حالجمـــهوريـــة الجـــزائريـــة الديـمقراطيـــــــة الشـــعبيـــــة
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعـلـيـم الـعـالـــي و الـبـحـث الـعـلـمـي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جـامـعـة عمّـــار ثـلـيـــجـــي بالأغـــواط
UNIVERSITE AMAR TELIDJI LAGHOUAT
كـلـيـــــة الـعـلـــــوم
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

Mémoire de MASTER

Domain : Mathématiques et Informatique
Filière : Informatiques
Option : Systèmes d'Information et de Décision

Par:
- Chemma Bachir
- Boukhari Ali

THEME

L'utilisation des règles d'association pour l'agrégation textuelle

---

Soutenu publiquement le 12-06-2017 devant le jury composé de:

Mr M. H.MAICHA	*M.C.(A)*	*Président*
Mr M.BOUAAKAZ	*M.C.(A)*	*Examinateur*
Mr Y.OUINTEN	*M.C.(A)*	*Encadreur*

Année Universitaire 2016/2017

Dédicaces

Nous dédions ce mémoire
à
nos parents

Nos frères et sœurs

Et
nos amis

Remerciements

Je remercie tout d'abord ALLAH, le tout puissant de m'avoir donné la force et la patience et de m'avoir rapproché des personnes qui m'ont soutenu et aidé pour accomplir ce travail.

Mes remerciements s'adressent également à toutes les personnes qui ont contribué de près ou de loin avec leurs conseils ou avec leurs encouragements à l'accomplissement de ce travail.

Je tiens à exprimer ma sincère reconnaissance et remerciements à Mr.Youcef OUINTEN, professeur à l'université de Laghouat d'avoir accepté d'encadrer et de diriger nos travaux.

Enfin, j'exprime mes vifs remerciements à toute ma famille et spécialement à mes parents, mon père AHMIDA et à ma mère BENZIANE HAFSSA que je leur souhaite une longue vie pleine de bonheur, de santé et de prospérité.

Bachir Chemma B_30.

Remerciements

Je remercie tout d'abord ALLAH, le tout puissant de m'avoir donné la force et la patience et de m'avoir rapproché des personnes qui m'ont soutenu et aidé pour accomplir ce travail

Mes remerciements s'adressent également à toutes les personnes qui ont contribué de près ou de loin avec leurs conseils ou avec leurs encouragements à l'accomplissement de ce travail.

Je tiens à exprimer ma profonde gratitude à Monsieur Youcef OUINTEN, professeur à l'université de Laghouat d'avoir accepté d'encadrer et de diriger nos travaux.

Un très grand merci à mes parents toujours présents mon père BOUKHARI MOHAMMED et ma mère LAIFA BOUCENNA par leur soutien et leur encouragement, dans les moments difficiles de la réalisation de ce mémoire.

Ali Boukhari.

ملخص:

تشهد المعلومات النصية يوما بعد يوم تصاعدا سريعا هذا الأخير جر معه العديد من الأبحاث قصد معالجة مجموعة من المشاكل، ومن بين هذه المشاكل ما يعرف بالتجميع النصي «l'agrégation textuelle» التي تستعمل لإعطاء نظرة أكثر تنظيما وشمولية للبيانات النصية.

في هذه المذكرة قمنا بتقديم احدى تقنيات التنقيب عن المعلومة والمعروفة بالعلاقات الترابطية «règle d'association» من أجل القيام بالتجميع النصي، وبعدها قمنا بدراسة مقارنة بين هاته التقنية وتقنية تعتمد على الوحدات المتكررة «fréquent Itemset».

استعملنا من أجل استخراج العلاقات الترابطية مجموعة برامج مخصصة للتنقيب عن البيانات على عينة من معطيات نصية حقيقية، من أجل ملاحظة مدى فاعلية العلاقات الترابطية في مجال التجميع النصي.

في نهاية هذا العمل خلصنا الى أن التقنية المعتمدة على العلاقات الترابطية أعطت نتائج جيدة في مجال التجميع النصي، وذلك على مستوى جميع مقاييس الأداء.

كلمات مفتاحية: التنقيب عن البيانات، التنقيب النصي، التجميع النصي، العلاقات الترابطية، الوحدات المتكررة، معطيات نصية

Résumé :

Jour après jour les données textuelle ont vu un a croissement rapide, ce dernier a engendré plusieurs travaux de recherche pour traiter un ensemble de problèmes, l'un de ces problèmes est l'agrégation textuelle qui sert à obtenir une vision globale plus agrégée et plus synthétique d'un ensemble de documents.

Dans ce mémoire nous avons présenté la technique de data mining des règles d'association pour faire une agrégation textuelle. Ensuite nous avons effectué une étude comparative entre cette technique et la technique basée sur l'extraction des motifs fréquents.

Pour l'extraction des règles d'association et les motifs fréquents nous utilisons des outils spécialisés au domaine de fouille de donnée sur un corpus de documents réels pour voire l'effet des règles d'association dans le domaine d'agrégation textuelle.

A la fin de ce mémoire on a conclu que la technique basée sur les règles d'association devient une des méthodes acceptable dans le domaine d'agrégation textuelle car elle donne de bons résultats pour toutes les mesures de performances.

Mots clés : Data mining, Text mining, agrégation textuelle, règles d'association, motifs fréquents.

Abstract:

Day after day, the textual data see a rapid increase; this latter generated more works to treat a set of problems. One of these problems is the textual aggregation, which aims at obtaining a global and more synthetic vision of data.

In this thesis, we present the data mining technique known as association rules to perform textual aggregation. We, also, carry a comparative study between this technique and the technique based on the extraction of frequent itemsets.

For the extraction of the association rules and frequent itemsets, we used tools dedicated to data mining domain on a corpus of reel documents to see the effect of association rules in the textual aggregation domain.

In the end of this thesis, we concluded that the technique based on association rules is an acceptable method in the textual aggregation domain, because it gives acceptable results for all measures of performance.

Keywords: Data mining, Text mining, textual aggregation, association rules, frequent patterns

Tables des matières :

Liste des tableaux

Liste des Figure

Introduction

Introduction

La recherche accorde ces dernières années, beaucoup d'importance au traitement des données textuelles. Ceci pour plusieurs raisons : un nombre croissant de collections mises en réseau et distribuées au plan international, le développement de l'infrastructure de communication et de l'Internet. Les traitements manuels de ces données s'avèrent très coûteux en temps et en personnel, ils sont peu flexibles et leur généralisation à d'autres domaines est presque impossible ; c'est pour cela que l'on cherche à mettre au point des méthodes automatiques.

Le domaine de la fouille de textes (text mining) s'est développé pour répondre à la volonté de gestion par contenu des sources volumineuses de textes. A l'heure actuelle, de nombreux logiciels d'agrégation de textes sont disponibles, ils ont fait l'objet de publications et leurs champs d'application s'élargissent de jour en jour. En général, ces systèmes sont basés sur des algorithmes d'apprentissage automatique (approche statistique, approche syntaxique et approche externe.)

Nous nous intéressons ici plus particulièrement à l'algorithme Apriori pour l'extraction des règles d'association et l'utilisation de ces dernier dans l'agrégation des documentes textuelle Nous résumons la présentation de notre travail ainsi :

Dans le premier chapitre nous présentons une initiation à la fouille de texte et quelque tâche élémentaire pour la fouille de texte.

Dans le deuxième chapitre nous avons exposé un état de l'art sur le domaine de l'agrégation des documentes textuelle.

Dans le troisième chapitre nous expliquons d'une part les règles d'association et quelque notation, et d'autre part une collection de mesures de performances.

Dans le quatrième chapitre nous avons exposé une partie expérimentale détaillée avec une étude comparative entre deux techniques la première technique basée sur motifs fréquents et la deuxième technique basée sur les règles d'association.

Chapitre 1

Initiation a fouille du text

1.1 Introduction à la fouille de données :

1.1.1Définition de la fouille de données :

La fouille de données est un domaine qui est apparu avec l'explosion des quantités d'informations stockées, avec le progrès important des vitesses de traitement et des supports de stockage. La fouille de données vise à découvrir, dans les grandes quantités de données, les informations précieuses qui peuvent aider à comprendre les données ou à prédire le comportement des données futures.

La fouille de données utilise depuis son apparition plusieurs outils de statistiques et d'intelligence artificielle pour atteindre ses objectifs.

La fouille de données s'intègre dans le processus d'extraction des connaissances à partir des données ECD ou (KDD : Knowledge Discovery from Data en anglais). Ce domaine en pleine expansion est souvent appelé le data mining. [3]

La figure 1.1 représente Schéma global de l'ECD.

