérablement l'espace de recherche afin d'élargir les résultats obtenus. L'expansion de requêtes consiste à ajouter des termes pertinents supplémentaires et fortement corrélés aux requêtes d'origine afin d'améliorer les performances des systèmes et de récupération d'informations.
Par exemple, si nous supposons que la requête initiale Q composée de n termes,
$Q = t1, t2, ..., ti, ti + 1, ..., tn$. La requête reformulée peut avoir deux composants :

- Des nouveaux termes $T' = t'1, t'2, ..., t'i, ..., t'k$ de la source de données $(s)D$;
- Des termes non utilisés $T'' = ti + 1, ti + 2, ..., ti + 3, ..., tn$;

La requête expansée peut être représentée comme suit :

$$Q_{exp} = (Q - T'')UT'$$
$$Q_{exp} = t1, t2, ..., ti, t'1, t'2, ..., t'k$$

Dans la définition ci-dessus, T' représente l'aspect clé. Il inclut l'ensemble des nouveaux termes significatifs ajoutés à la requête initiale de l'utilisateur pour extraire les informations les plus pertinentes et réduire les ambiguïtés.
À ce jour, plusieurs travaux de recherche ont montré que cet ensemble T' était calculé sur la base de la similarité des termes. Le choix de l'ensemble T' reste donc l'aspect clé des travaux de recherche.

Comme mentionné dans [Fernández-Reyes et al., 2018], les techniques d'expansion de requêtes peuvent être classées en méthodes globales ou locales. Dans les méthodes globales, la requête d'origine est expansée indépendamment de tout résultat d'extraction. En règle générale, les termes T' sont sélectionnés dans WordNet [Pal et al., 2014] et ils sont associés sémantiquement à la requête initiale. Dans les méthodes locales, ils utilisent le feedback de pertinence, en effectuant une première extraction, le résultats de cette extraction sera utilisé en réalité pour sélectionner les termes les plus prometteurs à ajouter à la requête initiale [Miyanishi et al., 2013, Parapar et al., 2014, Takeuchi et al., 2017].

D'autres chercheurs ont exploré le concept d'expansion des requêtes dans les avis d'utilisateurs en ligne. Bhogal et al. [2007] ont examiné l'expansion de la requête à l'aide d'une ontologie spécifique au domaine de la recherche. Les termes T' définis ci-dessus ont été sélectionnés dans l'ontologie du domaine. En revanche, Carpineto and Romano [2012] ont examiné les principales techniques d'expansion automatique des requêtes, mais leurs travaux excluent les recherches récentes sur les documents sociaux personnalisés, les méthodes de pondération et de classement des termes et la catégorisation de plusieurs sources de données. Azad and Deepak [2019] ont traité les techniques d'expansion des requêtes selon quatre aspects clés : (i) les sources de données, (ii) les applications, (iii) la méthodologie de travail et (iv) les approches de base. Récemment, les travaux des auteurs dans [Zhou et al., 2017] traitent de l'enrichissement de la requête en fonction du profil de l'utilisateur afin de faciliter l'expression de son besoin et de rendre les informations sélectionnées intelligibles et exploitables.

5.1 Méthodologie de l'expansion des requêtes

Le processus de génération d'extension de requête comprend principalement quatre étapes : prétraitement des sources de données, pondération et classement des termes, sélection des termes et reformulation de la requête (voir Figure 2.3). Chaque étape a été discutée dans les sections suivantes.

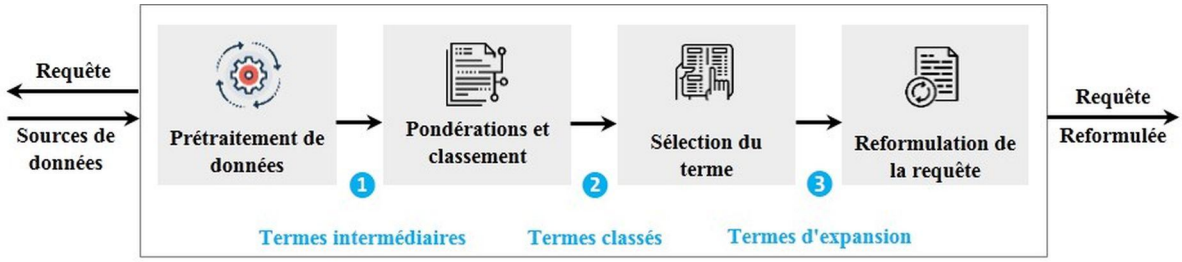


Figure 2.3: Modèle de d'expansion de requête

1. **Prétraitement de données :** le prétraitement d'une source de données dépend des sources de données et des approches utilisées pour l'expansion de la requête, et non de la requête de l'utilisateur. Le but principal de cette étape (prétraitement de la source de données) est d'extraire un ensemble de termes provenant de sources de données qui augmentent de manière significative la requête initiale de l'utilisateur. Il comprend les quatre sous-étapes suivantes :

   (a) Extraction de texte à partir de sources de données (extraction de la totalité des textes de la source de données spécifiées utilisées pour l'expansion de la requête);

(b) Tokenization (un processus de division du flux de textes en mots);

(c) Arrêtez la suppression de mots (suppression des mots fréquemment utilisés, par exemple, articles, adjectifs, prépositions, etc.);

(d) Mots dérivés (processus de réduction des mots dérivés ou infligés à leur mot de base);

2. **Pondération et classement :** Dans cette étape du QE, des poids et des rangs ont été attribués pour interroger les termes d'expansion (voir la Figure 2.3). La saisie de cette étape correspond à la requête de l'utilisateur ainsi qu'aux textes extraits des sources de données lors de la première étape. Les pondérations attribuées indiquent la pertinence des termes dans la requête étendue et sont ensuite utilisées dans le classement des documents extraits en fonction de la pertinence. Il existe de nombreuses techniques pour pondérer et classer les termes d'expansion des requêtes. Les auteurs dans [Azad and Deepak, 2019] classent les techniques en quatre catégories sur la base de la relation entre les termes de la requête et les fonctions d'expansion :

   (a) Association individuelle (One-to-One Association) : Tel que WordNet pour trouver des synonymes et des termes similaires pour les termes de la requête.

   (b) Association un à plusieurs (One-to-Many Association) : Met en corrélation un terme de requête avec de nombreux termes de requête développés.

   (c) Distribution des fonctionnalités des documents les mieux classés (Feature Distribution of Top Ranked Documents) : Traite avec les principaux documents extraits de la requête initiale et considère les termes les plus pondérés de ces documents.

   (d) Modélisation en langage de requête (Query Language Modeling) : Construit un modèle statistique pour la requête et choisit les termes d'expansion les plus probables.

   Dans notre travail, on s'intéresse à la première catégorie des techniques de pondération. Elle est considérée comme une approche de base pour pondérer et classer les termes d'expansion en fonction de l'association individuelle entre les termes d'interrogation et les termes d'expansion. Dans cette catégorie, chaque terme d'extension est connecté à un terme de requête individuelle et des poids sont attribués pour chaque terme de requête à l'aide de plusieurs techniques.

   En revanche, l'une des approches les plus influentes pour l'établissement d'une association individuelle consiste à utiliser des associations linguistiques, à savoir l'utilisation d'un thésaurus. WordNet [Voorhees, 1994] est l'un des thésaurus les plus célèbres :

chaque terme de la requête est associé à ses synonymes et à un ensemble de mots similaires, obtenus à partir de WordNet, dans la requête expansée.

Ensuite, chaque terme expansé se voit attribuer un score de similarité basé sur sa similarité avec le terme de la requête initiale. Seuls les termes avec des scores élevés sont conservés dans la requête expansée.

Dans la littérature, il existe plusieurs approches pour déterminer la similarité sémantique des termes, mais la sélection de la mesure de distance appropriée est l'un des défis rencontrés par les professionnels et les chercheurs lorsqu'ils tentent de calculer dans un jeu de données. La variété des mesures de similarité peut être source de confusion et de difficultés pour choisir une mesure appropriée. Les mesures de similarité peuvent donner des résultats différents pour des jeux de données de dimensions différentes [Carpineto and Romano, 2012], Dans ce travail nous focalisons notre choix sur la similarité cosinus, plus de détail sera présenté dans la section 5.2.

3. **Sélection du terme :** dans la section précédente, la pondération et le classement des conditions d'expansion ont été effectués. Après cette étape, les termes les mieux classés sont sélectionnés pour le développement de la requête.

4. **Reformulation de la requête :** il s'agit de la dernière étape de l'expansion de la requête, où la requête expansée est reformulée pour obtenir de meilleurs résultats lorsqu'elle est utilisée pour extraire des documents pertinents. La reformulation est effectuée sur la base des pondérations attribuées aux termes individuels de la requête étendue.

5.2 Représentation vectoriel et les mesures de similarité

Actuellement, plusieurs disciplines telles que l'analyse de données et la recherche de l'information sont utilisées en intégrant l'évaluation par des mesures de similarité pour l'analyse de données textuelles. Dans chacun de ces domaines, les similarités sont utilisées pour différents traitements :

- En analyse de données textuelles, les similarités sont utilisées pour la description et l'exploration de données ;
- En recherche d'information, l'évaluation des similarités entre documents et requêtes est utilisée pour identifier les documents pertinents par rapport à des besoins d'information exprimés par les utilisateurs.

En revanche, le calcul de la similarité entre les documents textuels effectués selon des techniques qui évidemment varient d'une discipline à une autre. Mais les documents textuels s'intègrent cependant le plus souvent dans une même approche générale en deux temps : (a) Les documents textuels sont d'abord associés à des représentations spécifiques dans un espace vectoriel de grande dimension. Les documents textuels sont représentés comme des vecteurs de caractéristiques représentant les termes qui apparaissent dans la collection. (b) Le calcul de similarités est subi à un modèle mathématique choisi par l'expert du domaine pour mesurer les similarités.

(a) **Représentation vectorielle :** la représentation d'un document sous forme vectorielle se déroule en deux étapes :

- **Extraction des termes pertinents :** il s'agit de prétraiter le texte des documents textuels en supprimant les mots-vides, la ponctuation, etc.
- **Calcul des poids :** le poids de chaque terme dans un document peut être obtenu de différentes manières : booléenne, fréquence des termes, Tf-Idf (Term Frequency - Inverse Document Frequency).

(b) **Calcul de similarités :** une mesure de similarité est, en général, une fonction qui quantifie le rapport entre deux objets comparés en fonction de leurs points de ressemblance et de dissemblance. Les deux objets comparés sont, bien entendu, de même type. Dans la littérature, plusieurs types de mesure de similarité existants mais ils ne sont pas des métriques. Pour être une métrique, une mesure doit satisfaire les 4 conditions suivantes :

Soit x, y et z, trois éléments d'un ensemble, et soit $d(x, y)$ la distance entre x et y.

- Positivité : $d(x, y) \geqslant 0$.
- Principe d'identité des indiscernables : $d(x, y) = 0 \equiv x = y$.
- Symétrie : $d(x, y) = d(y, x)$.
- Inégalité triangulaire : $d(x, z) \leqslant d(x, y) + d(y, z)$.

[Huang, 2008] et [Strehl et al., 2000] ont tous montré que les performances de la similarité cosinus est significativement meilleure que celles d'autre comme la distance euclidienne. Par ailleurs, la similarité cosinus est fréquemment utilisée [Baeza-Yates et al., 1999] en tant que mesure de ressemblance syntaxique. Elle permet de comparer des documents textuels en se basant sur les chaines de caractères qui les composent. Il s'agit de calculer le cosinus

de l'angle entre les représentations vectorielles des documents à comparer. La similarité obtenue $sim_{cosinus}(d1, d2) \in [0, 1]$.

$sim_{cosinus}(d1, d2) = \frac{\vec{d_1}.\vec{d_2}}{\|\vec{d_1}\|\|\vec{d_2}\|}$. Où $d1$ et $d2$ ces sont des documents.

5.3 Avantages et inconvénients liés au modèle vectoriel

Dans le tableau ci-dessous, nous tentons de lister (de manière non exhaustive) les avantages et les inconvénients liés au modèle vectoriel.

- Avantages	- Quelque soit la technique utilisée, basée sur le modèle vectoriel, a le même format initial, à savoir la représentation vectorielle. - Les techniques basées sur le modèle vectoriel sont faciles à développer, il s'agit uniquement de calcul vectoriel.
- Inconvénients	- Des mots identiques considérés comme peu pertinents peuvent parfois trop influer sur la valeur de la similarité. Par exemple, pour les phrases "Tout est bien qui finit bien" et "C'est notre seul bien", le terme "est" n'est pas vraiment pertinent et pourtant, il va avoir un poids certain.

Table 2.2: Avantages et inconvénients du modèle vectoriel

Notons cependant que le calcul du pois par Tf-Idf cité dans la section 5.2 et l'élimination des mots-vides permettent de pallier cet inconvénient.

5.4 Classification des approches de l'expansion de requêtes

Comme mentionnée dans la section ci-dessus, l'expansion de la requête consiste à ajouter de nouveaux termes à la requête d'origine afin d'augmenter le nombre d'informations récupérées correspondant aux besoins de l'utilisateur. Le problème important de l'expansion de la requête est le choix des mots-clés d'extension établis sur la requête initiale. Dans la littérature plusieurs approches ont été proposées afin de classifier les approches d'expansion de requêtes. Azad and Deepak [2019] ont proposé une classification d'analyse en deux groupes : globale et locale. Dans cette section on s'intéresse à l'analyse globale et plus précisément aux approches linguistiques.

Les approches de cette catégorie analysent les caractéristiques d'expansion telles que les relations de termes, sémantiques et syntaxiques pour reformuler ou développer les termes

de requête initiaux. Ils utilisent un thésaurus, des dictionnaires, des ontologies, le cloud LOD (Linked Open Data) ou d'autres ressources de connaissances similaires telles que WordNet. Ce dernier, est un thésaurus bien connu pour développer la requête initiale de l'utilisateur. [Fang, 2008] a montré que les techniques d'expansion de la requête utilisant WordNet ont enrichi les performances d'analyse. Cette méthode montre qu'une utilisation appropriée de WordNet peut en effet aider à obtenir des résultats utiles via l'expansion de requêtes.

Comme indiqué précédemment, de nombreux travaux de recherche utilisent WordNet pour enrichir la requête initiale. L'auteur dans [Voorhees, 1994] utilise WordNet pour rechercher les synonymes. Les auteurs dans [Smeaton et al., 1995] utilisent WordNet et POS Tagger pour développer la requête initiale. De même, Hsu et al. [2006] utilisent ConceptNet (diversité de concept supérieure) et WordNet (capacité de discrimination supérieure) en tant que sources de données pour l'expansion de la requête utilisateur. Pal et al. [2014] proposent un moyen nouveau et efficace pour utiliser WordNet pour l'expansion des requêtes, où les termes d'expansion candidats sont sélectionnés dans un ensemble de documents pseudo-pertinents d'où l'utilité de ces termes est déterminée en considérant plusieurs sources d'information. Le tableau ci-dessous (2.3) présente une synthèse sur certains travaux de recherche où, WordNet est utilisé comme une source de données pour l'expansion de la requête initiale de l'utilisateur.

Références	Sources des Termes	Méthodologie d'extraction de terme	Représentation des termes
[Voorhees, 1994]	WordNet	Hyponymes de la requête	Termes individuels
[Smeaton et al., 1995]	WordNet & Thesaurus	Sens des mots	Termes individuels
[Hsu et al., 2006]	ConceptNet & WordNet	Termes ayant le même concept	Termes individuels
[Pal et al., 2014]	WordNet	Synonymes, antonymes et homonymes des termes de la requête	Termes individuels

Table 2.3: Synthèse sur l'expansion de la requête utilisateur

En règle générale, l'utilisation de WordNet pour l'expansion d'une requête n'est utile que si les mots de la requête sont de nature non ambiguë [Gonzalo et al., 1998].

6 Filtrage d'informations et systèmes de recommandation

Sur l'ère du Big Data, le volume important de données et la diversification des sources de données ont posé le problème de surcharge d'informations. Il est presque quasiment impossible de cibler dès le départ de l'analyse l'information exacte qui reflète le besoin réel de l'utilisateur, et ce est dû au nombre énorme de l'information. Il est donc nécessaire de filtrer, de hiérarchiser les priorités et de diffuser efficacement les informations pertinentes afin d'atténuer le problème de la surcharge d'informations.

En revanche, le filtrage d'informations (ou Information Filtering (IF)) est l'une des méthodes utiles pour apporter des solutions à l'explosion de l'information. Dans la littérature, lorsque les informations fournies sous forme de suggestions, un système du filtrage d'informations est appelé un système de recommandation (Recommender System (RS)). Ce dernier représente des outils logiciels et des techniques qui fournissent des recommandations pour des produits ou des services pouvant plaire à un consommateur particulier [Das et al., 2017].

Les systèmes de recommandation ont les capacités de prédire si un utilisateur particulier préférera un élément ou non en fonction de son profil. Ils sont avantageux à la fois pour les fournisseurs de services et les utilisateurs [Pu et al., 2011]. Ils réduisent les coûts de transaction liés à la recherche et à la sélection des informations dans un environnement d'achat en ligne par exemple [Hu and Pu, 2009]. Les systèmes de recommandation se sont également avérés pour améliorer le processus de prise de décision et sa qualité [Pathak et al., 2010]. Ils sont définis comme une stratégie de prise de décision pour les utilisateurs dans des environnements d'information complexes. En outre, le système de recommandation permet aux utilisateurs de rechercher dans des enregistrements de connaissances liés à leurs intérêts et à leurs préférences. Récemment, une certaine revue de littérature, montre que plusieurs diverses approches pour la construction de systèmes de recommandation ont été développées, peuvent utiliser un filtrage collaboratif, un filtrage basé sur le contenu ou un filtrage hybride [Acilar and Arslan, 2009, Chen et al., 2008, Jalali et al., 2010].

6.1 Filtrage collaboratif basé sur le contenu

Par ailleurs, le filtrage basé sur le contenu (Content-Based Filtering -CBF-) est l'une des approches les plus simples pour personnaliser les suggestions ou les recommandations concernant un produit ou un service, en fonction du comportement précédent de l'utilisateur et du comportement d'autres utilisateurs partageant les mêmes idées. Le volume important des données et la diversification des sources de données ont posé des problèmes avec les algorithmes du système de recommandation, tels que le temps de calcul long et les performances médiocres en temps réel [Tewari et al., 2018].

Les méthodes d'apprentissage appliquées sur le filtrage basé sur le contenu essaient de trouver les informations les plus pertinentes en fonction des préférences de l'utilisateur. Une telle approche utilise le profil de l'utilisateur et la description de ses préférences pour comprendre les intérêts de l'utilisateur [Aggarwal, 2016, Bobadilla et al., 2013]. Le filtrage basé sur le contenu utilise différents types de modèles pour trouver des similitudes entre les documents, afin de générer des recommandations significatives. Il pourrait utiliser un modèle d'espace vectoriel, tel que la fréquence de document-inverse de fréquence (TF / IDF), ou des modèles probabilistes tels que le classificateur Naïve Bayes [Friedman et al., 1997], les arbres de décision [Duda et al., 2012] ou les réseaux de neurones [Bishop, 2006].

6.2 Méthode de pondération (Term Frequency-Inverse Document Frequency)

C'est le calcul le plus couramment utilisé dans les applications de traitement de texte et de récupération d'informations. Il s'agit d'une quantité statistique utilisée pour mesurer l'importance d'un mot par rapport à un corpus de documents. Le terme fréquence du terme i dans le document j est donné par l'équation suivante, où $n_{i,j}$ est le nombre d'occurrence du terme t_i dans le document d_j, $tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$.

La somme dans le dénominateur donne le nombre d'occurrence de tous les termes du document. La fréquence de document inverse mesure l'importance du mot en le comparant à son caractère commun d'occurrence dans d'autres documents. Plus précisément, la fréquence de document inverse est donnée par : $Idf_i = \log \frac{|\mathrm{D}|}{|d_j : t_i \in d_j|}$, où $|D|$ est le nombre total de documents et $|d_j : t_i \in d_j|$ est le nombre de documents où le terme t_i apparait. Finalement, le poids s'obtient en multipliant les deux mesures : $tfidf_{i,j} = tf_{i,j} \times idf_i$

6.3 Calcul de "Term Frequency" et "Inverse Document Frequency" dans le framework MapReduce

Afin de mettre cela dans le framework MapReduce, le problème est divisé en quatre tâches :

1. **Calculer la fréquence des mots dans un document**
   (a) Ceci est identique au premier exemple de cette section. La fonction *Mapper* prend en entrée *(nomdocument, contenu)* et en sortie *((terme, nomdocument),1)*.
   (b) La fonction *reducer* additionne les comptes de chaque mot du document et génère les résultats *((term, docname), n)*.
2. **Calculez le nombre de mots pour les documents**
   (a) La fonction *Mapper* prend en entrée *((term, docname), n)* et en sortie *(docname, (term, n))*.
   (b) La fonction *reducer* additionne les fréquences de chaque n dans le même document et envoie également les données d'origine à partir de la fonction *Mapper*. Ceci génère *((term, docname), (n, N))* où n est la fréquence du terme et N la longueur du document.
3. **Trouver la fréquence des mots dans le corpus de document**
   (a) Pour cela, la fonction *Mapper* prend en entrée *((term, docname), (n, N))* et en sortie *(term, (docname, n, N, 1))*, transmettant ainsi également les données déjà calculées.
   (b) La fonction *reducer* additionne les comptages du mot dans le corpus de document et génère les résultats *((terme, nomdocument), (n, N, m))*.
4. **Le travail final consiste à calculer la valeur TF-IDF**
   (a) Pour cela, la fonction *Mapper* prend en entrée *((term, docname), (n, N, m))* et calcule le TF-IDF en tant que $(n/N) * log(D/m)$ où D est la taille du corpus de document. D peut être supposé ou se trouve dans un autre cycle simple de *MapReduce*. Les sorties du *mappeur ((term, docname), TF * IDF)*.
   (b) La fonction *reducer* est dans ce cas une fonction d'identité.

6.4 Avantages et inconvénients du filtrage à base du contenu

Les techniques de filtrage basées sur le contenu surmontent les défis du filtrage collaboratif. Elles ont la possibilité de recommander de nouveaux articles même s'il n'y-a pas d'évaluation fournie par les utilisateurs. Ainsi, même si la base de données ne contient pas les préférences de l'utilisateur, la précision des recommandations n'est pas affectée. En outre, si les préférences de l'utilisateur changent, il est en mesure d'adapter ses recommandations en peu de temps. Elles peuvent gérer des situations dans lesquelles des différents utilisateurs ne partagent pas les mêmes éléments, mais uniquement des éléments identiques en fonction de leurs caractéristiques intrinsèques. Les utilisateurs peuvent obtenir des recommandations sans partager leur profil, ce qui garantit la confidentialité [Lam et al., 2006]. La technique CBF peut également fournir des explications sur la manière dont les recommandations sont générées pour les utilisateurs. Cependant, les techniques souffrent de divers problèmes, comme discuté dans la littérature [Adomavicius and Tuzhilin, 2005]. Les techniques de filtrage basées sur le contenu dépendent des métadonnées des éléments. En d'autres termes, ils ont besoin d'une description détaillée des éléments et d'un profil utilisateur très bien organisé avant de pouvoir faire des recommandations aux utilisateurs. C'est ce qu'on appelle l'analyse de contenu limité. L'efficacité du CBF dépend donc de la disponibilité de données descriptives.

7 Conclusion

Dans ce chapitre, nous avons présenté les éléments fondamentaux liés au profilage et la contextualisation, la personnalisation, l'expansion des requêtes utilisateur ainsi que le filtrage d'information dans les systèmes de recommandation. A travers l'enchaînement des différentes sections que nous avons présenté, et la diversité des travaux de recherches existant, nous constatons que la technique de personnalisation à base de profil utilisateur et l'expansion des requêtes utilisateur sont capables d'améliorer la qualité des résultats d'analyse dans les systèmes d'analyse multidimensionnelle OLAP dans le contexte du Big data.

Par ailleurs, la personnalisation dans les systèmes de recommandation ouvre de nouvelles possibilités de recherche d'informations personnalisées sur l'ère du Big Data. Cela contribue également à atténuer le problème de surcharge d'informations, phénomène très courant dans les systèmes d'analyse de Big Data. Dans le cadre de cette thèse, nous souhaitons apporter des contributions pour améliorer le processus de l'analyse OLAP dans le contexte du Big Data en introduisant la technique de filtrage collaboratif tel que TF/IDF.

Part II

Contributions

Chapter 3

Une approche basée sur le profil de l'utilisateur et le contexte de la recherche pour la reformulation des requêtes

1 Introduction

Les techniques d'expansion par reformulation de requêtes sont devenues un domaine de recherche plus attractif dans les dernières années. Elles ont pour but d'exploiter le profil de l'utilisateur pour reformuler sa requête initiale en y intégrant des éléments de son centre d'intérêt ou de ses préférences de recherche.

Dans ce chapitre nous présentons une contribution qui traduit un point de vue relatif à la personnalisation d'un ensemble de données volumineuses par l'intégration des éléments du profil utilisateur. Notre contribution [Menaceur et al., 2017a], consiste à proposer une nouvelle approche, qui permet de mettre en œuvre une technique d'analyse et de personnalisation capable d'effectuer des opérations analytiques multidimensionnelles rentables sur des données volumineuses. L'architecture du système est basée principalement sur les techniques de personnalisation citées dans le deuxième chapitre en intégrant le contexte de la requête de recherche utilisateur et les éléments contextuels stockés dans son profil. Un espace de données réduit a été construit et orienté vers la génération des cube OLAP personnalisés relatifs au besoin contextuel et au profil de l'utilisateur.

2 Prise en compte du contexte utilisateur dans la reformulation de la requête utilisateur

Aujourd'hui, face à la croissance énorme du volume de données, l'utilisateur se trouve souvent dans un processus d'analyse incapable de localiser son besoin exact en information. Il utilise une requête initiale formulée par ces propres termmmes, qui retourne souvent des

résultats dans la totalité est moins proche à son besoin réel.

A cet effet, la reformulation de la requête consiste donc à modifier la requête de l'utilisateur par ajout de termes significatifs. Cette idée d'affinement de requêtes n'est pas nouvelle. Comme nous avons détaillé dans le chapitre 2., plusieurs approches utilisant de différentes techniques pour sélectionner les termes à rajouter à une requête. Trois types d'approches sont distinguées pour la reformulation de requête, la différence entre ces approches réside, soit dans la source des termes utilisés dans la reformulation ou dans la ressource terminologique (réseau sémantique, thesaurus ou ontologie), soit dans le mécanisme qui permet de sélectionner les termes à ajouter à la requête initiale (probabiliste ou lien sémantique). L'approche de reformulation que nous proposons réside dans deux types d'approches que nous avons présenté précédemment. Nous utilisons le profil de l'utilisateur comme une source de termes et le calcul du lien sémantique comme mécanisme pour le choix des termes à ajouter.

2.1 Paramètres du profil utilisateur pour une approche de personnalisation

Notre choix s'est fixé sur l'utilisation du contexte de recherche et le profil de l'utilisateur pour la reformulation de la requête utilisateur. Nous avons présenté dans la section 2.2 du chapitre 2., les différentes formes du profil et les caractéristiques relatives à chaque forme. En se basant sur cette classification des profils, nous avons adopté une structure composée de deux catégories de classes, ces dernières représentent soit un contexte statique ou un contexte dynamique. Les paramètres caractérisant notre approche en termes de profil utilisateur sont inspirés depuis le travail de [Bouramoul, 2011]. Le tableau 3.1 présente selon le modèle de Bouramoul [2011] les paramètres et les différentes manières selon lesquelles le contexte est utilisé pour aider à la reformulation des requêtes.

2.1.1 Implication de l'utilisateur

L'utilisateur intervient en partie dans la définition de son profil. A la fin de chaque session d'analyse, le système récupère des résultats qui lui semblent pertinents par rapport au besoin et exigence définis dans la session d'analyse. L'utilisateur valide par la suite les résultats qu'il juge réellement pertinents parmi l'ensemble de propositions.

2.1.2 Moment de la reformulation

Il s'agit d'utiliser le profil utilisateur à la Pré-reformulation, et à la Post-reformulation. Le système dans ce cas-là, reformule le besoin de l'utilisateur en affinant l'expression de sa

Type de paramètre	Valeur possible	Valeur choisie
Selon l'implication de l'utilisateur	Directe	+
	Indirecte	
Selon la nature d'information	Profil d'identification	+
	Profil d'interrogation	+
Selon le moment de la reformulation	Pré-reformulation	+
	Post-reformulation	+

Table 3.1: Paramètres du profil utilisateur

requête selon deux étapes indépendantes.

2.1.3 Nature d'information

Nous utilisons à la fois et d'une façon complémentaire, un profil d'identification et un profil d'interrogation. Le premier sert à identifier un utilisateur à travers une série d'informations définies à la première connexion au système. Le deuxième est issu de l'historique des recherches faites par le même utilisateur dans des sessions antérieures, donc son contenu se développe à chaque fois que l'utilisateur procède à une nouvelle session d'analyse.

3 Présentation de l'architecture proposée

Notre proposition s'articule autour de cinq couches pour permettre la reformulation de la requête utilisateur en se basant sur son profil. Il s'agit dans un premier temps de la couche de contextualisation responsable de capturer le contexte de la requête de recherche utilisateur, ensuite le contexte nécessaire à la catégorisation de l'utilisateur. les résultats de cette couche serons utilisés dans la couche de profilage et de requêtage par le module de personnalisation pour générer des nouvelles requêtes à partir de la requête initiale. Ensuite, les nouvelles requêtes obtenues précédemment sont exploitées par la couche OLAPing pour la personnalisation de l'espace de recherche et la construction de cube OLAP. Enfin la couche d'analyse de données prend en charge la délivrance du résultat qui se rapproche le mieux aux besoins de l'utilisateur par l'analyse multidimensionnelle OLAP. Nous décrivons dans ce qui suit chacune de ces couches en donnant ses différents composants et son principe de fonctionnement. Le regroupement de ces cinq couches nous a permis par la suite de définir notre architecture pour la reformulation des requêtes à base de profils.

3.1 Couche externe

C'est la couche la plus proche aux utilisateurs du système. Elle fournit une interface de communication avec l'univers extérieur du système, elle permet de capturer le comportement de l'utilisateur ainsi que son besoin de recherche, et le transmettre vers la couche juste supérieure (couche contextualisation).

3.2 Couche de contextualisation

Cette couche est utilisée principalement pour la formulation des contextes définis dans notre système. Nous avons choisi de considérer deux types de contextes représentés dans notre système sous forme de deux concepts clé, à savoir :

- ***Contexte utilisateur :*** il peut être assimilé à tous les facteurs qui peuvent décrire les intentions de l'utilisateur et les perceptions de son environnement.
- ***Contexte de la requête :*** comprend l'intégration des connaissances linguistiques et sémantiques à la requête utilisateur afin d'explorer la compréhension la plus exacte des besoins d'information de l'utilisateur.

3.3 Couche de profilage et requêtage

La couche profilage et requêtage s'articule autour de deux modules, afin de permettre la reformulation multiple de la requête initiale de l'utilisateur, en se basant sur son profil et ses préférences de recherche. Il s'agit dans un premier temps de capturer le contexte de profil utilisateur par le module de profilisation, puis de les utiliser par le module de requêtage pour générer de nouvelles requêtes dites reformulées et enrichies, et ce, à la base de la requête initiale. Le regroupement de ces trois modules nous permet de présenter l'organigramme de la Figure 3.1.

3.4 Couche OLAPing

La couche OLAPing et la partie la plus importante de notre système, elle nous donne la possibilité de construire des cubes OLAP personnalisés selon le profil et les préférences de recherche de l'utilisateur. Un certain nombre de mesures doivent être prises avant de procéder à la création des cubes de données, à savoir :

- Migrer à l'aide de *Apache Sqoop* [Jain, 2013] les données stockées dans les plates-formes traditionnelles vers la plate-forme *Hadoop* [White, 2012] en utilisant *Apache*

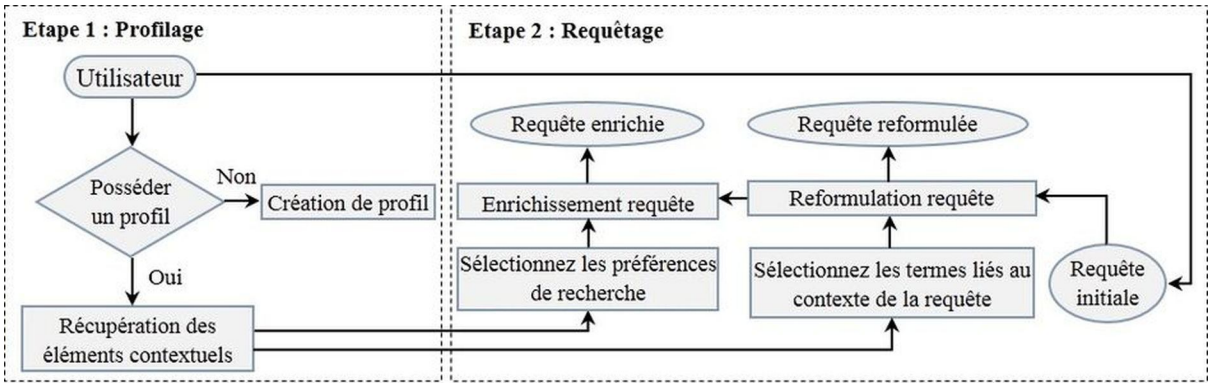


Figure 3.1: Organigramme de l'exécution du processus de profilage et de requêtage

Hive [White, 2012] comme une solution d'entreposage de données open source.

- L'utilisateur conçoit son besoin d'analyse dans la couche précédente, il met deux types de requêtes, où, chacune d'elles sera utilisée pour un contexte particulier. La requête reformulée par l'utilisation des termes plus proches au terme de la requête initiale utilise *HiveQL* [White, 2012] comme un moteur de requêtage, pour examiner l'ensemble de données stocké dans *Apache Hive*. Une *Data Mart* est construite dans *Apache Hive* rassemble les données versées dans le contexte de la requête définie dans la couche contextualisation.

Les cubes OLAP sont créés à partir du *Data Mart* stocké dans l'environnement Apache Hive par l'utilisation de *Apache Kylin* [Apache, 2015] qui est un moteur d'analyse distribué open source conçu pour fournir une interface *SQL* et une analyse multidimensionnelle (OLAP) sur *Hadoop*.
Dans *Apache Kylin*, chaque cube OLAP doit être soumis aux étapes suivantes lors de sa création :

- Charger toutes les métadonnées nécessaires des tables Data Mart de Hive Metastore.
- Calculer les cardinalités de chaque colonne du tableau, où un modèle de données est défini.
- Dans la phase de création du modèle, les tables de faits, les tables de recherche, les nouvelles conditions de jointure, etc., sont sélectionnées en fonction des préférences de recherche de la requête enrichie dans la couche profilage et requêtage

Le cube est maintenant défini et prêt à être construit. *Apache Kylin* commence à travailler en interne en interrogeant d'abord les tables *Apache Hive*, en récupérant les résultats de

ces tables et en stockant les résultats sous la forme de *HTable* dans *HBase* [White, 2012]. Ce processus peut prendre du temps en fonction de la taille des données.

3.5 Couche d'analyse de données

Dans cette couche une analyse multidimensionnelle est déclenchée sur la base des dimensions des cubes OLAP générés par *Apache Kylin*. A la fin de chaque session d'analyse un module de capture du contexte dynamique procèdera à l'extraction d'un ensemble des éléments relatifs au contexte de l'utilisateur. Ce dernier, valide ceux qu'il juge réellement pertinents et les ajoutent dans la base contextuelle de l'utilisateur.

4 Description générale de l'approche

La composition des cinq couches décrites précédemment, nous a permis de définir l'architecture générale de notre approche. Nous signalons que le fonctionnement des cinq couches est étroitement lié dans le sens où, les sorties de chaque couche sont les entrées de la couche suivante. La figure 3.2 présente l'architecture générale de notre système pour la personnalisation de l'analyse OLAP, on utilise le profil de l'utilisateur et le contexte de sa requête pour la reformulation des requêtes.

En revanche l'utilisateur, avant de lancer sa requête, s'identifie dans le système qui procède alors à la récupération de son Contexte Statique, il s'agit de ses caractéristiques personnelles pouvant influencer le contexte de l'analyse. Ces renseignements ont été enregistrés dans la Base des Contextes Utilisateurs lors de la première connexion au système. Dans le cas d'un utilisateur qui ne possède pas un profil, le système lui demande de remplir ses préférences et la Base des Contextes Utilisateurs sera mise à jour pour une éventuelle utilisation dans des prochaines sessions d'analyse.

Une fois le Contexte Statique est récupéré, l'utilisateur peut alors formuler sa requête, et le système procède à la personnalisation de l'espace de recherche. Le module de personnalisation de requête se charge de générer une nouvelle requête dite reformulée en sélectionnant les termes relatifs au contexte de la requête initiale, cette sélection est faite à partir de la Base des Contextes Utilisateurs. Les deux types de contextes (Statique et Dynamique), contribuent donc mutuellement à l'opération de reformulation. Par la suite, le système lance un filtrage sur l'espace de recherche, en utilisant la requête reformulée précédemment et qui été écrite par *HiveQL*. Les résultats du filtrage sont retournés à l'utilisateur sous forme d'un *Data Mart* qui représente un espace de recherche personnalisé. Le module personnalisation de requête dans une deuxième étape enrichie la requête reformulée précédemment par les préférences de recherches déclarées par l'utilisateur, une nouvelle

requête dite enrichie est générée.

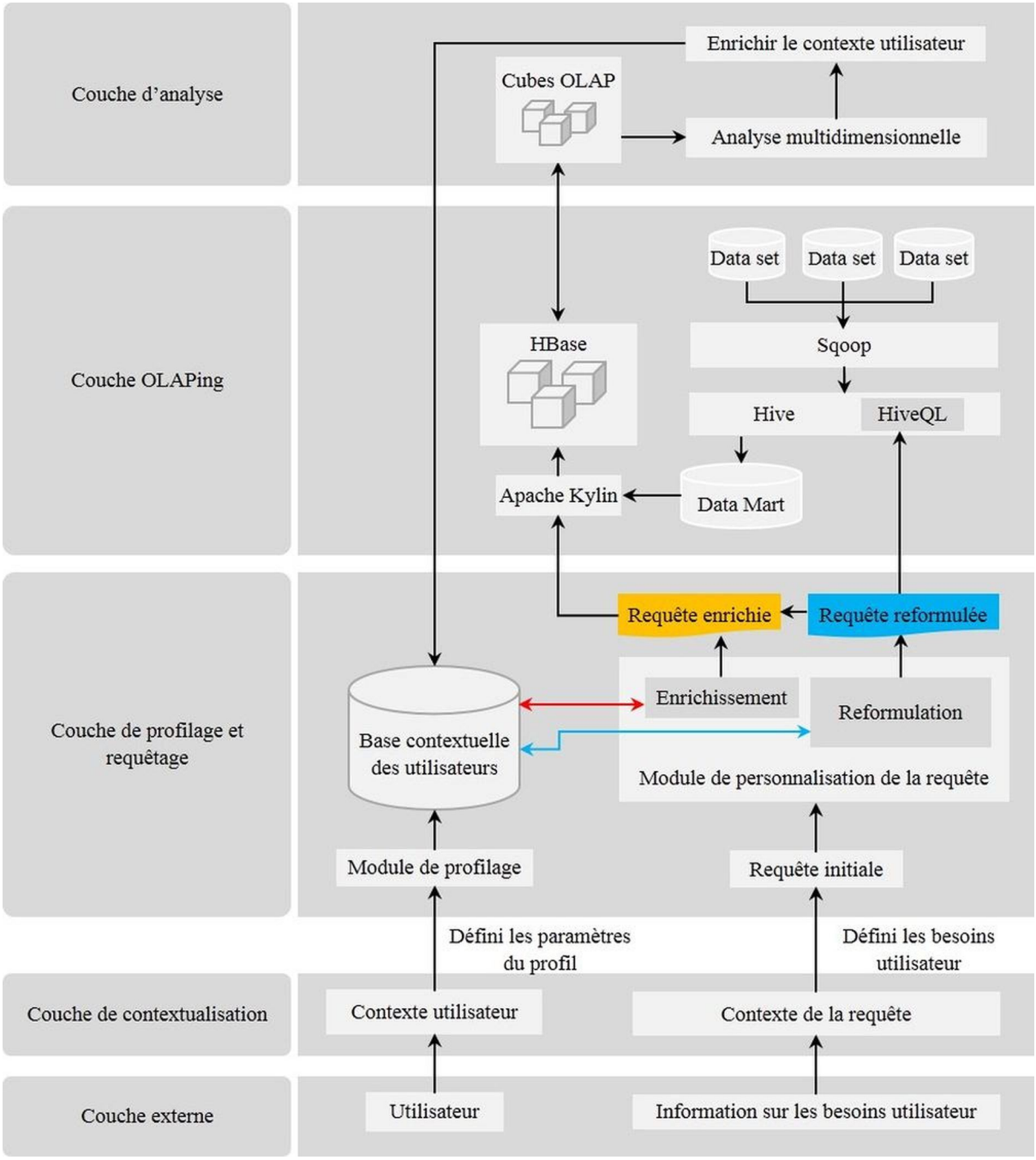


Figure 3.2: Approche pour la reformulation des requêtes

Dans la couche juste supérieure (OLAPing) le système utilise le moteur d'analyse *Apache Kylin* qui est conçu pour fournir une interface SQL et une analyse multidimensionnelle (OLAP) pour l'exécution de la requête enrichie précédemment. Les résultats sont stockés également dans *HBase* pour être utilisés par la suite dans la couche d'analyse.

A la fin de chaque session d'analyse dans la couche d'analyse, le contexte de l'utilisateur est mis à jour et stocké dans la base contextuelle de l'utilisateur.

5 Expérimentations et tests

Afin de montrer l'applicabilité de l'approche proposée, nous avons mis en place un environnement de stockage et de traitement basé sur la plateforme Hadoop en intégrant *Apache Hive, Apache Sqoop, Hbase et Apache Kylin.*

De plus, pour simplifier la configuration et l'installation de notre environnement de test, nous utilisons Hortonworks Sandbox 2.4[1] qui est une plateforme de données préconfigurées où *Apache Hive* et préinstallé avec d'autre outils de Hadoop. L'installation *d'Apache Kyline* est destinée à la création des cubes de données et faite manuellement et configurée aussi avec *Hbase* pour le stockage des résultats d'analyse pour la constitution des cubes OLAP. Les parties (structure) du logiciel Hortonworks sont visibles à la figure 3.3.