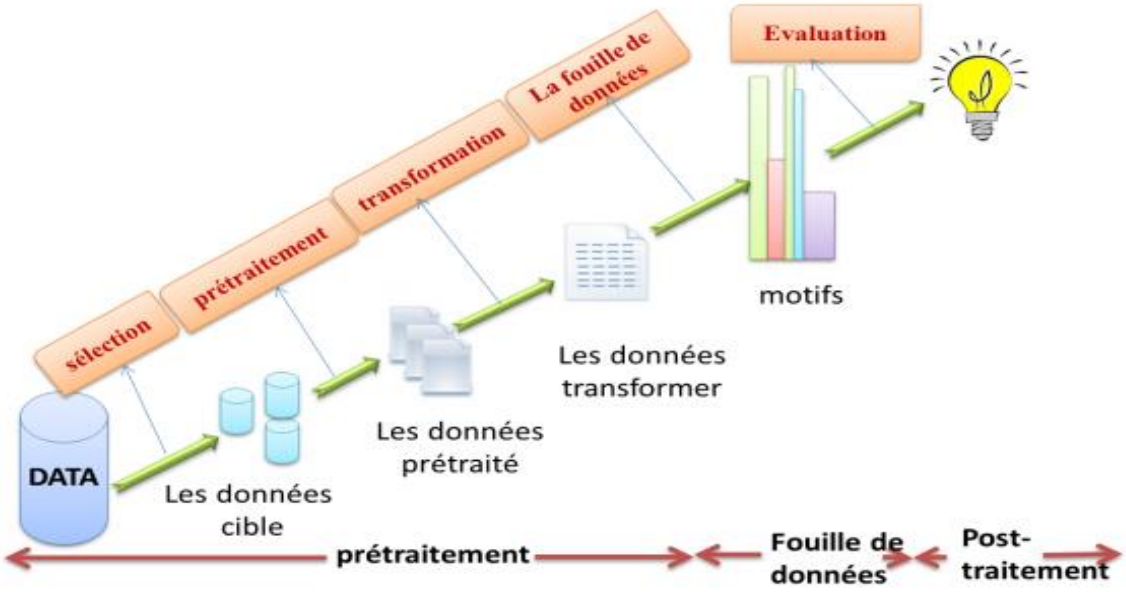


Figure 1.1 : Schéma global de l'ECD d'après [34]

1.1.2 tâches de la fouille de données :

Beaucoup de problèmes intellectuels, économiques ou même commerciaux peuvent être exprimés en termes des six tâches suivantes :

- La classification.
- L'estimation.
- Le groupement par similitude (règles d'association).
- L'analyse des clusters.
- La description.

1.1.3 Les limites de la fouille de données :

- Le plus grand obstacle de la fouille de données réside dans un accès à de grandes quantités de données. Cela trouve son explication dans la nature statistique des calculs effectués.
- Le deuxième plus grand obstacle réside dans l'énorme puissance informatique requise pour traiter ces gigantesques quantités de données.
- Finalement, le choix des variables et divers indicateurs influence énormément la qualité des modèles extraits des données. La puissance de calcul phénoménale des ordinateurs actuels ne remplace pas l'expertise d'un spécialiste du domaine d'application [26].

1.2 Fouille de données textuelles : Text Mining :

La fouille de données textuelles (ou forage de texte) que l'on peut traduire de l'anglais par «Text Mining» nous permet de déterminer le sens d'un texte sans nécessairement en lire tout le contenu dans le but de découvrir des informations cachées ou prendre automatiquement la bonne décision.

D'une manière plus précise, la fouille de données textuelles désigne l'ensemble des techniques et des méthodes destinées au traitement automatique de données textuelles non structurées, disponibles sous forme informatique, en assez grande quantité. Il s'agit de les organiser et de les structurer afin d'en dégager des thématiques, des relations dans une perspective d'analyse non littéraire rapide.

1.3 Les tâches élémentaires de la fouille de textes :

Les tâche de fouille de texte sont des briques de base au plusieurs tâches plus complexe, dans ce cadre nous définissons quatre tâches qui sont : La recherche d'information, la classification, l'extraction d'information, et l'agrégation.

Nous les présenterons dans cet ordre, de la moins spécifique à la plus spécifique d'un point de vue linguistique. Pour chacune d'entre elles, nous explicitons ici leur nature et leur intérêt applicatif [19].

1.3.1 La Recherche d'Information (RI) :

Pour la Recherche d'Information (ou RI, ou IR pour Information Retrieval en anglais). Le but de cette tâche est de retrouver un ou plusieurs document(s) pertinent(s) dans un corpus, à l'aide d'une requête plus ou moins informelle [6].

Dans [15] «La recherche d'informations (RI) traite de la représentation, du stockage, de l'organisation et de l'accès à l'information. Le but d'un système de recherche d'informations est de retrouver, parmi une collection de documents préalablement stockés, les documents qui répondent au besoin d'utilisateur exprimé sous forme de requête».

1.3.1.1 Les domaines d'application :

La RI est une tâche très populaire, à laquelle tous les usagers d'Internet font appel quotidiennement dès qu'ils utilisent un moteur de recherche. Ceux-ci appartiennent à plusieurs familles.

Il y a bien sûr, les moteurs généralistes (Google, Bing, Yahoo !,Baidu en Chine...) qui servent à s'orienter sur l'ensemble du Web, mais la plupart des sites importants (notamment tous les sites marchands ou institutionnels disposent aussi d'un moteur interne permettant de naviguer à l'intérieur de leurs pages.

Tout internaute sollicite donc quotidiennement, parfois sans le savoir, plusieurs moteurs de recherche. Des systèmes de recherche sont aussi intégrés au cœur même de chaque ordinateur,

pour aider l'utilisateur à fouiller dans son disque dur à la recherche d'un fichier ou d'un mail mal rangé.

Enfin, la RI existait déjà avant même l'invention du Web, dans le domaine des "sciences de la documentation". Elle était dans ce cadre cantonnée aux archives et aux bibliothèques, pionnières en matière d'indexation et de requête de corpus de textes numérisés. Plutôt que de moteurs de recherche, on parlait alors de "logiciels documentaires". [6]

Nous verrons que les techniques utilisées pour construire un programme de RI dans ces différents contextes peuvent varier, mais restent assez homogènes.

1.3.2 La Classification :

La classification est la tâche phare de la fouille de données, pour laquelle une multitude de programmes sont implémentés dans le logiciel Weka. Elle consiste à associer une "classe" à chaque donnée d'entrée. Comme l'illustre la figure 1.2.

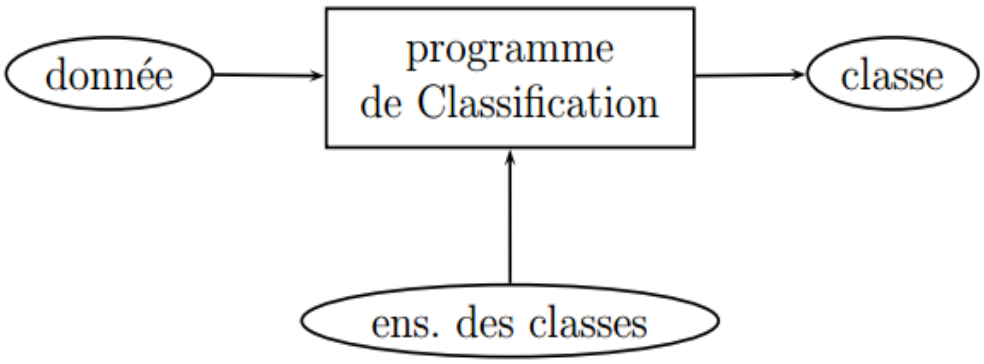


Figure 1.2 : schéma globale de classification [6]

Comme la recherche d'information, la classification peut s'appliquer à toutes sortes de données, et pas seulement aux textes : la classification des images, des vidéos, des musiques... de toute donnée, de manière générale, qu'il est possible de décrire à l'aide d'attributs, donne lieu à de multiples et florissantes applications.

La fouille de données était née dans les domaines des banques, des assurances, du marketing et de la médecine, pour aider à déterminer automatiquement la solvabilité d'un client, l'adéquation d'un produit ou encore l'efficacité d'un médicament. Tous ces objectifs peuvent être reformulés comme des tâches de classification.

1.3.2.1 Les domaines d'application :

La classification est une tâche qui donne lieu à une multitude d'applications. L'une d'elles est présente dans la plupart des gestionnaires de courriers électroniques : c'est la reconnaissance automatique des "spam", ces messages indésirables qui encombrent toutes les boites aux lettres. Cette fonctionnalité est généralement implémentée en mode "apprentissage automatique", l'utilisateur devant, au début, signaler ce qu'il considère comme indésirable afin que le programme apprenne progressivement à les reconnaitre lui-même.

La "fouille d'opinion" est un autre domaine d'application en plein essor. Elle vise à identifier les polarités (positives ou négatives) véhiculées par les textes (par exemple les commentaires d'internautes sur les sites marchands ou de loisir).