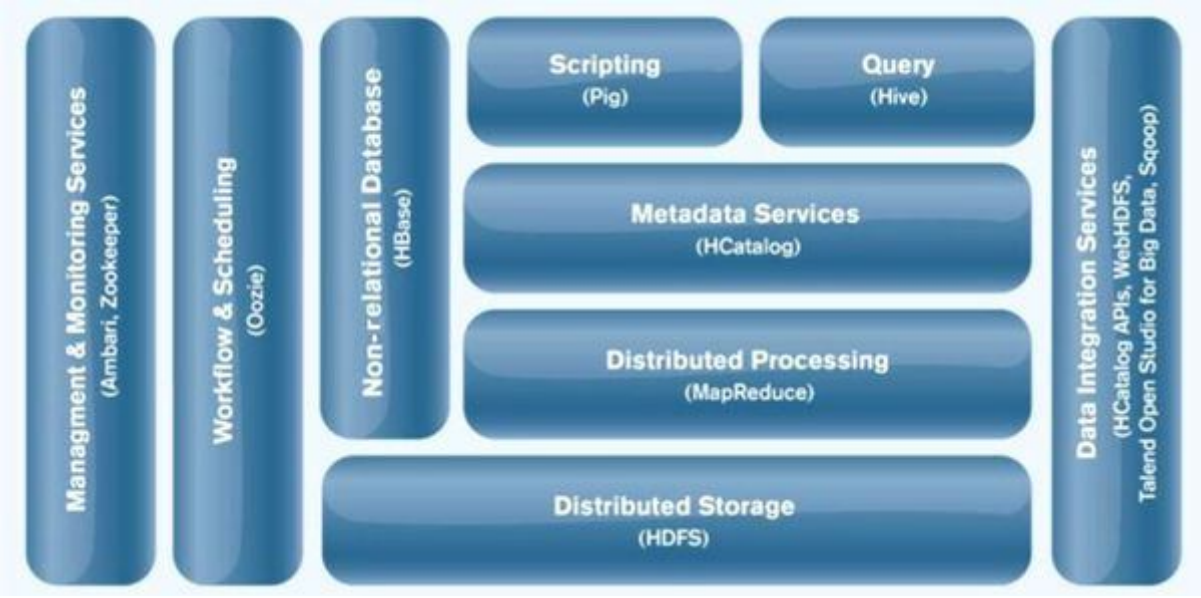


Figure 3.3: Plateforme de données Hortonworks [Blagov et al., 2015]

[1]https://fr.hortonworks.com/tutorials/

Pour évaluer l'applicabilité de notre architecture, il faut assurer l'intercommunication entre les différents outils de l'environnement de test que nous avons choisi, ce choix est motivé par la popularité de l'environnement dans la communauté Big data et l'efficacité des outils lors du traitement et d'analyse des données.

5.1 Description du jeu de données

Afin de tester notre approche, nous avons choisi d'utiliser un jeu de données réelles relatives au transport et à la distribution des produits pétroliers d'une société algérienne. Cet ensemble de données reprend les données des quatre dernières années de la commercialisation et de la distribution de produits pétroliers. Les données sont collectées mensuellement à partir des sites et des points de vente de la société et stockées dans son datacenter à l'aide d'un outil d'Extract Transform Load (ETL). La structure de l'ensemble de données est plus compliquée, nous choisissons une partie de l'ensemble où, la structure de la base de données est représentée par le schéma relationnel cité ci-dessous. De plus, la description statistique de l'ensemble de données est présentée dans le tableau 3.2.

$Centre\_Distribution(IdCentre, type, region, ville, adresse)$
$Station(IdStation, nom, Type, region, ville, adresse, service, IdCentre)$
$Produit(IdProduit, TypeProduit, Prix, Qts, IdStation)$
$Livreur(IdLivreur, Regien, Ville, IdStation)$
$Livraison(NumLivraison, DateLivraison, IdStation)$

Données brutes	Détails
Nombre d'entités	31
Nombre d'enregistrement	1.321.897

Table 3.2: Description statistique du jeu de données

5.2 Scénario de validation

- **Étape 1 : Reformulation en fonction du contexte de recherche** Soit la requête initiale Q formulée par les termes de l'utilisateur :

  Q : *Recherchez les stations-service les plus proches du centre-ville.*

  Ce dernier est transformé selon le modèle relationnel cité ci-dessus en une requête SQL, comme suit :

  Select Station.nom, Station.adresse From Station
  Where (region='Centre-Ville');

L'exécution du script SQL mentionné ci-dessus montre que les résultats obtenus contiennent seulement les noms et les adresses des stations-services. Ceci n'est pas souhaitable pour l'utilisateur, parce qu'il a besoin d'un ensemble de services pouvant être indisponibles dans certaines des stations qui ont été choisies, ce qui nécessite une nouvelle recherche.

La requête initiale de l'utilisateur sera transmise au module de personnalisation situé dans la couche de profilisation et requêtage pour une étape de reformulation à base des termes situés dans la base contextuelle des utilisateurs. Ici, le concept qui peut être inséré à la requête initiale c'est *Livreur*, la requête reformulée devienne:

$Q1$: *Trouver les stations-services et les livreurs les plus proches au centre-ville.*

Une fois la requête est reformulée, elle sera envoyée à un moteur de requêtage pour l'exécution. A la fin un *Data Mart* est construit en regroupant seulement les informations liées aux stations dans la région est Centre-ville et dans le programme de livraison est sont livrées. La requête reformulée est écrite par HQL (Hive query language) comme suite :

Select Station.nom, Station.adresse, Livreur.nom, Livreur.IdStation
From Station, Livreur
Where (region='Centre-Ville');

- **Etape 2 : Enrichissement à base de profil utilisateur** Maintenant, soit un utilisateur ayant les préférences de recherche suivantes, qui sont stockées dans son profil :
  - Il cherche un lubrifiant pour sa voiture.
  - Il ne veut pas dépenser plus de 1000 DA.
  - Il aime prendre des boissons à la station.
  - Il cherche un service de lavage après la vidange de sa voiture.

  Ces préférences peuvent être exprimées avec l'ensemble de conditions suivantes :
  - *Produit.TypeProduit = Lubrifiant.*
  - *Produit.Prix <= 1000.*
  - *Station.Type = A.*
  - *Station.Service = lavage.*

L'ensemble de ces préférences est envoyé au module personnalisation pour une étape d'enrichissement. La requête enrichie $Q2$ par le profil de l'utilisateur est la suivante :

Select Station.nom, Station.adresse, Livreur.nom, Livreur.IdStation
From Station, Livreur, Produit
Where((Station.IdStation=Produit.IdStation)and (Produit.Prix<=1000)
and (Produit.Type=Lubrifiant)) and ((Station.Type='A')
and (Station.Service='lavage')) ;

La requête enrichie $Q2$ sera exécutée sur l'ensemble des données du *Data Mart* auparavant obtenues, ou en utilisant *Apach Kylinc* pour la création des cubes de données stockées dans *HBase*. Les dimensions des cubes de données sont l'ensemble des préférences de recherche.

5.3 Expérimentation

Dans cette étude, notre objectif est d'évaluer et de comprendre l'impact de l'utilisation du profil utilisateur et le contexte de la recherche dans l'analyse OLAP. Pour ce faire, nous utilisons deux techniques de personnalisation selon deux étapes (reformulation basée sur le contexte de recherche et enrichissement basé sur le profil de l'utilisateur) afin de créer un cube OLAP personnalisé qui refait le contexte de l'utilisateur dans une session d'analyse. Pour ce faire, l'évaluation de l'efficacité des résultats obtenus à l'issue des jeux de test est effectuée selon deux types de critères :

(a) **Le temps : qui englobe :**

- Le temps de migration de la base MySQL vers Hive.
- Le temps de reformulation de la requête ainsi que le temps de construction de Data Mart.
- Le temps d'enrichissement de la requête ainsi que le temps de création des cubes OLAP sur Hbase.

(b) **La taille de Data Mart et la taille des cubes OLAP**

Le contenu principal de ce test consiste à comparer les impacts des requêtes reformulées et enrichies sur les performances de la création de cubes OLAP sur *Apache Kylin* en étudiant la taille du magasin de données générées, ainsi que la taille et le temps de création des cubes OLAP.

Le tableau 3.3 représente les résultats obtenus après les jeux de tests effectués sur trois jeux de données de différentes tailles tel que : 500 Mo, 700 Mo et 1500 Mo.

Requête	Taille			Taille DataMart	Taille du cube
		Hive	Kylin		
initiale	500 Mo	205.32 s	1.02 s	80 %	100 %
reformulée		248.76 s	0.97 s	51 %	63.75 %
reformulée & enrichie		0.23 s			25 %
initiale	700 Mo	450.25 s	1.14 s	79 %	100 %
reformulée		521.24 s	1.03 s	57 %	72.15 %
reformulée & enrichie		0.57 s			29 %
initiale	1500 Mo	805.45 s	1.24 s	83 %	100 %
reformulée		870.28 s	1.12 s	69 %	83.13 %
reformulée & enrichie		0.81 s			35 %

Table 3.3: Temps d'exécution des requêtes

- **Test de performance de la requête reformulée**

Le test consiste à exécuter directement une requête HQL reformulée sur les données stockées dans Apache Hive afin d'élargir l'espace de recherche de l'utilisateur, et de créer un magasin de données plus proche du contexte de recherche de l'utilisateur, en éliminant ainsi toutes sortes de masses d'informations inutiles.

Le tableau 3.3 montre que lorsqu'il s'agit d'une requête reformulée, la latence de la requête HQL s'aggrave avec la quantité croissante des données, est ce, en raison de l'augmentation du nombre d'enregistrements dans les tables, suite à l'élargissement de l'intervalle de recherche dans l'espace de recherche (voir figure 3.4-A-).

D'autre part, la taille du magasin de données générées présente des pourcentages significatifs par rapport à la taille de l'ensemble de données initial, ce qui démontre l'utilité de la reformulation pour garantir que la requête initiale est ouverte à d'autres concepts (voir figure 3.4 -B-).

Toutefois, le délai de réponse du moteur Apache Kylin lors de la création du cube OLAP diminue de cinq millisecondes par rapport au temps de création du cube OLAP lors de la requête initiale, ce qui démontre que le temps de réponse du moteur Apache Kylin

(voir figure 3.4 -C-) et la taille du cube OLAP (voir figure 3.4 -D-) sont fortement liés à la taille du magasin de données générée après la reformulation.

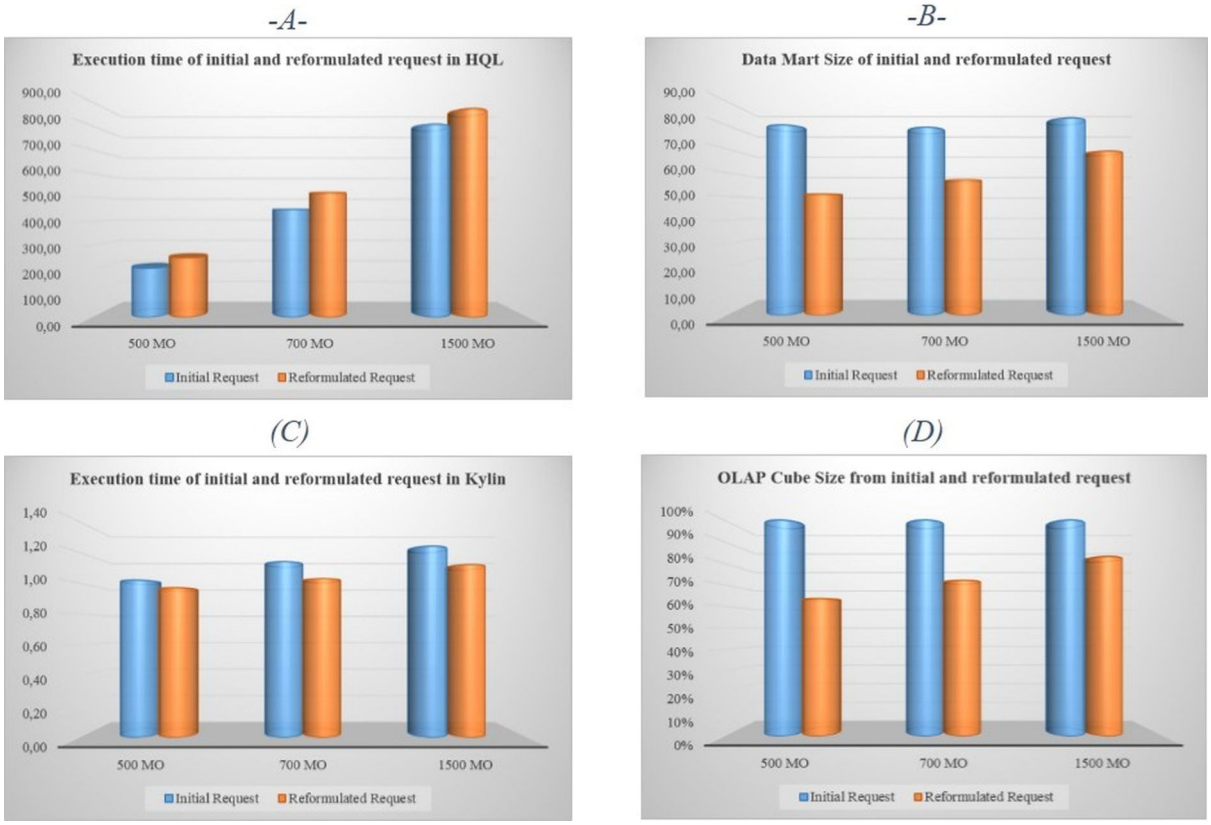


Figure 3.4: Test de performance de la requête reformulée

- **Test de performance de requête enrichie**

Dans ce cas, le test consiste à enrichir la requête HQL précédemment reformulée en ajoutant les éléments des préférences de recherche stockés dans le profil de l'utilisateur. L'analyse des résultats obtenus dans le tableau 3.3 montre que le temps de réponse du moteur Kylin lors de la création du cube OLAP est diminué par rapport au temps de réponse lors de la création du cube OLAP dans le cas de la requête reformulée (voir la figure 3.5 -A-).

Ce test nous a également permis de constater que la taille des cubes générés continuait de diminuer après l'étape d'utilisation du profil utilisateur. Le nombre d'enregistrements dans le cube OLAP est réduit et seuls les enregistrements liés au profil utilisateur sont conservés. Ce type de filtrage démontre son importance pour limiter l'espace de recherche (voir la figure 3.5 -B-).

Figure 3.5: Test de performance de la requête enrichie

6 Conclusion

Dans cette première contribution, nous avons proposé une approche pour la reformulation de la requête basée sur le contexte de recherche et le profil de l'utilisateur. Cette proposition s'articule principalement autour de cinq couches afin de permettre la personnalisation des cubes OLAP. Cette architecture offre deux optimisations majeures : l'ouverture de la requête de l'utilisateur à d'autres concepts, par l'intégration des nouveaux termes utilisant la technique de reformulation et la restriction de l'espace de recherche par le filtrage en fonction du profil de l'utilisateur.

La performance de l'architecture est évaluée. Les résultats expérimentaux montrent que, pour les données volumineuses, le délai de génération de cubes OLAP avec le moteur Apach Kylin est très rapide après des requêtes personnalisées afin d'augmenter son degré d'ouverture. La taille des cubes générés par Apach Kylin est plus petite en utilisant le filtrage basé sur le profil de l'utilisateur.

Chapter 4

Une architecture pour la personnalisation de l'analyse dans le Big Data

1 Introduction

Dans un contexte de Big Data, de nombreuses approches d'amélioration ont été introduites afin d'améliorer la réponse temporelle dans les traitements analytiques multidimensionnels. Le volume, la vitesse et la variété des sources des données ont créé de nombreux nouveaux défis dans les techniques d'analyse, et qui rendent difficile l'extraction des besoins spécifiques des utilisateurs depuis leurs requête initiale. Afin de résoudre ce problème, nous proposons dans ce chapitre une approche de personnalisation dynamique dans le contexte de Big Data où, on utilise OLAP pour l'analyse multidimensionnelle, cette approche utilise l'expansion de requêtes et le filtrage basé sur le contenu comme des techniques de personnalisation et de filtrage de l'information.

Notre proposition [Menaceur et al., 2019] consiste en premier lieu à traiter la requête initiale de l'utilisateur par une technique d'enrichissement, afin d'intégrer les éléments contextuels de son profil et le contexte de recherche comme première étape, et ce, pour réduire l'espace de recherche dans le cube OLAP, ensuite, utiliser la technique d'expansion des requêtes sur la requête précédemment enrichie pour étendre la portée de l'analyse dans le cube OLAP. Les résultats obtenus sont : "aussi pertinents que possible" par rapport à la demande initiale de l'utilisateur. Par ailleurs, nous utilisons les techniques de filtrage d'informations telles que le filtrage basé sur le contenu pour personnaliser l'analyse dans le cube de données réduit en fonction de la fréquence des termes et de la similarité des cosinus. Enfin, nous présentons une expérimentation selon une étude de cas pour évaluer et valider notre contribution.

2 Requêtes expansées et filtrage basé sur le contenu pour la personnalisation de l'analyse multidimensionnelle

Dans les deux dernières décennies, il y a eu une forte demande pour la conception des nouveaux modèles, techniques, algorithmes et plates-formes informatiques afin de prendre en charge les problèmes d'exploration et d'analyse dans le contexte du Big Data, d'autant plus que ces données relèvent de plusieurs classifications telles que, les données structurées et non structurées. Actuellement, de nombreuses nouvelles études préfèrent étudier et analyser des données non structurées telles que les recommandations en ligne des clients (Online Customer Reviews), afin d'extraire des informations importantes et utiles pour la prise de décision du client, en s'appuyant sur une analyse multidimensionnelle.

En revanche, l'analyse de ces recommandations à grande échelle basée sur des requêtes par mots clés est devenue une tâche fastidieuse, et nécessite des efforts extrinsèques pour réduire les données à une taille raisonnable. Cette technique donne souvent des résultats indésirables en raison de : (i) les mots-clés soumis par l'utilisateur peuvent être liés à plusieurs sujets, ce qui donne des résultats qui ne sont pas centrés sur le sujet qui vous intéresse. (ii) La requête peut être trop courte pour exprimer correctement ce que l'utilisateur recherche. (iii) L'utilisateur est souvent incertain de ce qu'il recherche jusqu'à ce qu'il voie les résultats. (iv) L'utilisateur sait ce qu'il recherche, mais il ne sait pas comment formuler la requête appropriée.

Pour surmonter ces problèmes, nous cherchons à travers cette contribution à concevoir une nouvelle approche combinant les techniques d'extension des requêtes et le filtrage d'information à base de contenu dans une analyse multidimensionnelle. Cette approche réduit considérablement l'espace de recherche en fonction du profil de l'utilisateur et permet de trouver les réponses les plus appropriées à la demande de l'utilisateur.

2.1 Représentation et paramétrage de profil utilisateur

Avant de présenter les détails de cette deuxième contribution, nous définissons d'abord le concept de profil utilisateur et leur différentes représentations selon le paramétrage défini dans le modèle de Bouramoul [2011], que nous avons déjà utilisé dans la précédente contribution [Menaceur et al., 2017a]. En revanche, d'après les études de l'état de l'art dans le Chapitre 2., l'ensemble des travaux menés dans le domaine du profilage ont montré empiriquement leur efficacité, ce qui nous mène à l'intégrer à nouveau dans notre approche comme un outil pour l'expansion des requêtes utilisateur afin de personnaliser l'espace de données stockées dans le cube OLAP d'une part, et d'enrichir le processus d'analyse

multidimensionnelle, d'une autre part.

Pour rendre l'utilisation des profils utile dans la reformulation des requêtes utilisateur et utilisable pour notre architecture de personnalisation dans le contexte de Big Data, nous modélisons le contexte du profil en deux grandes classes :

- **Contexte statique :** il prend les caractéristiques d'un profil d'identification étendu qui sera capturé dans une étape de pré-reformulation, et qui se caractérise par une implication directe de l'utilisateur (Voir Section 2.1 Chapitre 3.).
- **Contexte dynamique :** il constitue l'élément principal de notre proposition. Il regroupe les caractéristiques d'un profil d'interrogation étendu qui est utilisé dans une étape de requêtage et qui nécessite une implication directe de l'utilisateur.

2.2 Expansion des requêtes par utilisation d'une ressource externe

L'expansion des requêtes n'est pas une idée nouvelle dans le domaine de l'analyse multidimensionnelle de données. Comme nous l'avons constaté dans le Chapitre 2., l'expansion par l'utilisation d'une ressource externe consiste à analyser premièrement la requête utilisateur pour détecter les termes les plus pertinents. Dans notre cas, nous avons exploité le contenu de WordNet pour remplacer les termes précédemment détectés par des concepts proches en utilisant des relations sémantiques avec un lien de synonymie (voir Figure 4.1).

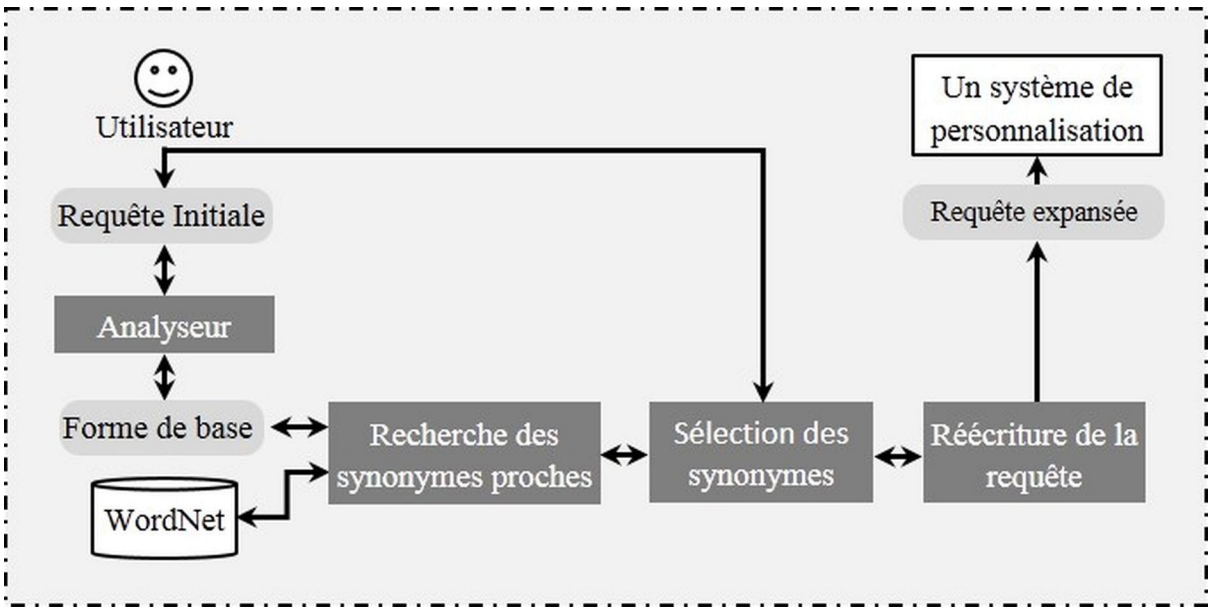


Figure 4.1: Schéma synoptique de l'expansion de requête

Le succès de cette méthode était lié à deux éléments : la qualité de la requête initiale et celle des synonymes trouvés [Voorhees, 1994]. Quand la requête initiale exprime déjà correctement le besoin en information, l'ajout de synonymes n'a pas vraiment d'intérêt pour la performance. D'un autre côté, l'ajout automatique (sans désambiguïsation) des synonymes dégrade la performance de la recherche. Par conséquent, Voorhees [1994] a réussi à améliorer les mauvaises requêtes par l'ajout de synonymes choisis manuellement dans WordNet. Le choix manuel des sens par l'utilisateur qui a créé la requête est idéal pour garantir une bonne désambiguïsation [Audeh, 2014].

2.3 Filtrage d'information basé sur le contenu

Comme indiqué déjà dans le Chapitre 2., l'objectif principal d'un système de filtrage d'informations est de filtrer un flux entrant d'informations de façon personnalisée pour chaque utilisateur, tout en s'adaptant en permanence à son besoin d'informations. Le filtrage basé sur le contenu utilisé dans notre cas, a pour objectif de filtrer les résultats obtenus après une étape de personnalisation en fonction de la fréquence des termes (TF-IDF), et de la similarité des cosinus (Cosine Similarity). La figure 4.2 présente le schéma général du filtrage d'information dans notre cas. En revanche, cette technique de filtrage offre un double avantage pour les décideurs, elle répond aux intérêts à long terme des utilisateurs d'une part, et permet au profil utilisateur d'évoluer naturellement par la restriction progressive sur les thèmes qui l'intéressent.

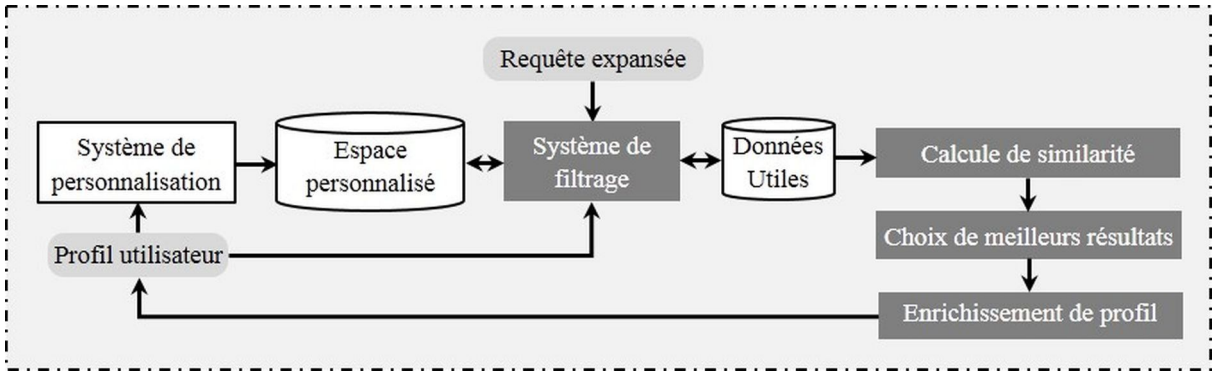


Figure 4.2: Schéma général du filtrage d'information

3 Personnalisation de l'analyse multidimensionnelle dans le contexte du Big Data

L'objectif principal de l'étude proposée est de mettre en œuvre une technique d'analyse multidimensionnelle améliorée pour la réduction et la personnalisation de données volumineuses telles que les recommandations en ligne des clients (Online Customer Reviews). Cette technique est capable de réduire considérablement l'espace de recherche pour effectuer des opérations d'analyse multidimensionnelles rentables.

L'architecture proposée repose principalement sur deux éléments complémentaires : (i) les techniques d'expansion des requêtes et (b) le filtrage des informations. Dans un premier temps, et dans un module de requêtage, nous utilisons les éléments de préférence de l'utilisateur stockés dans son profil pour créer une requête dite enrichie non fonctionnelle 'No Functional Enriched Query' afin de réduire l'espace de recherche dans le cube OLAP et d'extraire un ensemble de données dit 'Good Data'. Dans la deuxième étape, nous incorporons des mots équivalents (synonymes) à partir d'une source externe telle que WordNet pour tout ou partie des mots de la requête originale de l'utilisateur afin de créer une nouvelle requête plus significative dite expansée fonctionnelle 'Functional Expansion Queries', cette dernière va permette de personnaliser la recherche dans le cube OLAP réduit 'Good Data' en utilisant les techniques de filtrage à base de contenu pour extraire les données les plus utiles 'Useful Data'. À partir de ces résultats, le système mis à jour la base contextuelle des utilisateurs. En revanche, l'architecture de notre approche est composée de quatre modules principaux : (a) Module de profilage, (b) Module de requêtage, (c) module du traitement OLAP, et finalement un (d) Module d'analyse et de reporting. L'architecture générale de notre approche qui est illustrée dans la figure 4.3, où les quatre modules cités ci-après sont présentés en détails.

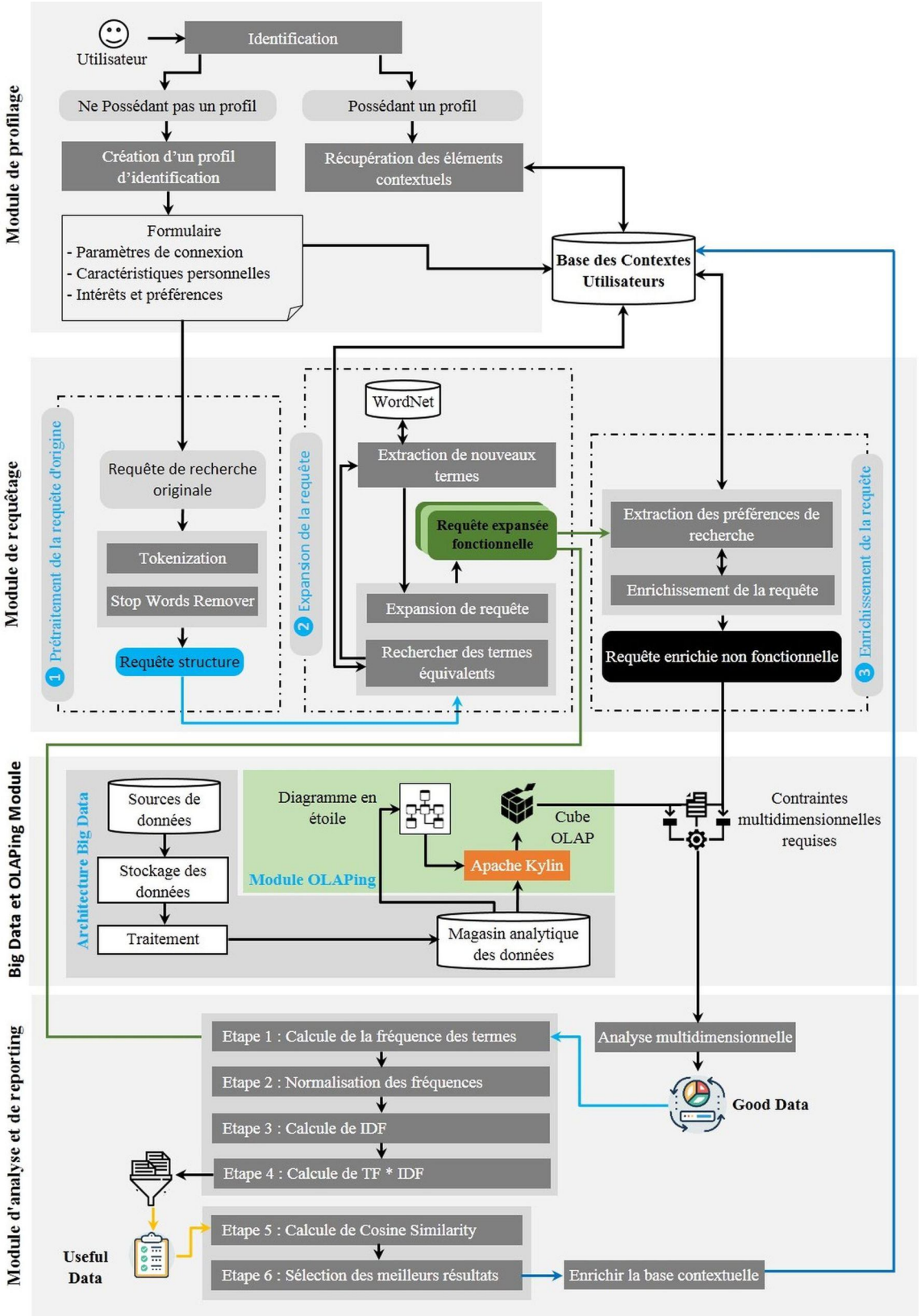


Figure 4.3: Architecture pour la personnalisation de l'analyse dans le Big Data

3.1 Module de profilage

Pour obtenir de meilleurs résultats de recherche en termes de précision, nous utilisons une approche d'analyse personnalisée dans laquelle l'ambiguïté de la recherche peut être réduite et où, les résultats de l'analyse ont plus de chances d'être intéressants pour l'utilisateur. Le contexte utilisateur est un facteur clé dans le profilage de l'utilisateur. Il peut être assimilé à tous les facteurs pouvant décrire les intentions de l'utilisateur et les perceptions de son environnement. Dans notre travail, le contexte du profil utilisateur est divisé en deux types :

- **Module pour la capture du contexte statique**

  Ce module permet de collecter un ensemble d'informations liées à l'utilisateur et définies lors de la première connexion avec le système (voir Table 4.1). À cette fin, nous avons défini trois catégories d'informations relatives au contexte statique, ces informations se résument en :

  (a) *Les paramètres de connexion* : e-mail, mot de passe.

  (b) *Les caractéristiques personnelles* : nom, prénom, pays, langue ...etc.

  (c) *Les intérêts et préférences* : domaine, domaine secondaire, spécialité ...etc.

Algorithme : contexte statique
Entrée : paramètres de connexion, intérêts, préférences,... ; **Sortie :** contexte statique du profil ;
1. Création identification profil ;

Table 4.1: Capturation du context statique

La figure 4.4 présente les éléments du contexte statique, et la manière selon laquelle les informations composant ce type de contexte sont capturées.

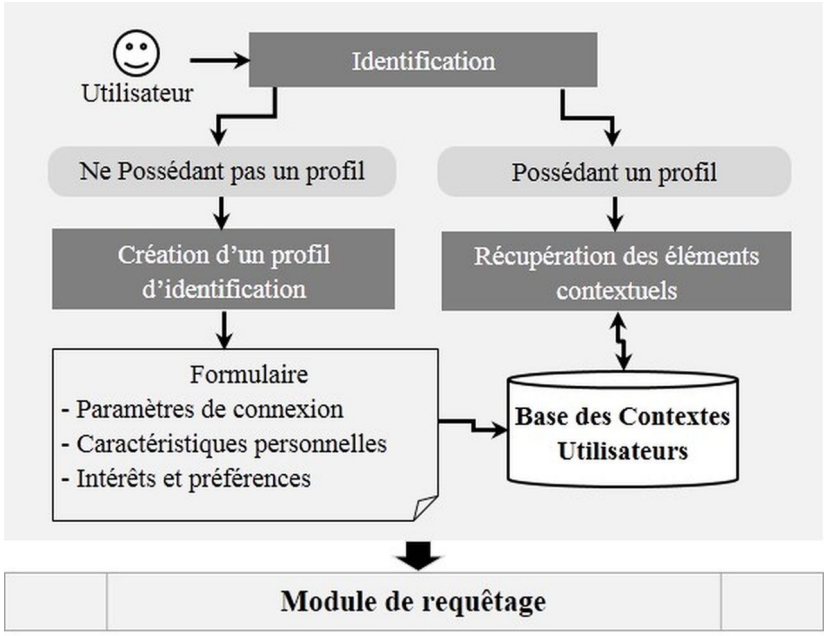


Figure 4.4: Module pour la récupération du contexte statique

- **Module pour la capture du contexte dynamique**

  La partie dynamique du profil utilisateur est mise à jour à la fin de chaque session d'analyse, où, le processus de capture de contexte dynamique extrait depuis les résultats d'analyse les éléments les plus pertinents au contexte utilisateur. La Table 4.2 montre le processus de la capture de contexte dynamique.

Algorithme : contexte dynamique
Entrée : Requête de recherche originale ; **Sortie :** Les éléments pertinents pour le contexte ;
1. Requêtage ; 2. OLAPing dans Big Data ; 3. Analyse et reporting ; 4. Extraction des éléments pertinents pour le contexte ;

Table 4.2: Capturation du contexte dynamique

La Figure 4.5 présente la manière selon laquelle les éléments du contexte dynamique sont capturés.

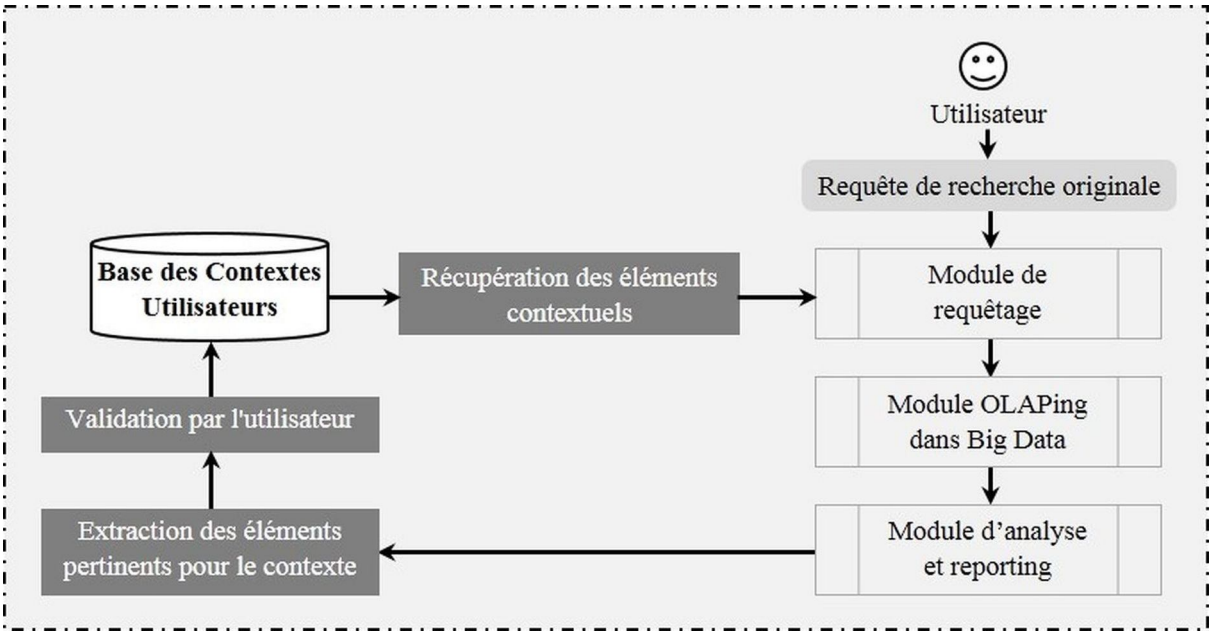


Figure 4.5: Module pour la récupération du contexte statique

3.2 Module de requêtage

La requête originale de l'utilisateur doit subir une transformation en trois étapes afin de pouvoir extraire les résultats 'aussi pertinents que possible' de l'ensemble de données. Les étapes du module de requêtage sont décrites ci-dessous.

- **Étape 1 : Prétraitement de la requête originale**

Dans cette étape, la requête originale de l'utilisateur est traitée selon deux phases complémentaires, la première est dite `Tokenization`, elle prend en entrée un texte tel qu'une phrase et génère un ensemble de termes ou mots. La deuxième phase est `StopWordsRemover`, elle suit la phase `Tokenization` et est basée sur une séquence de chaînes comme paramètres d'entrées (par exemple, la sortie de la phase `Tokenization`), elle permet de répertorier et supprimer tous les mots rares et vides. Dans notre champ d'étude la requête utilisateur est exprimée en anglais, les mots anglais inutiles pour la recherche d'informations sont appelés mots vides (par exemple les `stop words` incluent *the, as, of, and, or, to*, etc., en anglais).

Algorithme : Prétraitement de requête
Entrée : Q ; **Sortie :** Tokens ;

Etape 1 : Pour chaque entrée Q ;
`Tokenization (EWi)=` Q ; // *Extraction de tous les Termes, i=1, 2, 3...n dans EWi*
Etape 2 : Pour chaque EWi ;
`StopWordsRemover (SWRi) =EWi` ; // *appliquer le processus d'élimination des Stop Word pour supprimer tous les mots vides*
Etape 3 : Pour chaque SWRi ;
`Freq_Count (WCi)= SWRi ;`
`Return (SWRi) ;`
Etape 4 :
`Tokens (SWRi) ;`

Table 4.3: Prétraitement de la requête utilisateur

Cette phase est très essentielle dans l'étape de prétraitement de la requête originale, car elle présente certains avantages : elle réduit la taille du requête utilisateur en matière de termes, et améliore également l'efficacité globale de la recherche dans une session d'analyse.

Si nous prenons l'exemple Q : *Amazing location and great for business travelers* comme une requête utilisateur, et après avoir appliqué l'algorithme présenté dans la Table 4.3, les résultats obtenus sont présentés dans la Table 4.4.

	Liste Tokens				
$SWRi$	Amazing	Location	great	business	travellers
WCi	1	1	1	1	1

Table 4.4: Requête utilisateur après prétraitement

- **Étape 2 : Enrichissement de la requête**

Exploiter les préférences des utilisateurs est un élément crucial de notre approche de personnalisation. A cet effet l'étape d'enrichissement de la requête nous permet d'améliorer l'efficacité de la requête utilisateur prétraitée précédemment, par l'intégration des nouvelles préférences souhaitées par l'utilisateur, afin d'éviter toute sorte d'informations inutiles ou

bruitées, et donc améliorer la performance et l'exactitude des résultats obtenus dans une session d'analyse.

En revanche, dans le processus de prise de décision, les résultats sont plus précis et complets si les préférences de l'utilisateur sont exprimées de manière très précise. Par exemple, un utilisateur recherche une réservation dans un hôtel avec les préférences suivantes : *(warm welcome, change of bed linen every day, comfortable bed, good size bathroom, and well equipped, rich and varied breakfast).*

Toutes ces préférences peuvent être représentées comme suit : $P = a1, a2, a3, ..., ak$ où $a1, a2, a3, ..., ak$ désigne un ensemble d'attributs dans lesquels nous représentons les conditions de filtrage selon les préférences ajoutées par l'utilisateur. Ces nouvelles préférences de recherche sont stockées dans la partie dynamique du profil utilisateur et utilisées pour générer une nouvelle requête dite NFEQ (No Functional Enriched Queries), utilisée prochainement pour la personnalisation des données stockées dans le cube OLAP afin d'avoir les "Good Data" (voir Figure 4.3)

- **Étape 3 : Expansion de la requête**

Après avoir traité la requête d'origine de l'utilisateur et extrait tous les termes effectifs du processus d'analyse multidimensionnelle attendus sur le cube OLAP, nous avons besoin d'une technique pour renforcer le poids des termes et étendre la portée de l'analyse dans le cube OLAP, afin que les résultats obtenus soient plus précis et complets. L'expansion de requête est l'une des techniques les plus sollicitées qui tentent d'augmenter la probabilité d'une correspondance entre la requête de l'utilisateur et les données pertinentes, en ajoutant des termes sémantiquement liés (appelés termes d'expansion) à la requête de l'utilisateur [Mandal et al., 2007].