1.3.3 L'Extraction d'Information (EI) :

L'Extraction d'Information (EI ou Information Extraction en anglais, abrège en IE) est décrite par le schéma de la figure1.3, Le but de cette tâche, qui relève de l'ingénierie linguistique est d'extraire automatiquement des documents textuels des informations factuelles servant à remplir les champs d'un formulaire prédéfini.

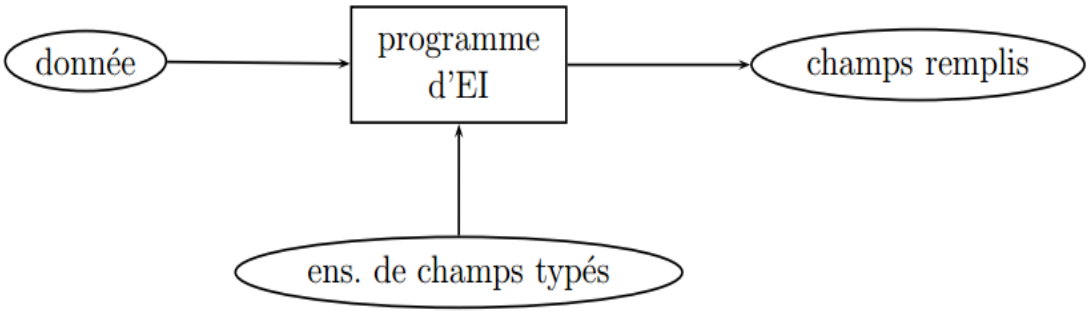


Figure 1.3 : schéma globale d'EI [6]

1.3.4 Fonction d'agrégation de données :

Il existe deux catégories de fonctions d'agrégations [14], la première englobe les fonctions opérant sur les données et mesures numériques. La deuxième représente les fonctions opérant sur les données textuelles.

1.3.4.1 Fonctions d'agrégation numériques :

Elles permettent d'effectuer, soit des opérations arithmétiques telles que les fonctions (SUM, AVG, MIN, MAX) en retournant respectivement, la somme, la moyenne, le minimum, le maximum d'un ensemble de valeurs ou d'opérations génériques telles que (COUNT, LIST),

- COUNT : la fonction de comptages qui retourne le nombre d'instance,
- LIST : liste les valeurs à agréger.

1.3.4.2 Fonction d'agrégations textuelles

Elles représentent les fonctions qui sont en mesure d'être appliquées surdes données textuelles. Quelques fonctions adaptées à ce type de données ont été proposées, par exemple, la fonction Top-Keyword [14] qui permet de retourner les k mot clés les plus représentatifs.

La fonction AVG-KW proposée dans [29], qui retourne les mots clefs d'un fragment de texte, une autre fonction qui consiste à ajouter la notion hiérarchie et niveau d'abstraction, en fournissant de divers agrégation selon le niveau d'abstraction voulu a été proposée. Il s'agit de la fonction Bien cube présenté dans [32].

1.4 Processus de la fouille de données textuelles

Le processus de fouille de textes reprend les étapes du processus de fouille de données et il en ajoute d'autres pour les adapter à son objectif tel que le type de données à analyser (données non structurées ou semi-structurées).

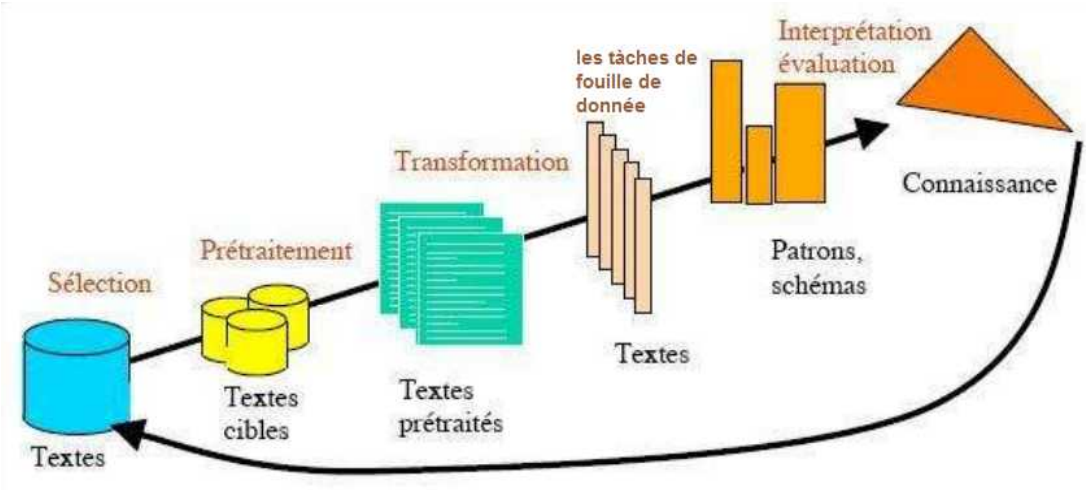


Figure 1.4 : Processus de la fouille de données textuelles [17]

Quelques systèmes de fouille de données textuelles ont pour objectif de structurer le contenu des textes en découvrant des modèles pour les décrire.

Ils se basent sur l'hypothèse d'une catégorisation a priori où il s'agit d'un prétraitement manuel des textes afin d'en extraire un certain nombre d'attributs comme les mots-clés ou les URL. Une fois les attributs extraits, les méthodes classiques de la fouille de données, telles que l'analyse statistique, les règles d'association, sont appliquées. Le processus de fouille de textes est schématisé dans la figure 1.4.

1.4.1 Le prétraitement du texte :

Dans tous les cas (utilisation de mots simple ou de groupes de mots), il est préférable d'effectuer quelque prétraitement afin de filtrer les mots non informatifs et de regrouper les mots de la même famille.

- La première opération «élimination des mots vides» consiste à supprimer les mots faisant partie d'une liste prédéfinie : liste des mots vides (Stopwords). Ce sont des mots génériques non porteurs de sens tels que les déterminants, les articles (le, la, les) et les auxiliaires qui sont a priori inutiles pour discriminer les différentes catégories. Ils peuvent donc être supprimés sans perte d'information utile.

- La deuxième opération «lemmatisation/Racinisation» consiste à conserver, non pas les mots eux-mêmes mais leur racine ou leur lemme (c'est-à-dire l'entrée-dictionnaire d'un terme), ce principe permet :

    - De prendre en compte les variations flexionnelle (singulier/pluriel, conjugaison,...) ou dérivationnelles (substantifs, verbes ou adjectives) en regroupant sous le même terme tous les mots de la même famille, par exemple «étude», «étudiant» et «étudier» pourraient être regroupés, et un seul terme serait ajouté à l'espace vectoriel plutôt que trois.

    - De baisser le nombre total de termes et donc les temps de calcul.

Dans le cas de la Racinisation, on utilise des heuristiques simples à partir de règles de remplacement des chaînes de caractères pour supprimer les affixes (suffixes et préfixes) des mots et extraire la racine [30].

L'objet d'étude de ce mémoire est l'agrégation de textes ; c'est un problème qui intéresse les chercheurs depuis relativement longtemps. On retrouve des travaux portant sur ce sujet. La recherche dans ce domaine est toujours très pertinente, car les résultats obtenus aujourd'hui sont encore sujets à amélioration.

Chapitre 2

État de l'art

Nous présentons dans ce chapitre quelques approches et techniques qui ont été proposées pour l'analyse et l'agrégation des données textuelles issues de différentes sources. Nous expliquons les principes de ces approches en définissant quelques concepts liés à ces techniques d'agrégation pour avoir une idée générale sur le domaine d'agrégation de données textuelle.

Pour l'organisation de cet état de l'art, nous avons choisis de classer les différentes approches que nous sommes en mesure de présenter selon trois principales catégories que nous allons définir brièvement.

I. Approche d'agrégation basée sur le contenu textuel :

Les travaux de recherche qui se focalisent sur les données textuelles sont très peu en comparant avec les travaux de recherche sur les autres formats de données.

Par conséquents les algorithmes qui opèrent sur les données textuelles sont peu et moins développés. Cependant il existe un ensemble de fonctions pour l'agrégation textuelles qui ont été proposées pour aide. Ces dernières ont été classées en différentes catégories. Cette classification est basée sur le principe que les algorithmes utilisent pour effectuer leur agrégation, alors on a cité les approches suivantes :

- Les approches basées sur les connaissances linguistiques.
- Les approches basées sur les connaissances externes.
- Les approches basées sur les statistiques.
- Les approches basées sur l'utilisation des graphes.

2.1 Les approches basées sur les connaissances linguistiques :

2.1.1 Analyse lexicale :

Poudat et Cleuziou en 1998 dans [10] présenté cette approchequi se concentre à réduire les textes en un ensemble de mots appelé sac de mots Bow (Bags of Words), en utilisant la classification thématique ou domaniale, chaque domaine représente ici un plan lexical.