Dans notre travail, nous utilisons WordNet dans sa version anglaise en tant que source externe d'expansion des requêtes, pour combiner les synonymes des termes avec les termes de la requête de l'utilisateur, afin d'avoir des nouvelles requêtes similaires. Ces dernières sont appelées FEQ (Functional Expansion Queries) et seront utilisées prochainement dans le module d'analyse et de reporting pour le filtrage des "Good Data", et extraire les "Useful Data" répond au besoin de l'utilisateur. La Figure 4.6 décrit le fonctionnement coopératif entre les trois étapes du module de requêtage détaillées précédemment.

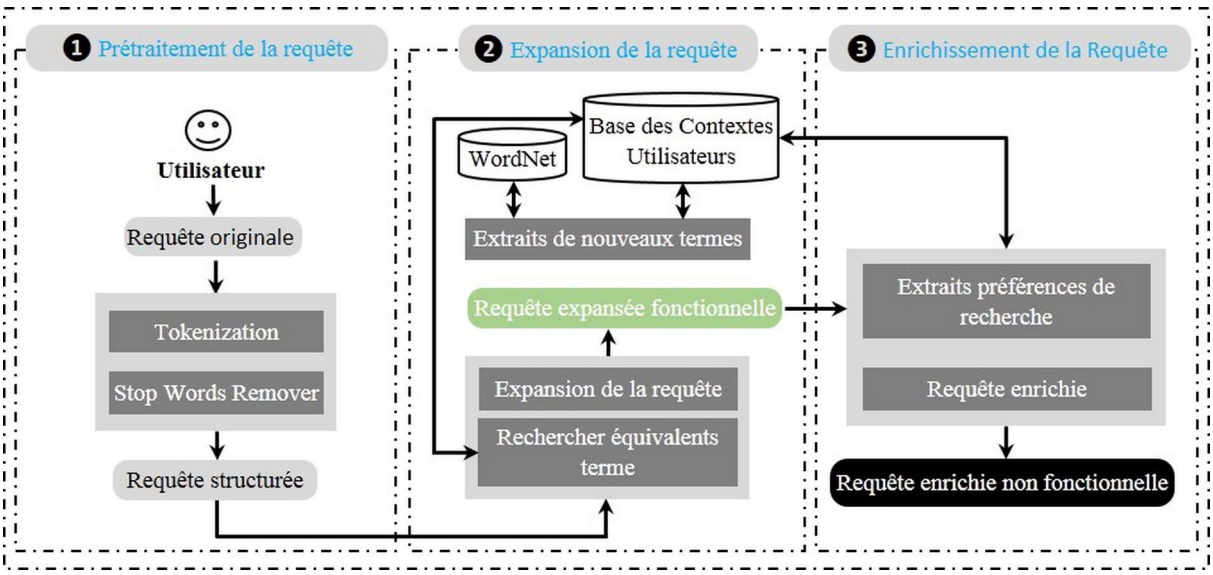


Figure 4.6: Requêtage et exploration des préférences de l'utilisateur

3.3 Module OLAPing

Le module OLAPing utilisé dans ce travail est basé principalement sur une architecture Big Data à quatre niveaux. Dans ce qui suit, nous détaillons tous d'abord les quatre niveaux de l'architecture Big Data utilisés dans notre approche.

1. **Sources de données :** toutes les solutions Big Data contiennent une ou plusieurs sources de données, chacune représentant une énorme quantité de données. Une solution Big Data peut être générée à partir de diverses sources, telles que: les systèmes intelligents, les e-mails, les transactions, les requêtes de recherche, les avis de clients en ligne, les audios, les vidéos, les réseaux sociaux, les fichiers multimédias, etc.

2. **Stockage de données :** le stockage de gros volumes de données utilise généralement un magasin de fichiers distribué pouvant contenir de gros volumes de fichiers volumineux dans divers formats. La vague initiale d'outils de stockage de données volumineuses s'est concentrée sur Hadoop et MapReduce. Maintenant, beaucoup se dirigent vers de nouvelles plateformes comme Apache Spark.

3. **Le traitement par lots :** dans toutes les solutions Big Data, le filtrage et l'agrégation des grands volumes de données à des fins d'analyse doivent utiliser des exécutions de travaux par lots. Ces dernières incluent la lecture des données dans les jeux de données sources, leur traitement et l'extraction des jeux de données réduites. Ce traite-

ment implique l'utilisation de divers outils tels que HIVE, Pig lors de l'utilisation de tâches U-SQL ou de tâches Map/Reduce dans un écosystème Hadoop, ou l'utilisation de programmes Java, Scala ou Python dans Apache Spark.

4. **Magasin de données analytiques :** dans toutes les solutions, le magasin de données d'analyse est utilisé pour stocker les données prétraitées dans les étapes précédentes dans un format structuré, qui peut être interrogé à l'aide d'outils analytiques. Ces outils d'analyse diffèrent en fonction de la nature du magasin de données d'analyse. Les données peuvent également être interrogées sous forme de base de données interactives telles que Spark SQL, Hive ou via une technologie de base de données NoSQL telle que HBase.

En revanche, exécuter une requête dans un cube de données OLAP multidimensionnel est la partie la plus importante de notre approche. Un certain nombre d'étapes doivent être franchies avant de procéder à l'exécution de la dite requête, à savoir :

- Migrer les données stockées sur les plates-formes traditionnelles vers la plate-forme Hadoop à l'aide d'Apache Sqoop. Par exemple, Apache Hive est intégré à Hadoop et utilisé comme solution d'entreposage de données.
- L'utilisateur a déterminé son besoin d'analyse dans la phase précédente. il formule deux types de requêtes où, chacune d'elles sera utilisée dans un contexte particulier.

Dans Apache Kylin qui est un moteur d'analyse distribuée open source conçu pour fournir une interface SQL et une analyse multidimensionnelle (OLAP) sur Hadoop, chaque cube OLAP doit être soumis aux étapes suivantes lors de sa création :

- Charger toutes les métadonnées nécessaires à partir des tables du dataset de Hive Metastore.
- Calculer les cardinalités de chaque colonne du tableau dans lequel un modèle de données est défini.
- Dans la phase de création du modèle, les tables de faits, les tables de recherche, les nouvelles conditions de jointure, etc., sont sélectionnées en fonction des préférences de recherche de la requête enrichie dans le module de requêtage.

La structure de cube est maintenant définie et prête à être exploitée. Apache Kylin commence à travailler en interne en interrogeant d'abord les tables Hive du magasin analytique

de données, en récupérant les résultats à partir de ces tables et en les stockant en tant que Htable dans HBase. Ce processus peut prendre du temps en fonction de la taille des données. Finalement, le cube OLAP est créé à partir du magasin de données analytiques à l'aide d'Apache Kylin.

3.4 Module d'analyse et de filtrage

En effet, le traitement dans ce module est subdivisé en deux phases complémentaires, à savoir :

1. **Analyse multidimensionnelle personnalisée :** dans cette phase, l'exécution de la *requête enrichie non fonctionnelle* (voir module requêtage) dans un système d'analyse multidimensionnelle OLAP, nous conduit à la création des cubes personnalisés en fonction du profil de l'utilisateur et ces préférences de recherche. Les résultats obtenus rassemblent des données dites "Good Data".

   Ainsi, dans ce cas, les utilisateurs peuvent désormais traiter des recherches basées sur des mots-clés multidimensionnels. Supposons, par exemple, qu'un utilisateur U effectue une recherche par requête afin d'obtenir l'opinion $o1$ composée des mots-clés suivants $w1$, $w2$, $w3$ dans les conditions des dimensions $d1$, $d3$ et $d5$; l'utilisateur obtiendra les cuboïdes personnalisés correspondant aux dimensions $d1$, $d3$ et $d5$ qui contiennent les mots-clés $w1$, $w2$, $w3$ à la suite de la demande utilisateur.

2. **Filtrage à base de contenu :** cette phase est orientée filtrage. Elle est utilisée pour capturer les informations les plus utiles par rapport au besoin déclaré par l'utilisateur. Elle exploite la *requête expansée fonctionnelle* définie dans les sections précédentes (voir module requêtage) dans une technique de filtrage à base de contenu. Le filtrage est déclenché sur les cuboïdes personnalisés générés après l'étape de l'analyse multidimensionnelle personnalisée. La technique choisie utilise la mesure $TF-IDF$ (fréquence inversée en fréquence) comme premier pas pour filtrer les "Good Data" afin d'avoir des "Useful Data" qui répondent au besoin réel de l'utilisateur, et utiliser la mesure de similarité en cosinus dans un deuxième pas, pour sélectionner les meilleurs résultats parmi les données utiles trouvées.

En revanche, afin de démonter le fonctionnement de ce module, nous utilisons un exemple qui concerne l'analyse et le filtrage des recommandations des clients en ligne (ORC) disponibles sur le site Web de Trip Advisor. L'importance d'un mot dans les ROC est représentée par une valeur statique ou une mesure. Dans les sections précédentes, le cube OLAP défini par Apache Kylin utilise les valeurs agrégées de TF, IDF et $TF-IDF$ correspondant à la combinaison de dimensions en fonction de la demande de l'utilisateur.

Les OCR ont été décomposés de manière décisive sous la forme de vecteurs d'opinion. Ce dernier est divisé en mots et leur fréquence est représentée par la mesure TF qui permet de déterminer la valeur de priorité des mots de la requête de l'utilisateur. IDF représente l'inverse de DF (Document Frequency). La pondération $TF - IDF$ produit un poids composite pour chaque terme des ORC.

D'une autre part, la requête de l'utilisateur doit être traitée et représentée sous forme de vecteur. Il est bien entendu possible de calculer les similitudes en cosinus entre le vecteur de la requête et ceux des ORC correspondants dans les cuboïdes. Le classement des résultats est basé sur la similarité entre la requête de l'utilisateur et les ORC récupérés.

Considérons l'exemple suivant :

> The hotel staff was very friendly and helpful. The room was clean and has a great view. My wife and I had a room with a terrace overlooking the water it was a great view

Cette recommandation est traitée par la phase de prétraitement des données et de l'extraction des opinions présentées dans la section de validation et expérimentation comme suit :

$$o_1 = (hotel, staff, very, friendly, helpful);$$
$$o_2 = (room, clean, great, view);$$
$$o_3 = (room, terrace, overlooking, water, great, view);$$

Imaginons que nous fassions une recherche sur ces opinions avec la requête suivante : *"great room and clean"*, cette requête peut être représentée après une étape de prétraitement de la requête comme suit :

$$Q = (great, room, clean);$$

Nous examinons un exemple en détail pour voir comment l'analyse fonctionne dans un cube OLAP personnalisé.

- **Étape 1: calcule de "Term Frequency (TF)"**

Dans cette étape nous déterminons l'ensemble des termes et leur fréquence pour chacune des opinions définies précédemment (voir Table 4.5).

o_1	hotel	staff	very	friendly	helpful	
TF	1	1	1	1	1	
o_2	room	clean	great	view		
TF	1	1	1	1		
o_3	room	terrace	overlooking	water	great	view
TF	1	1	1	1	1	1

Table 4.5: Termes et fréquence des opinions

Par exemple, dans o_1, le terme hotel apparaît une fois. Le nombre total de termes dans opinion o_1 est 5. Par conséquent, la fréquence des termes normalisés (NTF) est égale à 1/5 = 0,2. La Table 4.6 contient les fréquences des termes normalisés pour toutes les opinions.

o_1	hotel	staff	very	friendly	helpful	
NTF	0.2	0.2	0.2	0.2	0.2	
o_2	room	clean	great	view		
NTF	0.25	0.25	0.25	0.25		
o_3	room	terrace	overlooking	water	great	view
NTF	0.166666	0.166666	0.166666	0.166666	0.166666	0.166666

Table 4.6: Normalized Term Frequency (NTF)

- **Étape 2: Calcule de "Inverse Document Frequency (IDF)"**

Pour chaque terme t_i dans une recommandation donnée, l'IDF peut être décrite comme suit :

$$IDF(t_i) = 1 + \log \frac{\text{Nombre total d'opinions}}{\text{Nombre d'opinions avec le terme} t_i}$$

Dans notre exemple, il y a 3 opinions en tout = o_1, o_2, o_3. Le terme *hotel* apparaît dans o_1, donc :

$$IDF(hotel) = 1 + log(3/1)$$

$$= 1 + 1.098726209$$

$$= 2.098726209$$

La Table 4.7 ci-dessous, présente l'IDF pour tous les termes utilisés dans toutes les opinions.

Termes	**IDF**
hotel	2.098726209
staff	2.098726209
very	2.098726209
friendly	2.098726209
helpful	2.098726209
room	1.405465108
clean	2.098726209
comfortable	2.098726209
terrace	2.098726209
overlooking	2.098726209
water	2.098726209
great	1.405465108
view	1.405465108

Table 4.7: Calcule de l'Inverse Document Frequency

- **Étape 3: Calcule de "TF*IDF"**

Dans cette étape, nous essayons de trouver l'opinion pertinente pour la requête : "*great room and clean*". Pour chaque terme de la requête, multiplier sa fréquence normalisée par son IDF sur chaque opinion. Dans l'opinion o_2 pour le terme *great*, la fréquence du terme normalisée est de 0,25 et son IDF est 1,405465108. En les multipliant ensemble, nous obtenons 0.351366277 (0.25 * 1.405465108).

La Table 4.8 ci-dessous, présente le TF*IDF de la requête "*great room and clean*" pour tous les termes utilisés dans toutes les opinions.

	TF	**IDF**	**TF*IDF**
great	0.333333	1.405465108	0,468487901
room	0.333333	1.405465108	0,468487901
clean	0.333333	2.098726209	0,699536730

Table 4.8: TF*IDF de la requête utilisateur dans toutes les opinions

- **Étape 4: Calcule de "Vector Space Model et Cosine Similarity"**

Dans cette étape, nous utilisons la formule donnée ci-dessous pour déterminer la similarité entre deux opinions quelconques.

Cosine Similarity (o_1, o_2) = Produit scalaire (o_1, o_2) / $||o_1||$ * $||o_2||$
Produit scalaire (o_1, o_2) = $o_1[0]$ * $o_2[0]$ + $o_1[1]$ * $o_2[1]$ * ... * o_1[n] * o_2[n]
$||o_1||$ = racine carrée ($o_1[0]^2$ + $o_1[1]^2$ + ... + $o_1[n]^2$)
$||o_2||$ = racine carrée ($o_2[0]^2$ + $o_2[1]^2$ + ... + $o_2[n]^2$)

La requête entrée par l'utilisateur peut également être représentée sous forme de vecteur. Les résultats de calcul du TF * IDF sont présentés dans la Table 4.9.

	TF	**IDF**	**TF*IDF**
great	0.333333	1.405465108	0,468487901
room	0.333333	1.405465108	0,468487901
clean	0.333333	2.098726209	0,699536730

Table 4.9: TF*IDF de la requête utilisateur

Calculons maintenant la similarité cosinus de la requête utilisateur et l'opinion o_2. Nous pouvons faire le calcul en utilisant ce formulaire :

Cosine Similarity(Query, o_2) = Produit scalaire(Query,o_2)/||Query|| * $||o_2||$
racine carrée (Query, o_2) = ((0.468487901)*(0.351366277)+(0.468487901)* (0.351366277)+
(0.699536730)*(0.524681552))
= 0.69623649
$||Query|| = sqrt((0.468487901)^2 + (0.468487901)^2 + (0.699536730)^2) = 0.96349121$
$||o_2|| = sqrt((0.351366277)^2 + (0.351366277)^2 + (0.524681552)^2) = 0.72261841$
Cosine Similarity (Query, o_2) = 0.69623649/(0.96349121)*(0.72261841)
= 0.69623649 / 0.69623649
= 1

Le résultat de la similarité cosinus pour toutes les opinions est présenté dans le Tableau 4.10, où, l'opinion o_2 a le score le plus élevé qui est égal à 1, il contient tous les mots clés de la requête.

	o_1	o_2	o_3
Cosine Similarity	0	1	0.68764779

Table 4.10: Cosine Similarity pour touts les opinions

4 Expérimentations et tests

Afin de montrer l'applicabilité de l'architecture proposée, nous avons mis en place des jeux de test pour valider le passage d'un module de notre approche vers un autre. Dans cette section, nous décrivons les expériences menées pour évaluer l'approche proposée pour une analyse personnalisée dans un contexte du Big Data, telles que, les recommandations des clients en ligne (Online Customer Reviews (OCRs)), en utilisant des techniques d'expansion de requêtes et de filtrage basées sur le contenu.

4.1 Montage expérimental

Afin de montrer l'applicabilité de l'architecture de notre système proposé, nous avons mis en place un environnement de stockage et de traitement basé sur la plate-forme Hadoop en intégrant Apache Spark, Apache Hive, Apache Sqoop, Apache Kylin et Hbase. Cet environnement de stockage et de traitement est installé et configuré sur une architecture matérielle dotée d'un processeur Intel (R) Core TM i5-4210 H à 2,90 GHz, de 8 Go de RAM, d'un disque dur de 1 To et d'un système Linux.

Pour évaluer la performance de notre approche, il est nécessaire d'assurer l'intercommunication entre les différents outils de l'environnement de test que nous avons choisi. Ce choix est motivé par la popularité de l'environnement dans la communauté Big Data, et par l'efficacité des outils de traitement et d'analyse des données.

4.2 Jeux de données (Dataset)

Le jeu de données OCR contient 359,130 recommandations en langue anglaise pour 143 hôtels. La collection d'OCR sur les hôtels du site Web Trip Advisor est réalisée à l'aide d'un robot d'indexation Web, la partie de l'ensemble de données brutes $R(r1, r2, ..., r|R|)$ est présentée dans le Tableau 4.11. De plus, la description statistique de l'ensemble de données est présentée dans la Table 4.12.

State	Time	Hotel Name	*ri*	Recommendations des clients
Karnataka	Q1	Hotel Giraffe	r1	Outstanding hotel, staff was amazing and the location and views make this a great base to visit an exhibition. Clean hotel with amazing staff.
Delhi	Q2	Casablanca	r2	The room was perfect. The breakfast in the morning was very nice.
Rajasthan	Q1	Casablanca	r3	Outstanding hotel for business travelers. Nice and clean. Rooms are spacious with air conditioning. Good selection of breakfast. Within easy reach of thc airport and to thc cxhibition.
Karnataka	Q1	Hotel Giraffe	r4	This is a grcat littlc hotcl. Wc had a wonderful view from the room and patio. Good size. Breakfast was great. The only issue was the two twin beds pushed together. That was very uncomfortable, great view. Good for travelers.
Maharashtra	Q2	Wellington Hotel	r5	It was a very busy hotel. Delightful experience
Rajasthan	Q1	Casablanca	r6	Staff was so helpful, nice hotel. Amazing property. First-class experience. Good for travelers.
Himachal Pradesh	Q2	Holiday Inn	r7	Lovely hotel. Staff was so helpful. The room was perfect.
Karnataka	Q1	Hotel Giraffc	r8	Lovely hotel. Staff was so helpful. The room was pcrfcct.
Maharashtra	Q1	Casablanca	r9	We loved our stay at the Casablanca. The room was perfect. The breakfast in the morning was very nice.

Rajasthan	Q1	Casablanca	r10	Great Hotel. Very friendly crew. The Chinese restaurant is really awesome and highly recommendable. It has a spa, pool and gym in the basement. It is located closely to the airport and to the exhibition center- which makes it great for business travelers.

Table 4.11: Une partie de l'ensemble de données brutes

ri: représente la recommandation numéro i ou $i \in \{1.n\}$.

Données brutes	
Nombre de clients	215,243
Nombre d'hôtels	143
Nombre de recommandation	359,130
Nombre de phrases	1,725,298

Table 4.12: Description statistique de l'ensemble de données brutes

4.3 Prétraitement des données et extraction des opinions

Pour démontrer la faisabilité et les performances de notre approche proposée, nous nous référons à un cas réel, tel que mentionné ci-dessus, qui repose sur les recommandations des clients sur le site Web de Trip Advisor. Le principal objectif de la tâche de prétraitement des données et d'extraction des opinions est de convertir le texte OCR brut $R(r1, r2, ..., r|R|)$ en données structurées. Il permet de gérer les entrées répétitives, les valeurs manquantes et la suppression des mots inutiles, de segmenter les recommandations originales en mots et en phrases [Decker and Trusov, 2010], et de supprimer tous les mots vides [Silva and Ribeiro, 2003]. Les processus impliqués dans le prétraitement et l'extraction des opinions sont présentés dans la Figure 4.7.

Figure 4.7: Processus de prétraitement de données et extraction d'opinions

(a) **Tokenization :** chaque texte OCR brut est divisé en termes significatifs appelés jetons. Exemple - "The hotel staff was very friendly and helpful" est converti en "The, hotel, staff, was, very, friendly, and, helpful".

(b) **Stop word removal :** tous les termes dans le texte brut des OCR qui n'ont pas de signification particulière sont supprimés comme a, the, then, etc. Ensuite, après avoir supprimé les *Stop word*, l'exemple précédent devient (hotel, staff, very, friendly, helpful).

(c) **Opinions extraction :** Plus précisément, si nous prenons l'exemple de commentaire suivant : "The hotel staff was very friendly and helpful. The room was clean and comfortable. My wife and I had a room with a terrace overlooking the water it was a great view!" peut être analysé et extrait comme suit :

- o_1 = (hotel, staff, very, friendly, helpful);
- o_2 = (room, clean, comfortable);
- o_3 = (room, terrace, overlooking, water, great, view).

L'algorithme de prétraitement et d'extraction des opinions est illustré dans le Tableau 4.13.

Algorithme : Prétraitement des données et extraction des opinions
Entrée : collection d'OCR $R(r1, r2, ..., r\|R\|)$, liste Stop words $w1, w2, ..., wn$; **Sortie :** vecteur d'opinions $S = o1, o2, ..., on$;
Etape 1 : Scindez les OCR $R(r1, r2, ..., r\|R\|)$ en vecteur de jetons (Termes) $T = t1, t2, ..., tn$ avec Apache Spark; **Etape 2 :** Supprimer les mots vides en utilisons la liste des *Stop words* de Apache Spark; **Etape 3 :** Analyser et extraire les vecteurs d'opinion $S = o_1, o_2, ..., o_n$ avec Apache Spark;

Table 4.13: Algorithme de prétraitement des données et extraction des opinions

4.4 Modèle de cube de texte pour OLAP

Dans notre étude de cas, le cube OLAP résultant analyse les mesures afin de renvoyer les opinions bien adaptées à la demande de l'utilisateur. La figure 4.8 montre un exemple de modèle de cube de texte basé sur trois hiérarchies de dimensions, telles que le nom de l'hôtel, la date / heure et l'adresse. Il a stocké les opinions de différents hôtels sous forme de données texte, conformément au schéma en étoile présenté à la figure 4.9.

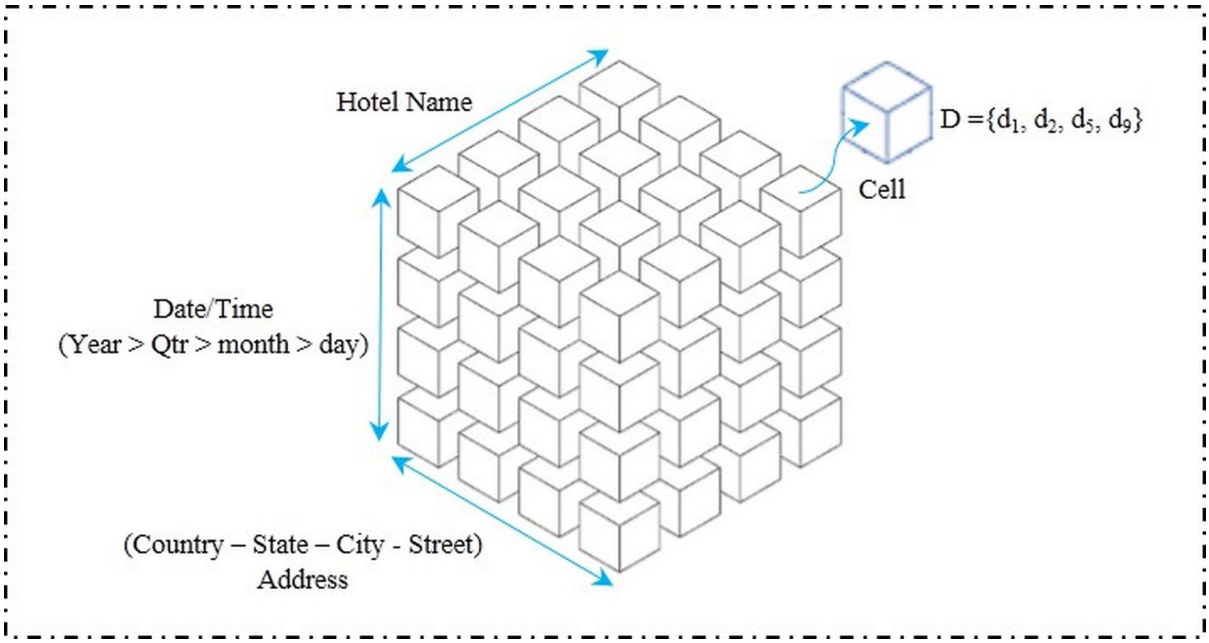


Figure 4.8: Modèle de cube de texte

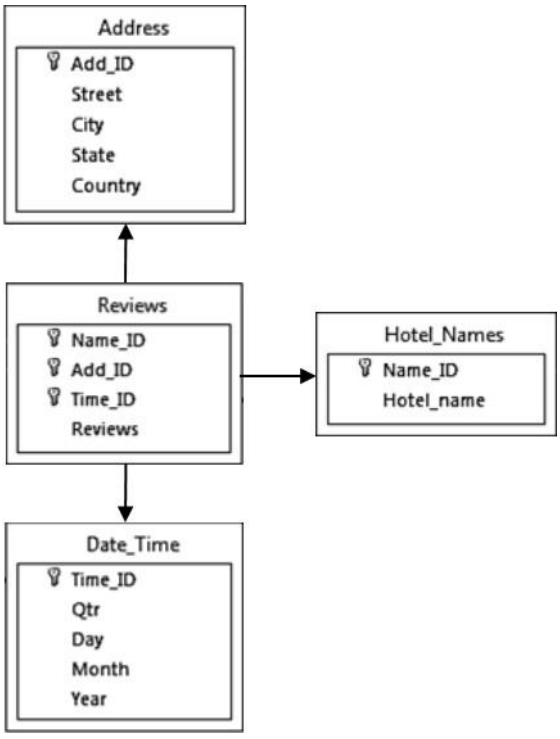


Figure 4.9: Schéma en étoile

4.5 Scénario de test

Afin de tester la validité de notre scénario présenté dans la figure 4.10, nous utilisons une partie de l'ensemble de données brutes présentées précédemment dans le tableau 4.11.

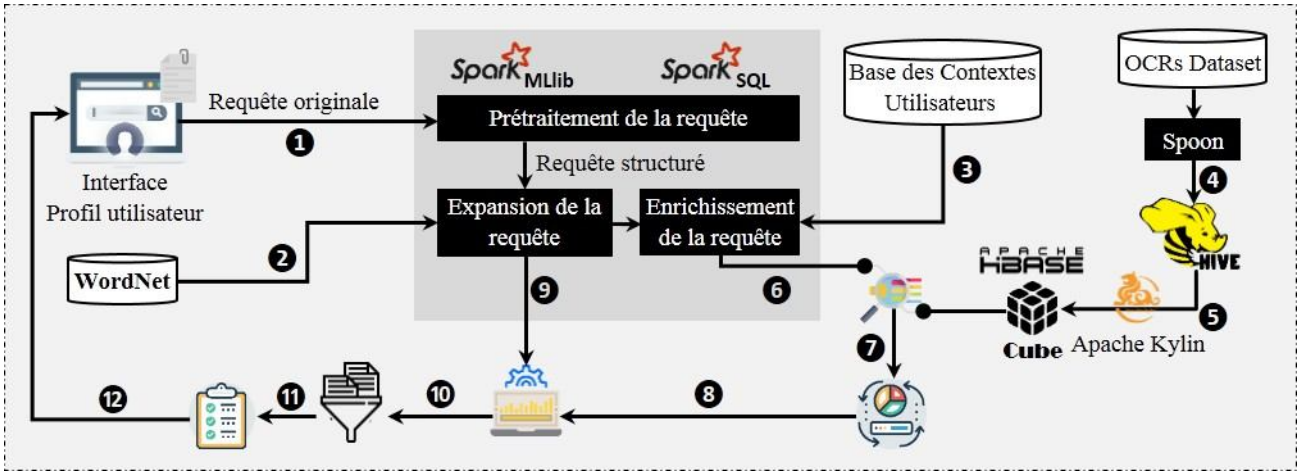


Figure 4.10: Scénario de validation

1. Après l'identification de l'utilisateur avec son profil, une requête initiale est faite par l'utilisateur, exprimant son besoin initial de recherche. La requête d'origine est envoyée à l'étape de prétraitement pour définir une requête structurée.

   Supposons, par exemple, qu'un utilisateur souhaite consulter les avis des clients les mieux adaptés, contenant la phrase "Great for business travelers".

   Le prétraitement de la requête d'origine selon le processus défini à la figure 4.7, à l'aide des classes `Tokenizer` et `StopWordsRemover` d'Apache Spark donnera la requête structurée suivante : $SQ : (Great|Business|Travelers)$.

2. L'expansion de la requête structurée conformément au processus défini dans la figure 4.6 selon le code définie dans la Table 4.14 donnera les requêtes expansées fonctionnelles suivantes :

   - $FEQ_0 : (Great|Business|Travelers)$
   - $FEQ_1 : (Outstanding|Business|Travelers)$

3. L'enrichissement des requêtes expansées fonctionnelles (FEQs) selon le processus défini à la figure 4.6 nécessitera l'inclusion d'autres contraintes multidimensionnelles stockées dans la partie dynamique du profil de l'utilisateur. Une requête enrichie non fonctionnelle (NFEQ) est représentée comme suit :

.....

from nltk.corpus import wordnet
synonyms = []
for syn in wordnet.synsets('Great'):
  for lemma in syn.lemmas():
  synonyms.append(lemma.name())
print(synonyms)

La sortie est: Outstanding

Table 4.14: Code source pour la recherche de synonyme dans WordNet

- State = 'Karnataka' or 'Rajasthan' and Time = q1 (Quarter-1);
- Hotel names= 'Hotel Giraffe' or 'Casablanca';

4. Comme indiqué dans la section Prétraitement des données et extraction d'opinions, nous définissons un vecteur structuré $S = o_1, o_2, ..., o_n$ qui élimine les mots non pertinents et stocke l'ensemble des données OCR prétraitées sous forme d'opinions à l'aide d'Apache Spark.

   Ce processus de prétraitement repose sur l'utilisation de deux classes d'Apache Spark. La première est la classe *Tokenizer* qui prend le texte (comme une phrase) et le divise en termes individuels (généralement des mots) ; la seconde est *StopWordsRemover* , prend en entrée une séquence de chaînes (par exemple, la sortie d'un Tokenizer) et supprime tous les mots d'arrêt des séquences d'entrée. La liste des mots vides est spécifiée par le paramètre *StopWords*.

   Une partie du résultat prétraité de l'ensemble de données brutes présentées dans la table 4.11est présentées dans le tableau 4.15 suivant.

O_n	S1
o_1	Outstanding|Hotel
o_2	Staff|Amazing|Location|Views|Great|Base|visit|Exhibition
o_3	Clean|Hotel|Amazing|Staff

O_n	S2
o_1	Outstanding|Hotel|Business|Travelers
o_2	Nice|Clean
o_3	Rooms|Spacious|Air conditioning
o_4	Good|Selection|Breakfast
o_5	Easy|Reach|Airport|Exhibition
O_n	S3
o_1	Staff|So|Helpful
o_2	Nice|Hotel|Amazing|Property
o_3	First-class|Experience
o_4	Good|Travelers
O_n	S4
o_1	Lovely|Hotel
o_2	Staff|Friendly|Helpful|Breakfast|Great
o_3	Great|Location|Not|Leave
o_4	Great|Hotel|Jacuzzi|Bath
O_n	S5
o_1	Great|Hotel
o_2	Very|Friendly|Crew
o_3	Chinese|Restaurant|Really|Awesome|Highly|Recommendable
o_4	Pool|Gym|Basement
o_5	Located|Closely|Airport|Exhibition|center
o_6	Great|Business|Travelers

Table 4.15: Résultat de prétraitement

5. Créer le cube OLAP à partir du magasin de données analytiques à l'aide du module OLAPing présenté dans la figure 4.3.

6. Un travail de filtre multidimensionnel est démarré ; la requête enrichie non fonctionnelle NFEQ est utilisée dans ce cas pour réduire l'espace de recherche dans le cube OLAP et extraire les "Good Data" associées au profil de l'utilisateur.

7. Les "Good Data" extraites depuis le cube OLAP sont présentées dans la Table 4.16.

O_n	S1
o_1	Outstanding|Hotel
o_2	Staff|Amazing|Location|Views|Great|Base|visit|Exhibition
o_3	Clean|Hotel|Amazing|Staff
O_n	S2
o_4	Outstanding|Hotel|Business|Travelers
o_5	Nice|Clean
o_6	Rooms|Spacious|Air conditioning
o_7	Good|Selection|Breakfast
o_8	Easy|Reach|Airport|Exhibition
O_n	S3
o_9	Great|Little|Hotel
o_{10}	Wonderful|View|Room|Patio|Good|Size
o_{11}	Breakfast|Great
o_{12}	Only|Issue|Twin beds|Pushed|Together|Very|Uncomfortable
o_{13}	Great|View|Good|Travelers
O_n	S4
o_{14}	Staff|So|Helpful
o_{15}	Nice|Hotel|Amazing|Property
o_{16}	First-class|Experience
o_{17}	Good|Travelers

O_n	S5
o_{18}	Lovely|Hotel
o_{19}	Staff|Friendly|Helpful|Breakfast|Great
o_{20}	Great|Location|Not|Leave
o_{21}	Great|Hotel|Jacuzzi|Bath
O_n	S6
o_{22}	Grcat|Hotcl
o_{23}	Very|Friendly|Crew
o_{24}	Chinese|Restaurant|Really|Awesome|Highly|Recommendable
o_{25}	Pool|Gym|Basement
o_{26}	Located|Closely|Airport|Exhibition|center
o_{27}	Great|Business|Travelers

Table 4.16: Résultat de prétraitement

8. Calcul de la mesure IF pour tous les termes apparaissant dans tous les résultats de "Good Data".

9. Chargement des requêtes expansées (FEQ_0 et FEQ_1) FEQ_0 pour calculer les mesures $TF * IDF$ (Voir figure 4.11).

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}	o_{11}	o_{12}	o_{13}	o_{14}	o_{15}	o_{16}	o_{17}	o_{18}	o_{19}	o_{20}	o_{21}	o_{22}	o_{23}	o_{24}	o_{25}	o_{26}	o_{27}
Great	0,00	0,29	0,00	0,00	0,00	0,00	0,00	0,00	0,69	0,00	1,05	0,00	0,52	0,00	0,00	0,00	0,00	0,00	0,42	0,52	0,52	1,05	0,00	0,00	0,00	0,00	0,69
Business	0,00	0,00	0,00	0,90	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,19
Travelers	0,00	0,00	0,00	0,73	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,73	0,00	0,00	0,00	1,45	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,96
Outstanding	0,00	0,00	0,00	0,90	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Business	0,00	0,00	0,00	0,90	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,19
Travelers	0,00	0,00	0,00	0,73	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,73	0,00	0,00	0,00	1,45	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,96

Figure 4.11: Calcule de $TF * IDF$ pour FEQ_0 et FEQ_1

10. Découvrir toutes les opinions pertinentes pour les requêtes : FEQ_0 et FEQ_1, en utilisant les mesures $TF * IDF$ présentées dans l'étape (9).

11. Filtrage sur les "Good Data" précédemment obtenues, en éliminant les opinions portant $TF * IDF = 0$. Un nouveau jeu de données appelé "Useful Data", présenté dans la Table 4.17, s'affichera et s'intégrera comme il convient dans les requêtes d'expansion fonctionnelles.

O_n	S_n
o_2	Staff|Amazing|Location|Views|Great|Base|visit|Exhibition
o_4	Outstanding|Hotel|Business|Travelers
o_9	Great|Little|Hotel
o_{11}	Breakfast|Great
o_{13}	Great|View|Good|Travelers
o_{17}	Good|Travelers
o_{19}	Staff|Friendly|Helpful|Breakfast|Great
o_{20}	Great|Location|Not|Leave
o_{21}	Great|Hotel|Jacuzzi|Bath
o_{22}	Great|Hotel
o_{27}	Great|Business|Travelers

Table 4.17: Useful Data

12. Analyser les résultats obtenus en calculant la similarité du cosinus entre FEQ_0, FEQ_1 et toutes les opinions qui représentent les "Useful Data" précédemment obtenues. Les résultats de calcul de la similarité du cosinus sont présentés dans la figure 4.12.

    La figure 4.12 montre que les opinions o_4 et o_{27} ont le score le plus élevé, car elles contiennent tous les mots-clés des requêtes.

	O_2	**O_4**	**O_9**	**O_{11}**	**O_{13}**	**O_{17}**	**O_{19}**	**O_{20}**	**O_{21}**	**O_{22}**	**O_{27}**
FEQ_0	0,20	1,79	0,48	0,73	1,07	1,41	0,29	0,37	0,37	0,73	2,84
$\|FEQ_0\|$	1,69	1,69	1,69	1,69	1,69	1,69	1,69	1,69	1,69	1,69	1,69
$\|o_i\|$	0,29	1,16	0,69	1,05	0,90	1,45	0,42	0,52	0,52	1,05	1,68
Cosine (FEQ_0, o_i)	0,41	0,91	0,41	0,41	0,71	0,57	0,41	0,41	0,41	0,41	**1,00**

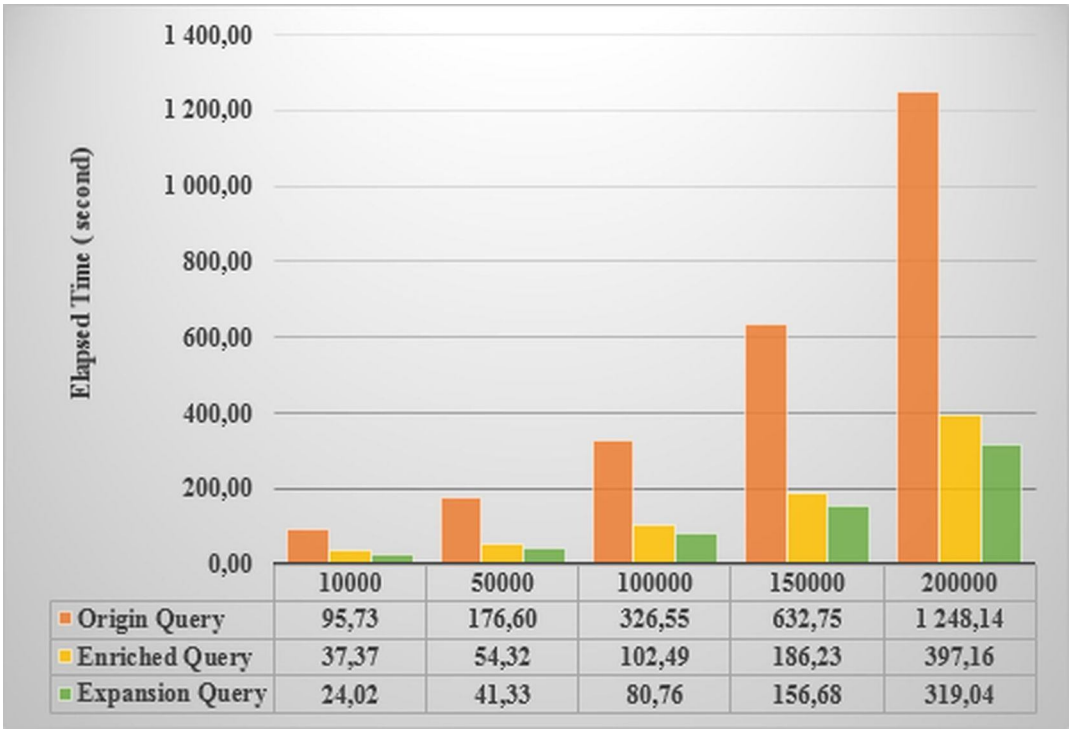
	O_4	**O_{13}**	**O_{17}**	**O_{27}**
FEQ_1	2,87	0,71	1,41	2,36
$\|FEQ_1\|$	1,96	1,96	1,96	1,96
$\|o_i\|$	1,47	0,73	1,45	1,53
Cosine (FEQ_1, o_i)	**1,00**	0,50	0,50	0,79

Figure 4.12: Similarité en cosinus pour FEQ_0, FEQ_1 et les opinions de Useful Data

4.6 Résultats expérimentaux et discussion

L'objectif de notre contribution est d'évaluer et comprendre l'impact de l'utilisation du profil utilisateur et les techniques d'expansion des requêtes dans la personnalisation de l'analyse dans un contexte Big Data, en utilisent un cube OLAP. Pour ce faire, nous avons effectué deux expériences : (1) le temps d'exécution des requêtes en fonction du nombre de n-uplets stockés dans le cube OLAP ; et (2) le nombre d'OCR extraits (opinions) à partir des n-uplets stockés dans le cube OLAP.

- **Temps d'exécution :** le premier test compare les temps d'exécution des différentes requêtes (originale, expansée et enrichie) en faisant varier le nombre de n-uplets stockés dans le cube OLAP. Dans chaque cas, un gain de temps significatif est obtenu. Plus précisément, les requêtes sur le cube personnalisé (contient les données personnalisation selon le profil de l'utilisateur) sont exécutées avec un temps presque quatre fois plus rapide que sur le cube initial (non personnalisé) (voir la figure 4.13).



	10000	50000	100000	150000	200000
Origin Query	95,73	176,60	326,55	632,75	1 248,14
Enriched Query	37,37	54,32	102,49	186,23	397,16
Expansion Query	24,02	41,33	80,76	156,68	319,04

Figure 4.13: Temps écoulé en faisant varier le nombre de n-uplets

La figure 4.14 illustre le pourcentage du gain de temps obtenu en exécutant des requêtes sur le cube initial (non personnalisé) et sur le cube personnalisé. En moyenne, le gain atteint est de 75% au maximum, quelque soit le nombre de n-uplets manipulés. Les courbes de tendance montrent que plus le nombre des tuples est élevé, le

gain de temps aura moins d'importance.