En premier lieu ces approches ont considéré les noms comme les mots les plus représentatifs pour des concepts scientifiques ou des domaines connus, alors que les adverbes, verbes ou adjectifs ont été considérées comme des mots vides de sens. Ces noms sont facilement extraits. Ces derniers seront classés en un certain nombre de classes pour être enfin agrégées en donnant à chaque classe un nom qui peut couvrir le domaine.

Abbes et Dichy ont proposé dans [27] l'amélioration de cette approche par le calcul de fréquence des mots.

Les auteurs ont effectué une expérience d'extraction de liste de fréquences lexicales spécifiques à la langue arabe à partir d'un corpus journalistique brut de deux millions de mots (quotidien Al-Hayât) au moyen du logiciel AraConc.

AraConc est un logiciel de concordance et de calcul de fréquences pour les mots arabes. Il traite des textes écrits en arabe, et fait l'analyse mot par mot. Le logiciel fait l'extraction des mots et les stocke dans des fichiers spécifiques, en vue de différents regroupements et de l'établissement de concordances, ensuite il propose l'étiquette racine pour agréger le groupe des mots.

Dans [35] l'auteur reprend la présentation classique de corpus sous forme de sac de mots et présente un objet comme un ensemble des ensembles des sacs de mots c'est ce qui était appelé la représentation en sac-de sacs de mots BoBoW(Bag of Bags of Words). La représentation BoBoW consiste simplement à décrire un document comme un ensemble de sous-documents, chacun était représenté par un sac de mots. Le calcul de similarité entre deux objets nécessite alors d'agréger toutes les similarités entre sacs de chacun des objets.

2.1.2 Analyse syntaxique :

Dans [26], l'auteur a indiqué que l'analyse syntaxique est basée sur la sélection de syntagmes qui peuvent être définis par «l'extension logique de la sélection de catégories lexicales pouvant être des noms, des verbes, des adverbes,…etc.

En effet un syntagme peut être défini comme un ensemble de mots formants une unité catégorielle et fonctionnelle et constituant une unité sémantique.

Habert et al [16] ont exploité les relations entre les éléments de syntagmes nominaux(les noms et les adjectifs) pour l'extraction des mots-clefs. Leur principe est de réduire récursivement les termes candidats trouvés par un extracteur en mesurant la proximité entre deux mots simples par le nombre de contextes partagés par ces mots. Cette méthode n'est pas efficace, car elle nécessite d'une part l'utilisation de la sémantique pour définir les relations entre les syntagmes nominaux, et l'intervention d'un expert pour interpréter et valider les syntagmes proposés d'autre part.

Dans [11] les auteurs ont proposé une typologie qui se déroule en deux étapes essentielles : La première, concerne l'acquisition de termes, il s'agit d'une étape où les outils qui permettent l'extraction des termes candidats à partir du corpus analysé sont regroupés. La deuxième est consacrée pour la structuration des termes et le regroupement conceptuel, elle nécessite des outils d'aide à la structuration des termes ainsi que des outils de repérage de relation pour effectuer la classification de termes.

2.2 Les approches basées sur les connaissances externes :

2.2.1 Les thésaurus

L'auteur de [18] définie «le thésaurus comme étant un vocabulaire contrôlé, il rassemble un ensemble de termes structurés choisis pour leur capacité à décrire un domaine ». Le thésaurus regroupe dans une classe les termes d'un domaine particulier. Ces termes sont reliés entre eux par des relations sémantiques : liens hiérarchiques (généralisation et spécialisation), synonymie et définition [9].

2.2.2 Les ontologies :

En informatique le terme ontologie possède plusieurs définitions. La première définition a été donnée dans [31] dans le cadre du domaine d'intelligence artificielle : « une ontologie définit les termes et les relations de base de vocabulaire d'un domaine, ainsi que les règles qui indiquent comment combiner les termes et les relations de façon à pouvoir étendre le vocabulaire ».

Dans le cadre d'utilisation des ontologies pour l'agrégation textuelle on trouve par exemple la fonction d'agrégation AVG-KW. Cette fonction se base sur une ontologie légère : cette dernière n'est qu'une ontologie dont les relations entre les termes est du type "est-un ".

La fonction prend en entrée un ensemble de mots-clefs et génère un nouvel ensemble de mots-clefs agrégés. Le processus d'agrégation se base sur le calcul de distance (nombre d'arc entre termes) entre chaque paire de mots-clés présents dans l'ontologie, cette distance sert à trouver le plus petit ancêtre commun (LCA) entre les paires des mots. L'un des problèmes qui ont été reconnue à ce niveau est quand les mots sont très éloignée le LCA retournée est la racine. Alors et afin d'éviter ce problème les auteurs dans [29] ont défini une distance maximale (Dmax) autorisée lors de l'agrégation des termes (cette distance est généralement comprise entre 3 et 5).

Cette fonction en plus d'utiliser l'ontologie qui est une connaissance externe et doit être défini à l'avance, elle présente un autre inconvénient : lors de l'agrégation cette fonction risque une perte de sens qui doit être toujours contrôlé [29]

2.3 Techniques basées sur les statistiques :

Ces approches sont simples et ne nécessitent pas de connaissances préalables sur le domaine, elles sont basées sur des informations statistiques pour attribuer un score de qualité aux mots clefs extraits.

Les méthodes proposées pour l'approche statistique permettent de regrouper tous les concepts dans des classes dont la corrélation entre les termes est définie par des critères. Plus le critère de corrélation est important, plus il y a de chance de trouver des liens sémantiques entre les membres de la classe.

Parmi les méthodes qui ont été proposé on cite : la méthode Topic dans [28], la fonction Top-Keyword dans [14], la méthode GOTA [22], la méthode basée sur les motifs fréquents [7].

2.3.1 La méthode Topic :

Une autre approche appelée Topic a été proposée par Christian et al [28] dans laquelle ils ont considéré que chaque sujet (Topic) est représenté par un ensemble de mots-clés. Leur technique consiste d'abord à créer une matrice de corrélations dans laquelle les mesures de distance entre les différents mots sont sauvegardées.

Ensuite les deux éléments qui ont la plus grande distance sont affectés comme étant les centres des deux classes. L'assignation des éléments aux centres est basée sur le principe de la classification défini dans k-means. Cet algorithme est expliqué en détails dans [13]. Nous reprenons le même processus d'éclatement à chacune des deux classes. L'éclatement dépend du

seuil spécifié. L'algorithme trouve donc un arbre binaire des classes et chaque classe représente un topic.

2.3.2 La méthode Top-Keyword :

Dans [14] les auteurs ont proposé une nouvelle fonction d'agrégation de mots clefs qu'ils ont appelé Top-Keyword. Elle consiste à extraire les K mots les plus représentatifs d'un corpus de documents. Pour cela, une notion de représentativité est définie, il s'agit d'une mesure calculée pour chaque terme du corpus. Ils ont utilisé la mesure TF-IDF issue du domaine de la recherche d'information pour l'évaluation de la représentativité d'un terme.

2.3.3 La méthode GOTA :

Une autre approche appelée GOTA (Google distance for Olap Textual Aggregation)[22] le principe de cette approche est d'utiliser l'algorithme k-means pour extraire K class. Ce dernier utilise la distance de similarité de Google comme une mesure de distance pour calculer la distance sémantique entre les différents mots clés obtenus, l'algorithme K-means prend les mots les plus représentative (le mot le plus fréquent dans chaque classe).Les mots obtenue ils seront considérés comme des agrégats qui représentent le corpus.

2.3.4 La méthode basée sur les motifs fréquents :

Une autre méthode basée sur les motifs fréquents proposée dans [7]. Le principe de cette approche et d'utiliser les algorithmes Apriori et close pour extraire les motifs fréquents, puis de calculer le poids des motifs fréquents qui ont été trouvés par les algorithmes précédents. Les termes ayant la plus grande valeur de poids sont considérés comme des agrégats.

2.4 Techniques basées sur les graphes :

Dans [22,] l'auteur propose une nouvelle fonction TAG (Textual Aggregation by Graph) pour l'agrégation textuelle. Cette approche consiste à extraire à partir d'un ensemble de termes ceux qui sont les plus représentatifs du corpus en utilisant un graphe. L'algorithme prend en entrée tous les termes extraits de corpus. Pour extraire les termes fréquents cette approche passe par 3 principales étapes la première étape est l'extraction des mots-clefs avec leurs fréquences à partir d'un corpus la deuxième étape consiste à construire la matrice d'affinité, les éléments de cette matrice correspondent aux degrés d'affinité entre les mots-clés. A partir de cette matrice on

construit le graphe d'affinité. La troisième étape est de trouver les circuits qui représentent les agrégats des mots-clés les plus représentatifs du corpus.