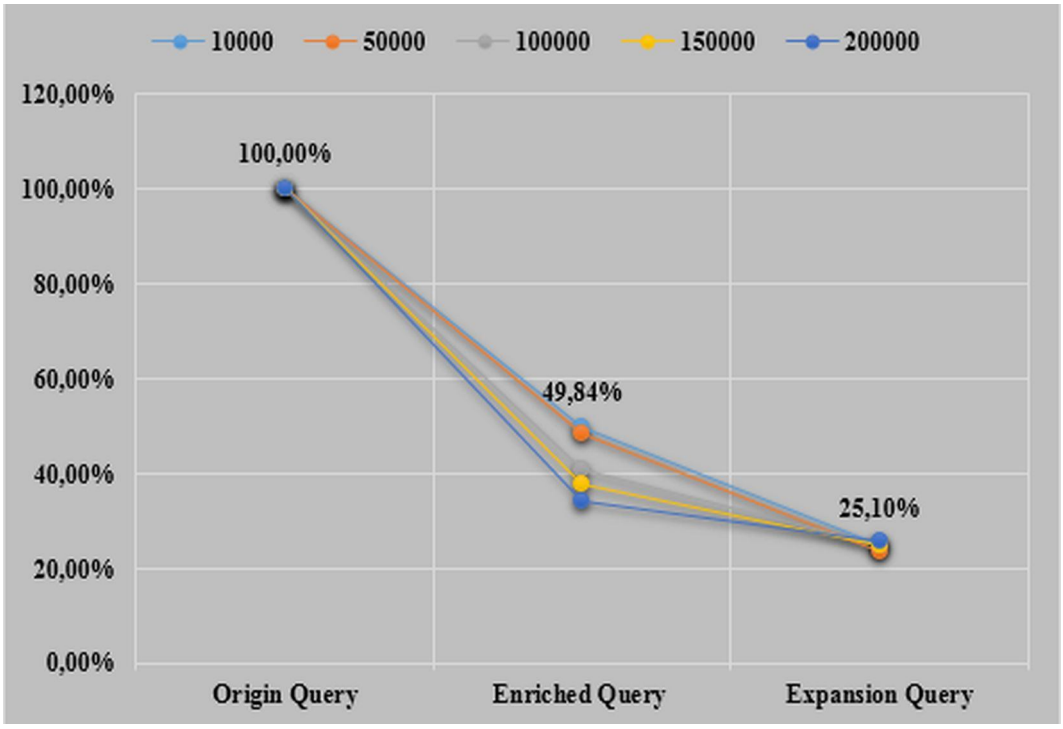
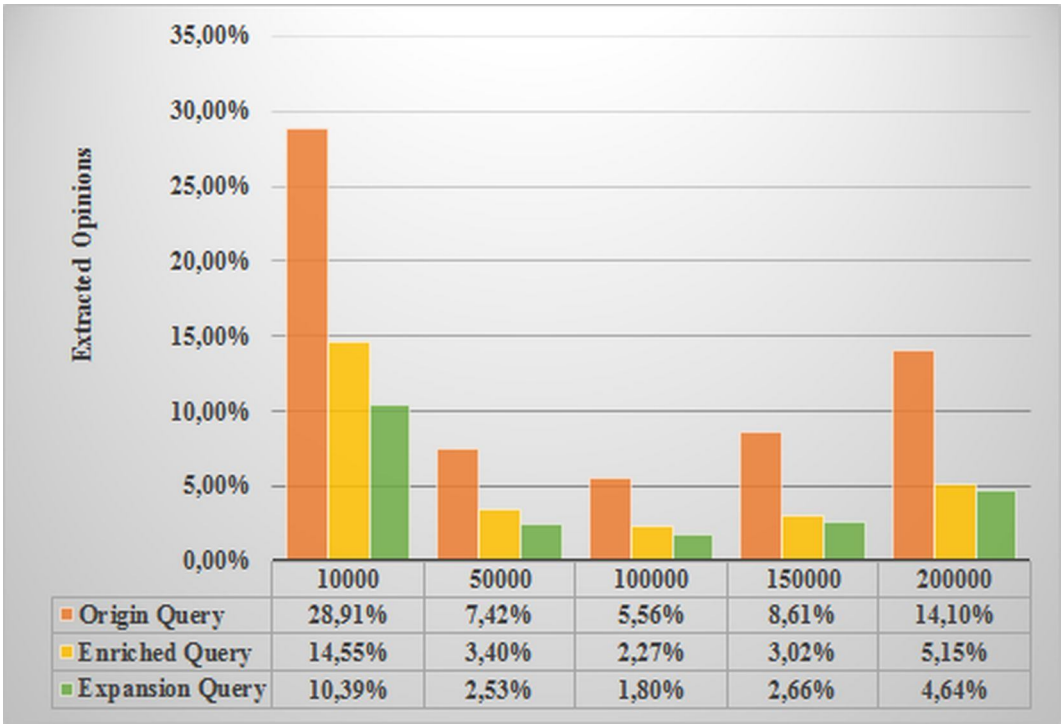


Figure 4.14: Gain en temps d'exécution des requêtes

- **Opinions extraites :** le deuxième test compare le nombre d'opinions extraites après l'exécution des différentes requêtes (d'origine, enrichie et expansée) en faisant varier le nombre de n-uplets stockés dans le cube OLAP. Dans chaque cas, le nombre d'OCR est considérablement réduit (voir figure 4.15).



	10000	50000	100000	150000	200000
Origin Query	28,91%	7,42%	5,56%	8,61%	14,10%
Enriched Query	14,55%	3,40%	2,27%	3,02%	5,15%
Expansion Query	10,39%	2,53%	1,80%	2,66%	4,64%

Figure 4.15: Opinions extraits du cube OLAP

5 Conclusion

Cette contribution présente une approche d'analyse multidimensionnelle améliorée, permettant de personnaliser l'analyse dans un contexte de Big Data, à l'aide des techniques d'expansion de requête et du filtrage basé sur le contenu. La principale contribution théorique de cette approche consiste à intégrer le profil de l'utilisateur et le contexte de sa requête de recherche dans la personnalisation du cube OLAP, afin d'obtenir les données les plus proches à son contexte. Ensuite, sur la base de cette personnalisation, nous calculons les fréquences et la similarité des opinions des clients précédemment obtenues pour les requêtes fonctionnelles de l'utilisateur. Un cube de données réduit contenant des données utiles "Useful Data" pour l'utilisateur est présenté. Notre expérience montre que l'approche proposée a donné des résultats très importants, elle est composée de quatre modules principaux : (a) module de profilage, (b) module de requêtage, (c) module OLAPing, et (d) module d'analyse et de filtrage. Pour valider la performance de l'approche proposée, le travail présenté a utilisé des données OCR du site Web de Trip Advisor, ainsi que le coefficient $TF * IDF$ et la similarité de cosinus pour mesurer la performance. Notre approche nécessite beaucoup de temps de calcul lors du prétraitement des données et de l'extraction des opinions.

En tant que perspective, nous visons à compléter les points suivants : Améliorer notre approche en utilisant des techniques sémantiques pour le traitement des opinions, et valider l'approche sur des flux de données en temps réel dans une architecture cloud.

Conclusion et perspectives

1 Conclusion

Les travaux de recherche présentés dans ce mémoire, s'orientent vers un rapprochement technologique qui consiste à intégrer les techniques de la personnalisation et le filtrage des informations dans l'analyse de Big Data, afin d'offrir une solution de Business Intelligence plus performante par un processus d'analyse en ligne OLAP personnalisé. Ce dernier est basé sur le concept de profil utilisateur et le contexte de la requête de recherche utilisateur comme des éléments essentiels pour la personnalisation de l'information dans le but d'accéder rapidement aux données les plus pertinentes, en tenant compte des besoins des utilisateurs, et en facilitant l'analyse interactive et la synthèse des données personnalisées.

En revanche, le profil utilisateur dans le contexte des systèmes de personnalisation d'informations, peut être défini comme une structure qui permet de modéliser et stocker des informations relatives au contexte de l'utilisateur, il est généralement construit à partir de l'historique des activités de l'utilisateur, et reste un critère indispensable, mais il n'est pas assez suffisant pour la personnalisation des informations, car il est souvent lié à d'autres critères tels que; les préférences et le contexte de recherche. En ce qui concerne les préférences, elles sont liées fortement à son profil et on ne peut en aucun cas séparer les unes des autres, mais leur description peut changer en fonction du contexte. Par ailleurs, le contexte d'une préférence dans nos travaux de recherche définit la portée des préférences, c'est-à-dire l'environnement dans lequel la préférence doit être prise en compte pour une éventuelle session d'analyse.

Les travaux présentés dans cette thèse traitent deux problématiques principales dans lesquelles nous avons proposé deux contributions :

1. Contribution pour l'exploitation du profil utilisateur dans la reformulation de requêtes.
2. Contribution pour la personnalisation de l'analyse dans le Big Data par l'utilisation des requêtes reformulées et le filtrage basé sur le contenu.

L'objectif de notre première contribution consiste à proposer une nouvelle approche, qui permet d'exploiter le profil de l'utilisateur et le contexte de la recherche utilisateur

dans la reformulation des requêtes utilisées pour réduire et personnaliser un espace de données volumineuses structurées (Big Data) afin d'effectuer des opérations analytiques multidimensionnelles rentables sur des cubes OLAP personnalisés. Cette approche offre deux optimisations majeures qui touchent le temps d'exécution des requêtes et la taille des cubes OLAP générés.

La deuxième contribution présente une approche d'analyse multidimensionnelle améliorée, permettant de personnaliser l'analyse dans un contexte de Big Data, à l'aide des techniques d'expansion de requête et de filtrage basées sur le contenu. La principale contribution théorique de cette approche consiste à introduire deux nouveaux formes de requêtes (fonctionnelles et d'autres non fonctionnelles) utilisées dans les différents modules de l'approche. Cette approche offre aussi des optimisations en matière de temps d'exécution des requêtes sur le cube personnalisé et le nombre d'opinions extraits après l'exécution des différentes requêtes (fonctionnelles et non fonctionnelles) en faisant varier le nombre de n-uplets stockés dans le cube OLAP généré.

Enfin, en ce qui concerne les deux questions posées dans l'introduction générale :

- **Comment peut-on intégrer le profil de l'utilisateur et le contexte de recherche pour la reformulation des requêtes utilisateur ?**

Les résultats des expérimentations nous ont permis de montrer l'efficacité et les optimisations majeures dans la personnalisation de l'analyse OLAP, par l'intégration du contexte de recherche et le profil de l'utilisateur dans la reformulation des requêtes. Un gain important dans le temps d'exécution des requêtes reformulées, et la taille des cubes OLAP personnalisés.

- **Comment peut-on personnaliser l'analyse dans le Big Data par l'utilisation des requêtes reformulées et le filtrage basé sur le contenu ?**

A travers les expérimentations effectuées sur les données collectées depuis TripAdvisor, nous avons pu constater que ; les résultats obtenus par l'approche de la personnalisation de l'analyse à base de profil utilisateur restent ambiguës devant ce type de données volumineuses. Ceci, nous a orienté vers l'intégration d'autres techniques afin d'assurer une personnalisation plus précise.

L'idée est de rajouter une autre source externe dans la reformulation de requêtes à côté de la base contextuelle des utilisateurs, pour distinguer deux types de requêtes expansées

(fonctionnelle et non fonctionnelle). Ces deux dernières ont été utilisées dans le but de séparer l'étape de l'analyse multidimensionnelle personnalisée de l'étape de filtrage.

L'étape de l'analyse multidimensionnelle nous conduit à la création des cubes personnalisés en fonction du profil de l'utilisateur et ses préférences de recherche. Les résultats obtenus rassemblent des données dites "Good Data". En revanche, l'utilisation d'une technique de filtrage basée sur le contenu est indispensable, elle utilise la mesure TF-IDF comme premier pas pour filtrer les "Good Data" afin d'avoir des "Useful Data" qui répondent au besoin réel de l'utilisateur, et utilise aussi la mesure de similarité en cosinus dans un deuxième pas pour sélectionner les meilleurs résultats parmi les données les plus utiles trouvées.

En conclusion, nous avons montré l'apport de nous contributions dans la reformulation de requêtes et la personnalisation d'analyse dans le Big Data. Les approches et les techniques proposées peuvent être améliorées encore et encore, mais nous espérons que cette préfiguration puisse ouvrir des pistes de recherche pour aller plus loin dans de futurs travaux.

2 Perspectives

Les deux problématiques abordées dans cette thèse sont relativement nouvelles dans le contexte des systèmes de personnalisation dans un contexte de Big Data, à cet effet plusieurs perspectives sont envisageables.

1. **Les perspectives envisageables à court terme sont les suivantes :**

   - **Validation de l'approche sur un flux de données en temps réel :** notre deuxième approche doit encore être testée sur de grandes sources de flux de données en temps réel, issues de projets réels. Ces éventuels tests nécessitent des changements dans l'architecture de Big Data proposée et le module OLAPing proposé.
   - **Déploiement dans une architecture cloud :** nous envisageons de déployer le module de l'analyse OLAP défini précédemment dans notre deuxième contribution comme un service dans le Cloud, afin d'aboutir au "OLAP as a service in the Cloud". Ce dernier prend efficacement en charge l'"analyse multidimensionnelle personnalisée des données volumineuses. Plus précisément, l'architecture proposée dans la deuxième contribution sera adoptée pour un environnement Cloud afin de résoudre les problèmes de prolifération des données et de l'obsolescence de la technologie.

2. **Les perspectives envisageables à long terme sont les suivantes :**

   - **Intégration de l'aspect sémantique :** nous proposons également d'enrichir le module de requêtage par des techniques sémantiques, afin d'élargir la combinaison des synonymes des termes avec les termes de la requête de l'utilisateur, et ce, pour avoir des nouvelles requêtes sémantiquement plus proches au contexte de l'utilisateur.
   - **Personnalisation à base de profil sociale :** dans cette perspective, nous envisageons d'intégrer la notion de profil sociale (construit à partir des informations contenues dans son réseau social) à côté de profile classique pour personnaliser des données volumineuses issues d'un type de réseau social, tel que, les réseaux de chercheurs scientifiques qui sont souvent étudiés et exploités en tant qu'échantillon de test par le monde académique. L'intérêt de l'utilisation du profil utilisateur et du profil social est de raffiner les résultats des requêtes de l'utilisateur issus d'une étape de recherche.

Publications et Communications

Publications internationales

1. Menaceur, S., Derdour, M., and Bouramoul, A. (2017a). Personalized online analytical processing in big data context using user profile and search context. *International Journal of Strategic Information Technology and Applications (IJSITA)*, 8(4):67-80.

2. Menaceur, S., Derdour, M., and Bouramoul, A. (2019). Using query expansion techniques and content-based filtering for personalizing analysis in big data. *International Journal of Information Technology and Web Engineering (IJITWE).*, Sous Presse.

Communications internationales

1. Menaceur, S., Derdour, M., and Bouramoul, A. (2016). Olaping and big data mining: A survey. In *The 2nd International Conference on Pattern Analysis and Intelligent Systems*, PAIS '2016, Khenchela, Algeria.

2. Menaceur, S., Derdour, M., and Bouramoul, A. (2017b). Vers une approche pour la prise en compte de l'usager dans l'analyse olap. In *La 11ème édition de la conférence maghrébine sur les Avancées des Systèmes Décisionnels*, ASD '2017, Tabarka, Tunisie.

Références bibliographiques

Abdelhédi, F. (2014). *Conception assistée d'entrepôts de données et de documents XML pour l'analyse OLAP*. PhD thesis, Université de Toulouse.

Acharjya, D. P. and Ahmed, K. (2016). A survey on big data analytics: challenges, open research issues and tools. *International Journal of Advanced Computer Science and Applications*, 7(2):511–518.

Acilar, A. M. and Arslan, A. (2009). A collaborative filtering method based on artificial immune network. *Expert Syst. Appl.*, 36:8324–8332.

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:734–749.

Aggarwal, C. C. (2016). An introduction to recommender systems. In *Recommender systems*, pages 1–28. Springer.

Al-Aqrabi, H., Liu, L., Hill, R., and Antonopoulos, N. (2012). Taking the business intelligence to the clouds. In *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, pages 953–958. IEEE.

Apache (2015). Extreme olap engine for big data. `http://kylin.apache.org`.

Aubay (2015). Le big data. `https://www.aubay.com/wp-content/uploads/2015/03/Regard-Aubay-Big-Data-Web.pdf`.

Audeh, B. (2014). *Sémantique query reformulation for ad hoc information retrieval on the Web*. Phd theses, Ecole Nationale Supérieure des Mines de Saint-Etienne.

Azad, H. K. and Deepak, A. (2019). Query expansion techniques for information retrieval: a survey. *Information Processing Management*, 56(5):1698 – 1735.

Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval*, volume 463. ACM press New York.

Banerjee, A., Bandyopadhyay, T., and Acharya, P. (2013). Data analytics: Hyped up aspirations or true potential ?. *Vikalpa*, 38(4):1–12.

Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., and Laurent, D. (2005). A personalization framework for olap queries. In *Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP*, DOLAP '05, pages 9–18, New York, NY, USA. ACM.

Benjelloun, F., Lahcen, A. A., and Belfkih, S. (2015). An overview of big data opportunities, applications and tools. In *2015 Intelligent Systems and Computer Vision (ISCV)*, pages 1–6.

Bentayeb, F., Boussaid, O., Favre, C., Ravat, F., and Teste, O. (2009). Personnalisation dans les entrepôts de données: bilan et perspectives. In *EDA*, pages 7–22.

Bentayeb, F., Favre, C., and Boussaid, O. (2008). A user-driven data warehouse evolution approach for concurrent personalized analysis needs. *Integrated Computer-Aided Engineering*, 15(1):21–36.

Bhogal, J., MacFarlane, A., and Smith, P. (2007). A review of ontology based query expansion. *Information Processing Management*, 43:866–886.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, NY. Softcover published in 2016.

Blagov, A., Rytcarev, I., Strelkov, K., and Khotilin, M. (2015). Big data instruments for social media analysis. In *Proceedings of the 5th International Workshop on Computer Science and Engineering*, pages 179–184.

Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46:109–132.

Bouramoul, A. (2011). *Recherche d'Information Contextuelle et Sémantique sur le web*. PhD thesis, Université MENTOURI de Constantine. Algérie.

Bradley, K., Rafter, R., and Smyth, B. (2000). Case-based user profiling for content personalisation. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 62–72. Springer.

Brown, P. J., Bovey, J. D., and Chen, X. (1997). Context-aware applications: from the laboratory to the marketplace. *IEEE personal communications*, 4(5):58–64.

Bruande, M.-F. and Chevallet, J.-P. (2003). Assistance intelligente à la recherche d'information. *Edition Hermes, Auteurs: Gaussier, E., Stefanini, MH Chapitre*, 3:85–118.

Brusilovsky, P. (1998). Methods and techniques of adaptive hypermedia. In *Adaptive hypertext and hypermedia*, pages 1–43. Springer.

Burke, R. D. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12:331–370.

Carpineto, C. and Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1:1–1:50.

Chatzopoulou, G., Eirinaki, M., and Polyzotis, N. (2009). Query recommendations for interactive database exploration. In Winslett, M., editor, *Scientific and Statistical Database Management*, pages 3–18, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chaudhuri, S., Dayal, U., and Narasayya, V. (2011). An overview of business intelligence technology. *Communications of the ACM*, 54(8):88–98.

Chen, C. P. and Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information sciences*, 275:314–347.

Chen, H., Chiang, R. H., and Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS quarterly*, 36(4).

Chen, L.-S., Hsu, F.-H., Chen, M.-C., and Hsu, Y.-C. (2008). Developing recommender systems with the consideration of product profitability for sellers. *Information Sciences*, 178(4):1032–1048.

Chen, W., Wang, H., Zhang, X., and Lin, Q. (2017). An optimized distributed olap system for big data. In *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*, pages 36–40. IEEE.

Codd, E. F., Codd, S. B., and Salley, C. T. (1993). Providing olap (on-line analytical processing) to user-analysts: An it mandate. *Codd and Date*, 32:31.

Cuzzocrea, A., Bellatreche, L., Song, I.-Y., et al. (2013). Data warehousing and olap over big data: current challenges and future research directions. In *DOLAP*, volume 13, pages 67–70.

Daoud, M., Tamine-Lechani, L., Boughanem, M., and Bilal, C. (2008). Construction des profils utilisateurs à base d'une ontologie pour une recherche d'information personnalisée. In *francophone en Recherche d'Information et Applications (CORIA 2008)*.

Das, D., Sahoo, L., and Datta, S. (2017). A survey on recommendation system. *International Journal of Computer Applications*, 160(7):6–10.

Das, T. and Mohapatro, A. (2014). A study on big data integration with data warehouse. *International Journal of Computer Trends and Technology*, 9(4):188–192.

Debortoli, S., Müller, O., and vom Brocke, J. (2014). Comparing business intelligence and big data skills. *Business & Information Systems Engineering*, 6(5):289–300.

Decker, R. and Trusov, M. (2010). Estimating aggregate consumer preferences from online product reviews. *International Journal of Research in Marketing*, 27(4):293–307.

Dey, A. K. (2001). Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7.

Di Tria, F., Lefons, E., and Tangorra, F. (2018). A proposal of methodology for designing big data warehouses. *Preprints*, https://doi.org/10.20944/preprints201806.0219.v1.

Domshlak, C. and Joachims, T. (2007). Efficient and non-parametric reasoning over user preferences. *User Modeling and User-Adapted Interaction*, 17(1-2):41–69.

Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons, 2 edition.