Dans ce chapitre, nous avons présenté un état de l'art qui résume les principaux travaux qui ont été menés autour de l'agrégation des données textuelles.

Nous avons vu des approches d'extraction des mots clés les plus représentatifs à partir d'une collection de documents ou corpus dans une perspective d'agrégation.

Pour notre travail, nous avons choisi les règles d'association comme une technique d'extraction des mots clé qui représente le corpus, cette technique a été classée dans la catégorie basée sur les statistiques. Le principe de cette technique et quelque notation sont présentés dans le chapitre suivant.

Chapitre 3

L’agrégation textuelle basée sur les Règle d’association

Dans ce chapitre nous allons présenter notre technique qui est basée sur l'extraction des règles d'association pour l'agrégation textuelle. Les règles d'association ont été étudiées en analyse de données, puisen fouille de données afin de trouver des régularités, des corrélations dans des bases de données de grandes tailles. Les règles d'association comme méthode de fouille de données ont un aspect séduisant par l'apparente facilité à interpréter les résultats.

3.1 Les règles d'association :

a) Définition d'une Règle d'association :

Soit A et B deux sous-ensembles d'éléments qui appartiennent à un ensemble M. On appelle A→B la règle d'association entre A et B.

L'évaluation de cette relation se base principalement sur le calcul de deux paramètres qu'on appelle le support et la confiance.

b) Définition d'une Transaction :

On considère un ensemble $E = \{E_1, \dots E_n\}$ de n éléments (items) distincts. On appelle une transaction T le sous ensemble E' inclus dans$E$. Dans une base de données$D$, chaque transaction est identifiée par une clé unique.

c) Définition d'un Itemset : Un Itemset désigne un ensemble d'objets ou d'articles.

d) Définition d'un K-Itemset : Un K-Itemset est un Itemset de k Items.

e) Définition d'un Support d'un Itemset :

Soit A un Itemset de n éléments. Dans une base de données transactionnelle D , le support de A est le nombre de transactions dans D incluant A divisé par le nombre total des transactions

$$Support\ (A) = \frac{\{t \in D\ /\ A \sqsubseteq t\}}{|D|} \qquad (3.1)$$

f) Itemset Fréquent :

Un Itemset A est fréquent si et seulement si son support est supérieur à un support minimum défini par l'utilisateur.

g) Support d'une règle d'association :

Dans une base de données D, le support d'une règle d'association $A \Rightarrow$ B est le nombre de transactions qui contiennent A et B divisé par le nombre total des transactions :

$$Support\ (\mathrm{A} \Rightarrow \mathrm{B}) = \frac{\{t \in D\ /\ (A \cup B) \sqsubseteq t\}}{|D|} \tag{3.2}$$

h) Confiance :

La confiance d'une règle d'association $A \Rightarrow$ B est le rapport entre le nombre de transactions de D contenant Au B , et le nombre de transactions de D contenant A .C'est-à-dire que:

$$Confiance\ (\mathrm{A} \Rightarrow \mathrm{B}) = \frac{\{t \in D\ /\ (A \cup B) \sqsubseteq t\}}{\{t \in D\ /\ A \sqsubseteq t\}} \tag{3.3}$$

Si la confiance est supérieure ou égale à un seuil donné σ_c et son support supérieur ou égal au seuil σ_c, la règle est dite **valide**. Si la confiance est de 1, la règle est dite **exacte** (c'est une implication au sens logique), sinon, elle est **approximative**.

3.2 Avantages et inconvénients des règles d'association :

3.2.1 Avantage :

Les règles d'association entent plusieurs avantages parmi lesquels on peut citer par exemple :

1. Leur application dans plusieurs domaines de la vie quotidienne, comme l'analyse du panier de la ménagère.
2. La découverte de connaissances utiles, cachées dans les grandes bases des données.

3. Leur simplicité, efficacité et facilité de compréhension.
4. Leur formalisme non supervisé et général.
5. Leurs résultats clairs et faciles à interpréter [2].

3.2.2 Inconvénients :

Malgré les grands avantages que les règles d'association peuvent représenter, elles ont aussi des faiblesses qu'on peut résumer dans :

1. Le temps énorme consacré à la recherche des ltemSets fréquents.
2. La grande quantité des règles d'association générées.
3. La difficulté d'évaluer la qualité des règles d'associations par des indices statiques ou par l'expert du domaine.
4. La production des règles triviales et inutiles qui n'apportent pas de nouvelles informations. [4] [8]

3.3 Les étapes d'extraction des règles d'association :

Le processus d'extraction des règles d'association se déroule en quatre étapes comme définie dans [2] :

3.3.1 Préparation des données :

Cette étape consiste à réduire la quantité des données en gardant seulement celles les plus pertinentes, et en transformant par la suite, ces derniers en un contexte d'extraction, c'est-à-dire une transformation en un triplé constitué : d'un ensemble d'objets, d'un ensemble d'ltemsets ainsi qu'une relation binaire entre les deux. Cette transformation des données en données binaires permettra d'améliorer la qualité des règles d'association.

3.3.2 Recherche des ItemSets fréquents :

Un Itemset fréquent est un ensemble d'éléments dont le support est supérieur ou égal à un certain support minimal spécifié par l'utilisateur. Cette étape est très coûteuse en temps d'exécution. Pour un ensemble de n items par exemple, le nombre d'ltemsets fréquents qui peut être générés est de 2^n.

3.3.3 Production des règles d'association :

La génération des règles d'association consiste à déterminer les règles d'association dont le support et la confiance sont supérieurs ou égaux à un certain support et confiance minimaux définis par l'utilisateur.

3.3.4 L'élagage des règles d'association :

Tout élément de connaissance doit être validé par des experts mais leur temps et leur énergie sont comptés, il est donc très important de pouvoir concentrer leur interaction sur les points essentiels.

Une des solutions pour réduire le très grand nombre de règles extraites consiste à les trier ou les filtrer pour ne garder que les règles les plus "intéressantes". Le support et la confiance sont deux mesures exploitées par les algorithmes d'extraction de règles d'association, notamment, pour en réduire le nombre, mais elles ne sont pas suffisantes pour qualifier l'intérêt de ces règles aux yeux d'un expert.

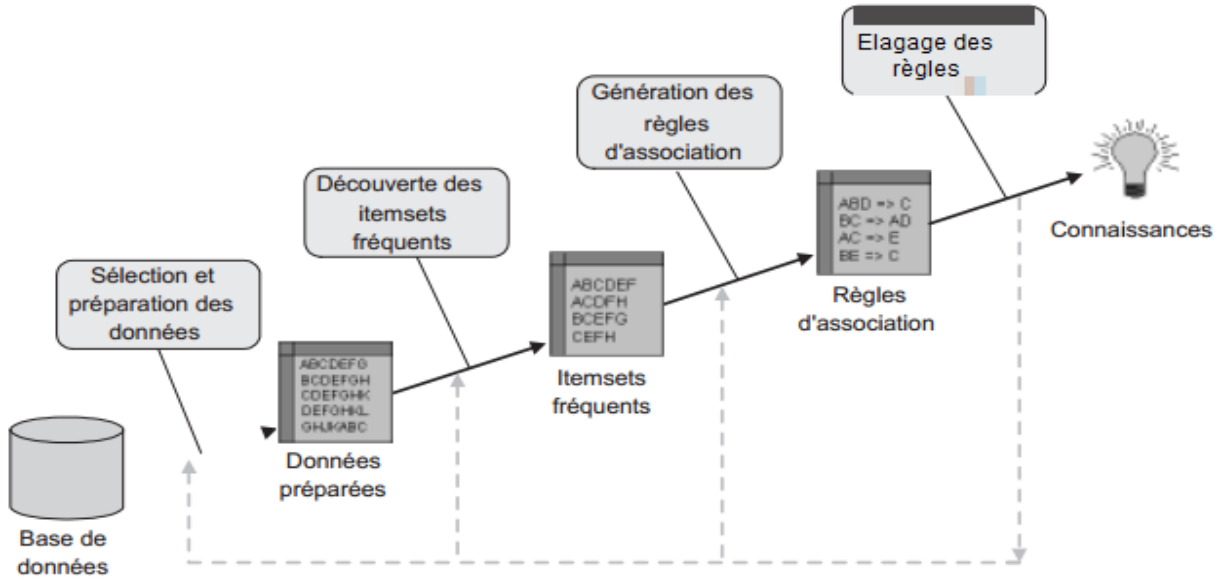


Figure 3.1 : Étapes du processus d'extraction de règles d'association [25]

3.4 Les algorithmes de recherche des règles d'association :

Il existe plusieurs algorithmes de recherche de règles d'association, qui permettent de trouver et d'extraire des règles d'association dans des grandes bases de données.[2], on cité quelque algorithme et on focalise sur l'algorithme Apriori qui est le plus connu.