Dursun, D. and Hamed, Z. (2018). The analytics paradigm in business research. *Journal of Business Research*, 90:186–195.

EDUCBA (2019). Business intelligence vs big data. `https://www.educba.com/business-intelligence-vs-big-data/`. Accessed: 2019-03-30.

Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011). Collaborative filtering recommender systems. *Found. Trends Hum.-Comput. Interact.*, 4(2):81–173.

Elbashir, M. Z., Collier, P. A., and Davern, M. J. (2008). Measuring the effects of business intelligence systems: The relationship between business process and organizational performance. *International Journal of Accounting Information Systems*, 9(3):135–153.

Facebook.com (2018). Stats. `http://newsroom.fb.com/companyinfo/`. Accessed: 2019-03-30.

Fang, H. (2008). A re-examination of query expansion using lexical resources. In *Proceedings of ACL-08: HLT*, pages 139–147, Columbus, Ohio. Association for Computational Linguistics.

Fekete, D. and Vossen, G. (2015). The gobia method: Towards goal-oriented business intelligence architectures. In *Proceedings of LWA*, pages 409–418, Trier, Germany.

Fernández-Reyes, F. C., Valadez, J. H., and y Gómez, M. M. (2018). A prospect-guided global query expansion strategy using word embeddings. *Information Processing Management*, 54:1–13.

Franco (2014). Les 5 défis du big data selon talend. `https://fr.blog.businessdecision.com/bigdata/2015/03/defis-big-data`. Accessed: 2019-03-30.

Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29:131–163.

Garrigós, I., Pardillo, J., Mazón, J.-N., and Trujillo, J. (2009). A conceptual modeling approach for olap personalization. In Laender, A. H. F., Castano, S., Dayal, U., Casati, F., and de Oliveira, J. P. M., editors, *Conceptual Modeling - ER 2009*, pages 401–414, Berlin, Heidelberg. Springer Berlin Heidelberg.

Gartner (2015). Gartner says 6.4 billion connected "things" will be in use in 2016, up 30 percent from 2015. `http://www.gartner.com/newsroom/id/316531`. Accessed: 2019-03-30.

Gauch, S., Chaffee, J., and Pretschner, A. (2003). Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems: An international Journal*, 1(3, 4):219–234.

Ghemawat, S., Gobioff, H., and Leung, S.-T. (2003). The google file system. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, pages 20–43, Bolton Landing, NY.

Giacometti, A., Marcel, P., and Negre, E. (2008). A framework for recommending olap queries. In *Proceedings of the ACM 11th International Workshop on Data Warehousing and OLAP*, DOLAP '08, pages 73–80, New York, NY, USA. ACM.

Giacometti, A., Marcel, P., and Negre, E. (2009). Recommending multidimensional queries. In Pedersen, T. B., Mohania, M. K., and Tjoa, A. M., editors, *Data Warehousing and Knowledge Discovery*, pages 453–466, Berlin, Heidelberg. Springer Berlin Heidelberg.

Glogowski, J. (2014). Keeping up with the quants: Your guide to understanding and using analytics by thomas h. davenport and jinho kim. *Journal of Business & Finance Librarianship*, 19(1):86–89.

Godoy, D. and Amandi, A. (2008). Hybrid content and tag-based profiles for recommendation in collaborative tagging systems. *2008 Latin American Web Conference*, pages 58–65.

Golfarelli, M. (2010). From user requirements to conceptual design in warehouse design: A survey. *Data Warehousing Design and Advanced Engineering Applications: Methods for Complex Construction*, pages 1–16.

González, G., López, B., and De La Rosa, J. L. (2002). The emotional factor: An innovative approach to user modelling for recommender systems. In *Workshop on Recommendation and Personalization in e-Commerce*, pages 90–99.

Gonzalo, J., Verdejo, F., Chugur, I., and Cigarrán, J. M. (1998). Indexing with wordnet synsets can improve text retrieval. *CoRR*, cmp-lg/9808002:1–7.

Gorrab, A., Kboubi, F., and Ghézala, H. B. (2019). Social Information Retrieval and Recommendation: state- of-the-art and future research. *Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées*, Volume 27 - 2017 - Special issue CARI 2016.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., and Pirahesh, H. (1997). Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery*, 1(1):29–53.

Gu and Li (2013). Memory or time: Performance evaluation for iterative operation on hadoop and spark. In *2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing*, pages 721–727.

Gupta, B., Goul, M., and Dinter, B. (2015). Business intelligence and big data in higher education: Status of a multi-year model curriculum development effort for business school undergraduates, ms graduates, and mbas. *CAIS*, 36:23.

Hollingsworth, M. R. (2012). Hadoop and hive as scalable alternatives to rdbms: A case study.

Hsu, M.-H., Tsai, M.-F., and Chen, H.-H. (2006). Query expansion with conceptnet and wordnet: An intrinsic comparison. In *Information Retrieval Technology*, pages 1–13, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hu, B., Ma, Y., Zhang, L.-J., Shi, J., and Zhong, J. (2014). A key-value based application platform for enterprise big data. In *2014 IEEE International Congress on Big Data*, pages 446–453. IEEE.

Hu, R. and Pu, P. (2009). Acceptance issues of personality-based recommender systems. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 221–224, New York, NY, USA. ACM.

Huang, A. (2008). Similarity measures for text document clustering. In *Proceedings of the Sixth New Zealand Computer Science Research Student Conference*, pages 49–56, Christchurch, New Zealand.

IBM (2012). The four v's of big data. `https://www.ibmbigdatahub.com/infographic/four-vs-big-data`. Accessed: 2019-03-30.

Inmon, W. H., Terdeman, R. H., Montanari, J., and Meers, D. (2001). *Data Warehousing for E-Business*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition.

internetlivestats.com (2018). Twitter statistics. `http://www.internetlivestats.com/twitter-statistics/`. Accessed: 2019-03-30.

Ioannidis, Y. and Koutrika, G. (2005). Personalized systems: Models and methods from an ir and db perspective. In *Proceedings of the 31st international conference on Very large data bases*, pages 1365–1365. VLDB Endowment.

Jain, A. (2013). *Instant Apache Sqoop: Transfer data efficiently between RDBMS and the Hadoop ecosystem using the robust Apache Sqoop*. Packt Publishing Ltd.

Jalali, M., Mustapha, N., Sulaiman, M. N., and Mamat, A. (2010). Webpum: A web-based recommendation system to predict user future movements. *Expert Syst. Appl.*, 37:6201–6212.

Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., and Robinson, W. (2011). The brave new world of design requirements. *Information Systems*, 36(7):992–1008.

Jerbi, H. (2012). *Personnalisation d'analyses décisionnelles sur des données multidimensionnelles*. PhD thesis. Thèse de doctorat dirigée par Zurfluh, Gilles Informatique Université Toulouse 1 Capitole 2012.

Jerbi, H., Pujolle, G., Ravat, F., and Teste, O. (2010). Personnalisation de systèmes OLAP annotés. *CoRR*, abs/1005.0198:1–15.

Jerbi, H., Ravat, F., Teste, O., and Zurfluh, G. (2008). Management of context-aware preferences in multidimensional databases. In *2008 Third International Conference on Digital Information Management*, pages 669–675.

Kaur, K. and Bharti, V. (2019). *A Survey on Big Data—Its Challenges and Solution from Vendors*, pages 1–22. Springer Singapore, Singapore.

Khan, N., Alsaqer, M., Shah, H., Badsha, G., Abbasi, A. A., and Salehian, S. (2018). The 10 vs, issues and challenges of big data. In *Proceedings of the 2018 International Conference on Big Data and Education*, pages 52–56. ACM.

Khemiri, R. (2015). *Vers l'OLAP collaboratif pour la recommandation des analyses en ligne personnalisées.* PhD thesis, Lyon 2.

Kimball and Caserta (2011). *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting.* John Wiley.

Kimball, R. et al. (1996). *The data warehouse toolkit: practical techniques for building dimensional data warehouses*, volume 1. John Wiley & Sons New York.

Kimble, C. and Milolidakis, G. (2015). Big data and business intelligence: Debunking the myths. *Global Business and Organizational Excellence*, 35(1):23–34.

Kostadinov, D. (2003). *La personnalisation de lŠinformation, définition de modèle de profil utilisateur. rapport de dea.* PhD thesis, MasterŠs thesis, Université de Versailles, France.

Kostadinov, D. (2007). *Personnalisation de l'information: une approche de gestion de profils et de reformulation de requêtes.* PhD thesis, Université de Versailles-Saint Quentin en Yvelines.

Koutrika, G. and Ioannidis, Y. (2004). Personalization of queries in database systems. In *Proceedings. 20th International Conference on Data Engineering*, pages 597–608. IEEE.

Koutrika, G. and Ioannidis, Y. (2005). Personalized queries under a generalized preference model. In *21st International Conference on Data Engineering (ICDE'05)*, pages 841–852. IEEE.

Kowalczyk, M. and Buxmann, P. (2014). Big data and information processing in organizational decision processes. *Business & Information Systems Engineering*, 6(5):267–278.

Kraft, R., Maghoul, F., and Chang, C. C. (2005). Y! q: contextual search at the point of inspiration. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 816–823. ACM.

Krishnan, K. (2013). Chapter 10 - integration of big data and data warehousing. In Krishnan, K., editor, *Data Warehousing in the Age of Big Data*, MK Series on Business Intelligence, pages 199 – 217. Morgan Kaufmann, Boston.

Kuldeep, D. and Bhimappa, D. (2014). Limitations of data warehouse platforms and assessment of hadoop as an alternative. *International Journal of Information Technology and Management Information System*, 5(2):51–58.

Kulkarni, R. and Inc., S. I. (2013). Transforming the data deluge into data-driven insights: Analytics that drive business. In *Keynote Speech presented at the 44th Annual Decision Sciences Institute Meeting, Baltimore, MD*.

Kwok, K. L. (1998). Book review: Information storage and retrieval by r. r. korfhage. *Inf. Process. Manage.*, 34(4):490–492.

Lam, S. K. T., Frankowski, D., and Riedl, J. (2006). Do you trust your recommendations? an exploration of security and privacy issues in recommender systems. In *Emerging Trends in Information and Communication Security*, pages 14–29, Berlin, Heidelberg.

Laney, D. (2001). 3d data management: Controlling data volume, velocity and variety. *META group research note*, 6(70):1.

Lara Pahins, C. A., Ferreira, N., and Comba, J. (2019). Real-time exploration of large spatiotemporal datasets based on order statistics. *IEEE Transactions on Visualization and Computer Graphics*, 14(8):1–12.

Lee, I. (2017). Big data: Dimensions, evolution, impacts, and challenges. *Business Horizons*, 60(3):293–303.

Lustig, I., Dietrich, B., Johnson, C., and Dziekan, C. (2010). The analytics journey. *Analytics Magazine*, 3(6):11–13.

Mandal, D., Dandapat, S., Gupta, M., Banerjee, P., and Sarkar, S. (2007). Bengali and hindi to english cross-language text retrieval under limited resources. In *CLEF (Working Notes)*.

Marjanovic, O. (2013). Sharing and reuse of innovative teaching practices in emerging business analytics discipline. In *2013 46th Hawaii International Conference on System Sciences*, pages 50–59. IEEE.

Marjanovic, O. (2015). From analytics-as-a-service to analytics-as-a-consumer-service: exploring a new direction in business intelligence and analytics research. In *2015 48th Hawaii International Conference on System Sciences*, pages 4742–4751. IEEE.

Mashingaidze, K. and Backhouse, J. (2017). The relationships between definitions of big data, business intelligence and business analytics: a literature review. *International Journal of Business Information Systems*, 26(4):488–505.

Mc Gowan, J. P. (2003). *A multiple model approach to personalised information access.* PhD thesis, Citeseer.

Menaceur, S., Derdour, M., and Bouramoul, A. (2016). Olaping and big data mining: A survey. In *The 2nd International Conference on Pattern Analysis and Intelligent Systems*, PAIS '2016, Khenchela, Algeria.

Menaceur, S., Derdour, M., and Bouramoul, A. (2017a). Personalized online analytical processing in big data context using user profile and search context. *International Journal of Strategic Information Technology and Applications (IJSITA)*, 8(4):67–80.

Menaceur, S., Derdour, M., and Bouramoul, A. (2017b). Vers une approche pour la prise en compte de l'usager dans l'analyse olap. In *La 11éme édition de la conférence maghrébine sur les Avancées des Systèmes Décisionnels*, ASD '2017, Tabarka, Tunisie.

Menaceur, S., Derdour, M., and Bouramoul, A. (2019). Using query expansion techniques and content-based filtering for personalizing analysis in big data. *International Journal of Information Technology and Web Engineering (IJITWE).*, Sous Presse.

Mervis, J. (2012). Agencies rally to tackle big data. *Science*, 336(6077):22–22.

Miyanishi, T., Seki, K., and Uehara, K. (2013). Improving pseudo-relevance feedback via tweet selection. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 439–448, New York, NY, USA. ACM.

Mohanty, S., Jagadeesh, M., and Srivatsa, H. (2013). *Big Data Imperatives: Enterprise Big Data Warehouse, BI Implementations and Analytics.* Apress, Berkely, CA, USA, 1st edition.

Mortenson, M. J., Doherty, N. F., and Robinson, S. (2015). Operational research from taylorism to terabytes: A research agenda for the analytics age. *European Journal of Operational Research*, 241(3):583–595.

On-At, S. (2017). *Temporalité et réseaux sociaux: prise en compte de l'évolution dans la construction du profil utilisateur.* PhD thesis, Université de Toulouse, Université Toulouse III-Paul Sabatier.

Oussous, A., Benjelloun, F.-Z., Lahcen, A. A., and Belfkih, S. (2018). Big data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, 30(4):431–448.

Pal, D., Mitra, M., and Datta, K. (2014). Improving query expansion using wordnet. *JASIST*, 65:2469–2478.

Parapar, J., Quindimil, M. A. P., and Barreiro, A. (2014). Score distributions for pseudo relevance feedback. *Inf. Sci.*, 273:171–181.

Pathak, B., Garfinkel, R., Gopal, R. D., Venkatesan, R., and Yin, F. (2010). Empirical analysis of the impact of recommender systems on sales. *Journal of Management Information Systems*, 27(2):159–188.

Phillips-Wren, G. E., Iyer, L. S., Kulkarni, U. R., and Ariyachandra, T. (2015). Business analytics in the context of big data: A roadmap for research. *CAIS*, 37:23.

Proulx, M. J., . B. Y. (2004). Le potentiel de l'approche multidimensionnelle pour l'analyse de données géospatiales en comparaison avec l'approche transactionnelle des sig. In *In Colloque Géomatique*, pages 27–28.

Pu, P., Chen, L., and Hu, R. (2011). A user-centric evaluation framework for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 157–164, New York, NY, USA. ACM.

Radha, K. and Rao, B. T. (2016). A study on big data techniques and applications. *Int. J. Adv. Appl. Sci*, 5:101–108.

Ranawade, S. V., Navale, S., Dhamal, A., Deshpande, K., and Ghuge, C. (2017). Online analytical processing on hadoop using apache kylin. *International Journal of Applied Information Systems*, 12(2):1–5.

Ravat, F., Teste, O., and Zurfluh, G. (2007). Personnalisation de bases de données multidimensionnelles. In *Congrès Informatique des Organisations et Systèmes d'Information et de Décision - INFORSID'07*, pages 231–247, Perros-Guirec, France.

Rob, P., Coronel, C., and Crockett, K. (2008). *Database systems: design, implementation & management*. Cengage Learning EMEA.

Russom, P. et al. (2011). Big data analytics. *TDWI best practices report, fourth quarter*, 19(4):1–34.

Samadi, Y., Zbakh, M., and Tadonki, C. (2018). Performance comparison between hadoop and spark frameworks using hibench benchmarks. *Concurrency and Computation: Practice and Experience*, 30(12):1–13.

Sangupamba Mwilu, O. (2018). *De la business intelligence interne vers la business intelligence dans le cloud: modèles et apports méthodologiques*. PhD thesis, Paris, CNAM.

Santos, M. Y., Martinho, B., and Costa, C. (2017). Modelling and implementing big data warehouses for decision support. *Journal of Management Analytics*, 4(2):111–129.

Sarraj, L. E., Espinasse, B., and Libourel, T. (2014). Personnalisation de l'exploitation d'un entrepôt de données dirigée par des ontologies : Application au management hospitalier. In *EDA*, volume B-10 of *RNTI*, pages 93–102. Hermann-Éditions.

SAS (2013). Five big data challenges. `https://www.sas.com/resources/asset/five-big-data-challengesarticle.pdf`. Accessed: 2019-03-30.

Sawant, N. and Shah, H. (2013). *Big Data Application Architecture*, pages 9–28. Apress, Berkeley, CA.

Schmidt, A., Aidoo, K. A., Takaluoma, A., Tuomela, U., Van Laerhoven, K., and Van de Velde, W. (1999). Advanced interaction in context. In *International Symposium on Handheld and Ubiquitous Computing*, pages 89–101. Springer.

Selene Xia, B. and Gong, P. (2014). Review of business intelligence through data analysis. *Benchmarking: An International Journal*, 21(2):300–311.

Sharma, S., Tim, U. S., Wong, J., Gadia, S., and Sharma, S. (2014). A brief review on leading big data models. *Data Science Journal*, 13:138–157.

Siddiqa, A., Karim, A., and Gani, A. (2017). Big data storage technologies: a survey. *Frontiers of Information Technology & Electronic Engineering*, 18(8):1040–1070.

Silva, C. and Ribeiro, B. (2003). The importance of stop word removal on recall values in text categorization. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 1661–1666. IEEE.

Singh, V. K., Taram, M., Agrawal, V., and Baghel, B. S. (2018). A literature review on hadoop ecosystem and various techniques of big data optimization. In *Advances in Data and Information Sciences*, pages 231–240, Singapore. Springer Singapore.

Smeaton, A. F., Kelledy, F., and O'Donnell, R. (1995). Trec-4 experiments at dublin city university: Thresholding posting lists, query expansion with wordnet and pos tagging of spanish. *Harman*, pages 373–389.

Song, J., Guo, C., Wang, Z., Zhang, Y., Yu, G., and Pierson, J.-M. (2015). Haolap: a hadoop based olap system for big data. *Journal of Systems and Software*, 102:167–181.

Stefanidis, K. and Pitoura, E. (2008). Fast contextual preference scoring of database tuples. In *Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '08, pages 344–355, New York, NY, USA. ACM.

Strehl, A., Ghosh, J., and Mooney, R. (2000). Impact of similarity measures on web-page clustering. In *Workshop on artificial intelligence for web search (AAAI 2000)*, volume 58, page 64.

Sun, Y., Li, H., Councill, I. G., Huang, J., Lee, W.-C., and Giles, C. L. (2008). Personalized ranking for digital libraries based on log analysis. In *Proceedings of the 10th ACM workshop on Web information and data management*, pages 133–140. ACM.

Takeuchi, S., Sugiura, K., Akahoshi, Y., and Zettsu, K. (2017). Spatio-temporal pseudo relevance feedback for scientific data retrieval. *IEEJ Transactions on Electrical and Electronic Engineering*, 12(1):124–131.

Talan, P. P., Sharma, K. U., Nawade, P. P., and Talan, K. P. (2019). An overview of hadoop mapreduce, spark, and scalable graph processing architecture. In *Recent Developments in Machine Learning and Data Analytics*, pages 35–42. Springer.

Tewari, A. S., Singh, J. P., and Barman, A. G. (2018). Generating top-n items recommendation set using collaborative, content based filtering and rating variance. *Procedia computer science*, 132:1678–1684.

Ting-Peng, L. and Yu-Hsi, L. (2018). Research landscape of business intelligence and big data analytics: A bibliometrics study. *Expert Syst. Appl.*, 111:2–10.

Torlone, R. (2003). Conceptual multidimensional models. In *Multidimensional databases: Problems and solutions*, pages 69–90. IGI Global.

Ularu, E. G., Puican, F. C., Apostu, A., Velicanu, M., et al. (2012). Perspectives on big data and big data analytics. *Database Systems Journal*, 3(4):3–14.

Voorhees, E. M. (1994). Query expansion using lexical-semantic relations. In *SIGIR '94*, pages 61–69, London. Springer.

Wahlster, W. and Kobsa, A. (1986). Dialogue-based user models. *Proceedings of the IEEE*, 74(7):948–960.

Wang, B., Gui, H., Roantree, M., and O'Connor, M. F. (2014). Data cube computational model with hadoop mapreduce. In *WEBIST (1)*, pages 193–199. SciTePress.

Wang, H., Xu, Z., Fujita, H., and Liu, S. (2016). Towards felicitous decision making: An overview on challenges and trends of big data. *Information Sciences*, 367:747–765.

White, T. (2012). *Hadoop: The definitive guide.* " O'Reilly Media, Inc.", second edition.

worldwidewebsize.com (2018). The size of the world wide web (the internet). `http://www.worldwidewebsize.com/`. Accessed: 2019-03-30.

Xia, T. (2008). Large-scale sms messages mining based on map-reduce. In *2008 International Symposium on Computational Intelligence and Design*, volumc 1, pagcs 7–12. IEEE.

YouTube.com (2018). Statistics-youtube. `http://www.youtube.com/yt/press/statistics.html`. Accessed: 2019-03-30.

Zhou, D., Wu, X., Zhao, W., Lawless, S., and Liu, J. (2017). Query expansion with enriched user profiles for personalized search utilizing folksonomy data. *IEEE Transactions on Knowledge and Data Engineering*, 29:1536–1548.