Parmi les algorithmes de recherche des règles d'association on trouveFP-growth (frequent pattern growth) [20] qui utilise une structure d'arbre (FP-tree) pour stocker une forme compressée d'une base de données, autre algorithme appelée l'algorithme OneR (one attribute-rule) consiste à trouver un attribut à utiliser qui fait le moins d'erreurs de prédiction possibles. Il existe aussi d'autres algorithmes de recherche des règles d'association comme Eclat [21], procedure ASSOC de GUHA [1].

3.4.1 Algorithme Apriori :

L'algorithme Apriori (Agrawal et Srikant, 1994) représente la base de tous les algorithmes de recherche des règles d'association. Il utilise une stratégie de recherche des ItemSets fréquents en commençant par les ItemSets les plus généraux vers les plus spécifiques [12].

Cet algorithme se base principalement sur les deux règles suivantes Soit S un Itemset et $S' \subseteq S$ alors

- Si *S* est non fréquent, alors les ItemSets qu'on construit de *S* sont aussi nonfréquents.
- Si *S* est fréquent, alors *S'*est aussi fréquent.

Le déroulement de l'algorithme Apriori peut être décomposé en cinq étapes :

1. Recherche de tous les nouveaux candidats.
2. Pour chaque candidat trouvé, on calcule son support.
3. Évaluation du support calculé par l'algorithme par rapport au support minimum défini par l'utilisateur.
4. On supprime les candidats dont le support est inférieur au support minimum.
5. Construction de toutes les règles ayant une confiance supérieure à Confiance minimumdéfini par l'utilisateur.

[1]http://www.cs.cas.cz/coufal/guha/

3.5 Outil d'extraction des règles d'association :

Les outils de fouille de données sont des programmes spécialisés dans l'analyse et l'extraction des connaissances à partir des données informatisées. Il existe des nombreux outils de statistiques et de Data mining. Il y a ceux qui sont gratuits et d'autre sont payants. Ces outilssont largement suffisants pour la plupart des études.

La situation est un peu plus difficile si nous souhaitons traiter de grandes bases de données avec des milliers d'items.

Parmi les logiciels libres, on a trouvé : KNIME, Weka, Tanagra, RapidMiner, Orange. Ces outils permettent d'exécuter différentes tâches de fouille de donnée comme l'extraction des règles d'association, la classification. Ces outils donne des résultats très efficaces et plus précis.

Figure 3.2 : différents outils destiné au Data Mining

3.6 Application des règles d'association a l'agrégation textuelle :

On cherche des relations entre les mots d'un corpus sous forme de règles : si les mots $x_1, \ldots, x_m$ apparaissent dans un **texte** alors les mots $x_m + 1, \ldots, x_n$ apparaissent aussi dans le texte que l'on formalise en :

$$x_1 \cdots x_m \rightarrow x_m + 1 \cdots x_n$$

3.6.1 Les règles d'association et l'agrégation :

Notre technique est basée sur l'extraction des agrégats d'un texte on utilise le concept des règle d'association de data mining. Ce dernier consiste à extraire les mots les plus associées entre eux (la probabilité d'apparition d'une collection des mots ensemble).

Voici un petit exemple qui explique le déroulement de notre travail :
On considère les 5 documents suivants comme des transactions où chaque transaction a des items qui sont les mots clé de chaque document :

1. Algorithm, network, graph, multicast, processor, system, parallel
2. Cluster, network, design, message, processor, system, framework
3. Algorithm, software, graph, method, session, analysis, parallel
4. Switch, load, design, power, path, system, timing
5. Cable, load, energy, power, current, motor, signal
6. Protocol, system, server, connection, processor, network, router

- On fixe le support à la valeur 0.5 et la confiance à 1 et on a trouvé ces résultats à l'aide d'un outil open source Orange :

Les règles d'association	***Confiance***	***supporte***
Processor system ==> network	1	3/6
Processor ==> network	1	3/6
Network==> system	1	3/6
Processor ==> system	1	3/6

Tableau 3.1 : résultants obtenu par l'outil Orange

Dans notre exemple on trouve que la première règle est la plus représentative dans ce cas, car elle regroupe les autre règles. Donc l'agrégat de cet ensemble des documents est formé par les termes de la première règle **(Processor, system, network).**

3.7 Mesures de performances :

Nous avons choisis comme mesure de performance le Rappel, la Précision, ainsi que le rapport entre ces deux mesutres caractérisé par la F-mesure. Ces mesures sont destinées a mesurer la pertinence des systèmes de recherche d'information ”RI” face à une requête d'utilisateur et une masse de données très grande.

3.7.1 Le rappel :

Ou "racall" en anglais, en générale il est défini par le nombre de documents pertinents retrouvés par rapport au nombre de documents pertinent. Selon [1], Le rappel mesure la capacité du système à restituer l'ensemble des documents pertinents (en lien avec le silence documentaire) plus le résultat correspond au besoin de l'utilisateur plus le rappel est élevé, il est déni par la formule 3.4 :

$$Rappel = \frac{nbr\ documents\ pertinents\ retrouves}{nbr\ de\ documents\ pertinents} \qquad (3.4)$$

Dans notre contexte, le rappel correspond au nombre des documents dans lesquelles tous les termes du résultat d'agrégation sont présents par rapport au nombre de documents total, nous l'avons calculé à l'aide de la formule3.5.

$$Rappel = \frac{Nbr\ de\ document\ contenant\ \text{les}\ termes\ de\ l\prime agregation\acute{}}{Nbr\ documents\ total} \qquad (3.5)$$

3.7.2 La précision :

Cette mesure correspond au nombre de documents pertinents retrouvés par rapport au nombre de documents total, selon [1], Précision moyenne non interpolée par rapport à l'ensemble des documents pertinents. Elle mesure la capacité du système à ne restituer que des documents pertinents (en lien avec le bruit documentaire).

Dans notre contexte, elle correspond à la somme des nombres des documents contenant le terme ti sur le nombre de document total par, ce rapport a été calculé à l'aide de la formule 3.6 :

$$précision = \frac{\sum \frac{nbr\ doc\ contenant\ term\ 1}{nbr\ doc\ total} + .. + \frac{nbr\ doc\ contenant\ term\ 2}{nbr\ doc\ total}}{Nbr\ de\ temre\ de\ l'agregation} \quad (3.6)$$

3.7.3 La F-Mesure :

Une mesure qui combine entre le rappel et la précision en effectuant un rapport entre ces deux, plusieurs variantes de cette mesure ont été proposées, la plus simple consiste à diviser le rappel par la précision comme le montre la formule 3.7

Une autre variante très populaire est dénie par la formule 3.8

$$F\_Mesure = \frac{2 * (Rappel * Precision)}{Rappel + Precision} \quad (3.8)$$

3.8 Autres mesure de qualité des règles :

Soient $D(B), D(H)$ et $(D\ (B \cup H) = D(B) \cap D(H))$ les ensemble de textes de D possèdent respectivement tous les termes de $B, H\ et\ B \cup H$ (voir figure 3.3), trois valeur de probabilités ont un impact sur la valeur des mesures que nous utilisons .il s'agit de :

$P\ (B), P(H)\ et\ P(B,H)$ *Qui* se définissent par la formule générale suivant :

- $P(x) = \frac{|\mathcal{D}(X)|}{|\mathcal{D}|}$Comprit entre 0 et 1.
- *P (B, H)*est le supporte de la règle La probabilitéconditionnelle
- *P (H/B)* $= \frac{P(B,H)}{P\ (B)}$en est la confiance.

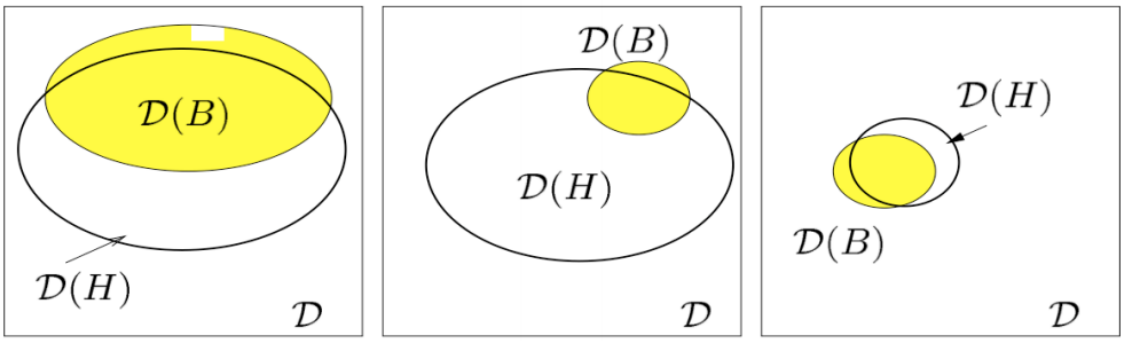


Figure 3.3 Différentes situations illustrant une règle H –→ B [36]

Nous présentons d'autres indices, dont une synthèse se trouve dans [24] qui permettent différents classements des règles.

3.8.1 L'intérêt :

L'intérêt (ou lift) mesure la déviation du support de la règle par rapport au cas d'indépendance. Rappelons que pour deux événements indépendants B et$H$, $P(H|B) = P(H)$ et donc $P(B \cup H) = P(B) * P(H)$. La valeur de l'intérêt est donnée par la formule 3.9 :

$$intérèt\ [B \longrightarrow H] = \frac{P(B \cup H)}{P(B) \times P(H)} \qquad (3.9)$$

L'intérêt varie dans l'intervalle [0, +∞ [.

3.8.2La conviction :

La conviction est similaire à l'intérêt mais appliqué aux contre-exemples. Dans notre contexte, un contre-exemple correspond au motif $B \cup \neg H$ tel que $\neg H$ signifie l'absence d'au moins un terme du motif H. $/D\ (\neg H)| = |D| - |D(H)| et\ P\ (\neg H) = 1 - P(H)$.

$$conviction[B \longrightarrow H] = \frac{P(B) \times P(\neg H)}{P(B \cup \neg H)} \qquad (3.10)$$

La conviction vaut donc $\frac{1}{intérèt\ [B \dashrightarrow \neg H]}$, elle varie dans l'intervalle [0,+∞[et n'est pas symétrique.

3.8.3 La dépendance :

La dépendance est utilisée pour mesurer une distance de la confiance de la règle parrapport au cas d'indépendance de $B\ et\ H$,elle est définie par la formule (3.11).

$$Dependance\ [B \dashrightarrow H] = |P(H|B) - P(H)| \tag{3.11}$$

Cette mesure varie dans l'intervalle [0,1[.

3.8.4 La nouveauté et la satisfaction :

La Nouveauté [24] est définie par la formule (3.12) :

$$nouveauté\ [B \dashrightarrow H] = P(B \cap H) - P(B) \times P(H) \tag{3.12}$$

Nouveauté est une mesure de l'écart à l'indépendance entre la prémisse et le conséquent de la règle. Elle est symétriquedonc non implicative. Elle dépend de la taille dedonnés.Elle prend ses valeurs sur l'intervalle [-1,1].

La nouveauté est symétrique alors que larègle $B \rightarrow H$ peut avoir plus de contre-exemples que la règle $B \rightarrow H$. Pour cette raison, [Yannick Toussaint 2012] introduire la satisfaction :

$$satisfaction\ [B \dashrightarrow H] = \frac{\big(P(\neg H) - P(\neg H|B)\big)}{P(\neg H)} \tag{3.12}$$

Dans ce chapitre, nous avons vu les principes de fonctionnement de notre approche(les règles d'association) ainsi que les adaptations que nous avons apporté pour les utiliser dans notre contexte. Nous avons également présenté un exemple illustrant l'avantage de cette technique.

Dans le chapitre suivant nous allons présenter notre processus expérimental pour comparer nos résultats avec ceux obtenu avec Apriori.

Chapitre 4

Expérimentation

L'objectif de ce chapitre est de présenter les différentes expérimentations effectuées pour évaluer les résultats des deux approches (règles d'association, motifs fréquents). On commence par citer les étapes d'implémentation.

4.1 Environnement expérimental :

Nous avons implémenté les deux approches d'agrégation en JAVA. il s'agit d'un langage de programmation orienté objet gratuit et portable. Ce qui lui a permis d'être parmi les langages les plus utilisé. Il a été développé par la firme Sun Microsystems en 1995. Cette dernière a été rachetée en 2009 par Oracle. Le JDK (Java Development Kit) et le JRE (Java Runtime Environnent) peuvent être gratuitement téléchargés sur le site officiel.

Nous avons utilisé la version JDK1.8.0-91 dans un ordinateur portable doté d'un processeur Intel ® CORE™ I3 @ 1.9 GHz (4 CPUs). Avec une mémoire vive (RAM) de 4GO qui fonctionne sur le système d'exploitation Microsoft Windows 10 professionnel 64 bits.

Le programme est écrit dans l'éditeur de code NetBeans IDE 7.0 . C'est un éditeur parmi les IDE (Integrated Development Environment) Java. Il simplifie grandement l'édition et la gestion d'un programme. Ils intègrent les fonctionnalités suivantes :

- Editeur de textes avec mise en couleur des mots clés Java, des commentaires.
- Complétion automatique (menus contextuels proposant la liste des méthodes d'un objet).
- Génération automatique des dossiers nécessaires à l'organisation d'un programme et des paquetages des classes.
- Intégration des commandes Java et de leurs options dans des menus et des boîtes de dialogue appropriés.
- Débogueur pour corriger les erreurs.

4.2 Déroulement de travail :

Afin de comparer les deux approches d'agrégation, il est nécessaire d'appliquer les deux approches sur les mêmes données et dans le même contexte. Pour cela nous avons choisi d'utiliser les mêmes valeurs de paramètres pour le support.

La figure 4.1 représente l'organigramme global présentant les différentes étapes d'implémentation.

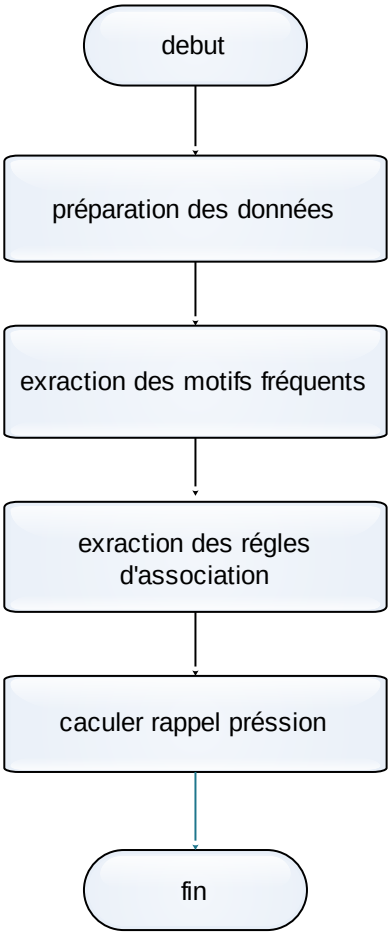


Figure 4.1 : Organigramme globale de déroulement du travail

4.2.1 Etape 1 : Préparation des données

C'est une étape très importante. il s'agit d'une opération de mise en forme des données dans laquelle le programme lit les données à partir du fichier texte pour ensuite les traduire dans

un fichier de type CSV pour l'utiliser dans un outil open source appelée Orange dédié à la fouille de données et à l'apprentissage automatique (voir Annexe 1).

L'outil Orange peut importer différents formats de fichier. Mais il propose surtout un format propriétaire CSV. Donc on a adapté notre corpus a se format. Nous avant implémenté cette tâche à l'aide de l'environnement JAVA Netbeans.

4.2.2 Etape 2 : Construire les motifs fréquents et les règles d'association avec l'outil ORANGE

Après l'étape de préparation de donnée. Le fichier CSV obtenue est utilisé comme fichier d'entrée dans l'outil Orange.

Par défaut. Toutes les observations et toutes les variables sont sélectionnées dans ORANGE. Il nous faut donc placer le composant ASSOCIATION RULES et FREQUENT ITEM SETS de l'onglet ASSOCIATE.

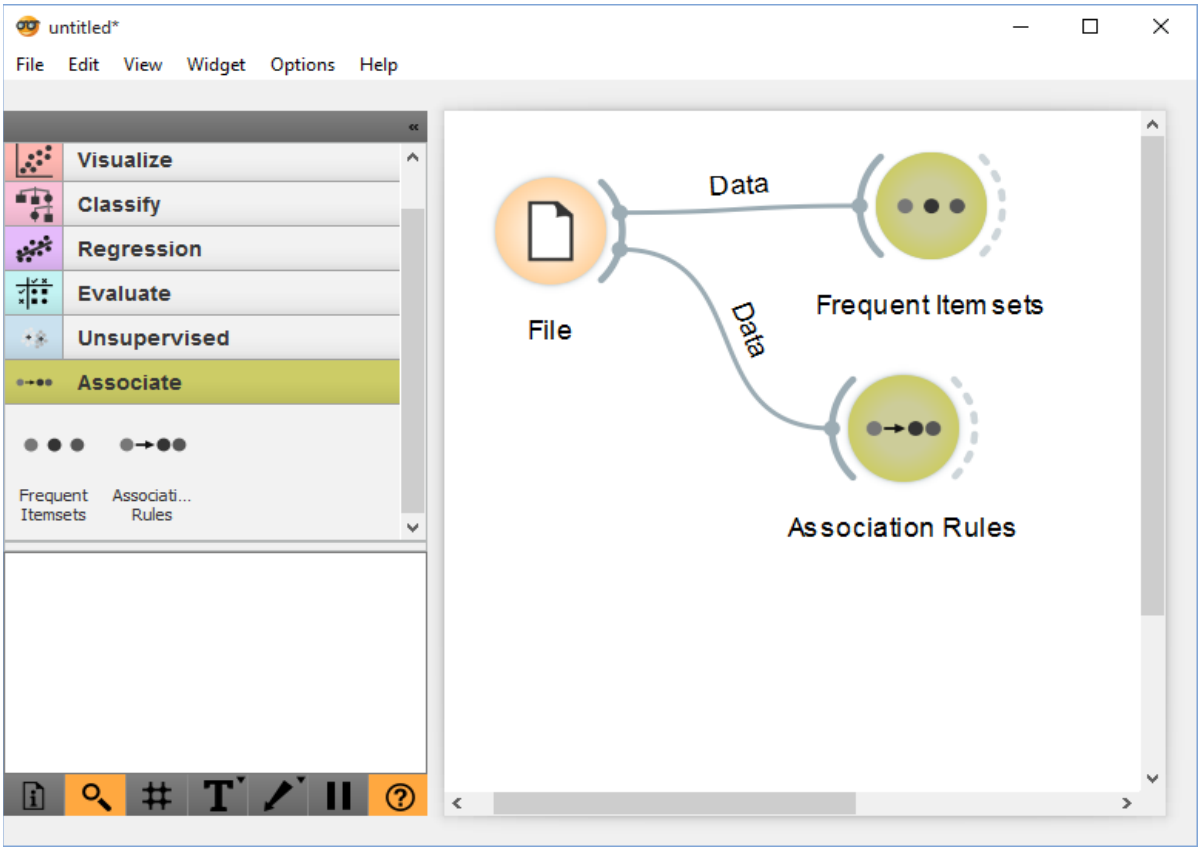


Figure 4. 2 : capture d'écran workflow de l'outil orange

L'icône ASSOCIATION RULE permet d'ouvrir l'interface de définir les valeurs du support et de la confiance. Les paramètres sont très explicites dans ORANGE.

Les calculs sont automatiquement exécutés lorsque nous relions le composant FILE aux composants ASSOCIATION RULE et FREQUENT ITEM SETS.

Pour voir les règles. Nous cliquons sur le menu OPEN du composant ASSOCIATION RULES dans le Workflow voire Annexe 3.

On sauvegarde les résultants dans un fichier text pour l'utilisée dans l'étape suivant.

4.2.3 Etape 3 : choisir le meilleur agrégat de chaque approche

Cette étape c'est l'étape très important dans notre travail. Elle consiste à sélectionnée le meilleur agrégat pour les deux approche. Pour l'approche fréquente ItemSets. Parmi les ItemSets qui ont un support au plus égal à 0.4. On sélectionne les dix ItemSets ayant les supports les plus élevés. Puis on calcule le poids de chaque Itemset. Les ItemSets ayant le poids le plus élevé sont choisie comme agrégat.

L'approche que nous présentons dans ce travail consiste à trier par support les règles qui sont produites par l'outil Orang. Les ItemSets de la règle ayant le maximum de confiance sont choisis comme agrégat.

4.2.4 Etape 4 : calcule le rappel et précision

Cette étape consiste à calculée le rappel et la précision de chaque technique. Nous avons implémenté ces fonctions dans l'environnement java sur netbeans.

4.3 Résultat de l'expérimentation :

Pour assurer une comparaison efficace entre les deux techniques, nous avons testé sur un ensemble de données textuelles correspondant à des articles publiés dans la conférence « Innovation 2011 » qui s'est déroulée à Dubaï en 2011.

Ces articles tournent autour des innovations de recherches dans le domaine des technologies de l'information. Tous les articles publiés dans cette conférence sont en langue anglaise. Ils traitent en général les domaines suivants :

- l'informatique et les Systèmes embarqués.
- les systèmes intelligents.
- communication et network.
- sécurité de l'information.
- les applications Internet et les services Web.
- technologies d'apprentissage et de l'éducation.

Nous avons utilisé dans notre travail 81 articles avec 110 termes de cette conférence comme étant un corpus d'expérimentation.

Dans cette section on va présenter les différents résultats pour quatre expériences. La première expérience pour la comparaison entre la technique de sélection des agrégats utilisée dans [7] et notre technique. Les deux techniques sont basées sur Apriori dans la recherche des motifs fréquents et différent dans la manière de sélectionner les agrégats.

Dans notre approche nous avons trié les itemsets obtenues en fonction de leurs supports et nous avons sélectionné les 10 premiers pour lesquels on a calculé leur poids. L'itemset avec le plus grand poids parmi les 10 premiers est choisi comme agrégat. Par contre dans l'approche utilisée dans le travail de [7]. Ils ont calculé le poids de tous les itemsets et ont sélectionné celui avec le plus poids.

Dans les trois autres on utilise différentes valeurs de confiance (0.5. 0.7 et 1) pour une valeur fixe du support de 0.4 et nous comparons les résultats obtenus.

Pour la première expérience on a utilisé les mêmes données et les mêmes paramètres que dans [7]. Les résultats obtenus sont exposés dans Tableau 4.1. Les valeurs de F-mesure pour les résultats obtenus par [7] ont été recalculées parce que celles reportées dans le mémoire n'étaient pas correctes.

	Apriori (Notre Approche)			***Apriori de[7]***		
K	***Rappel***	***Précision***	***F-mesure***	***Rappel***	***Précision***	***F-mesure***
3	80.246	92.181	85.8	55 .56	67 .08	60 .78
4	71.60	91.66	80.44	45 .67	71 .91	55 .86
5	91.358	91.358	91.358	70 .37	72 .10	71 .22
6	87.654	90.534	89.071	76 .54	67 .48	71 .73
7	86.419	91.005	88.653	79 .01	60 .84	68 .74
8	82.716	90.432	86.402	65 .43	61 .88	63 .61
9	70.37	89.574	78.819	64 .19	58 .57	61 .25

Tableau 4.1 : les mesures de performance d'Apriori (Notre Approche) et Apriori de [7]

On remarque que nos résultats sont meilleurs que les résultats obtenus dans [7]. Le meilleur rappel obtenu par notre approche est de 91.35% et correspond à k=5. Par contre le meilleur rappel obtenu dans [7] est de 79.01% et correspond à k=7.Le meilleur précision obtenu par notre approche est de 92.18%et correspond à k=3. Par contre le meilleur rappel obtenu dans [7] est de 72.10% et correspond à k=5.Le meilleur F-mesure obtenu par notre approche est de 91.36% et correspond à k=5. Par contre le meilleur F-mesure obtenu dans [7] est de 71.73% et correspond à k=7.

Etant donné que notre approche basée sur les motifs fréquents obtenus par Apriori nous a donnée de meilleurs résultats par rapport à celle utilisée dans [7]. Nous l'utilisons pour la comparer avec notre deuxième approche basée sur les règles d'association correspondantes. On fixe le support à la valeur 0.4 et on a utilisé différentes valeurs la confiance : 0.5. 0.7 Et 1. Ces expériences sont effectuées pour des valeurs de k (le nombre des agrégats) variant de 3 à 9.

Le tableau 4.2 résume les résultats obtenu pour un support de 0.4 et une confiance de 0.5.

	Apriori			***R_Association***		
K	***Rappel***	***Précision***	***F-mesure***	***Rappel***	***Précision***	***F-mesure***
3	80.24	92.18	85.80	82.71	93.41	93.41
4	71.60	91.66	80.44	75.30	92.59	83.06
5	91.35	91.35	91.35	91.35	90.61	90.98
6	87.65	90.53	89.07	88.88	90.94	89.90
7	86.41	91.00	88.65	85.18	89.41	87.25
8	82.71	90.43	86.40	81.48	89.81	85.44
9	70.37	89.57	78.81	65.43	88.47	75.22

Tableau 4.2 : les mesures de performance d'Apriori et règle d'association

Le tableau 4.3 représente les agrégats obtenus par les deux approches pour différentes valeurs de k :

K	***Apriori***	***R_Association***
3	number .system . .data	number . system. result
4	number case data system	number . system . result . data
5	number .case .time .data .system	Application . result . number . data . system
6	system .information .case .time . number .data	system . result . case . time . number . data
7	system .result .information .case .time .number .data	System .result . work . case . time . number . data
8	information .result .case .application .time .number .data .system	order . data . result . approach . case .time . number . system
9	information .application .result .order .case .time .number .data .system	problem .approach . result . order .case . time .number . data .system

Tableau 4.3 les agrégats pour les deux approches

Nous avons représenté graphiquement nos différents résultats. La figure 4.3 représente le rappel en fonction de K. La figure 4.4 représente la précision par en fonction de K. La figure 4.5 représente le f_mesure en fonction de K.

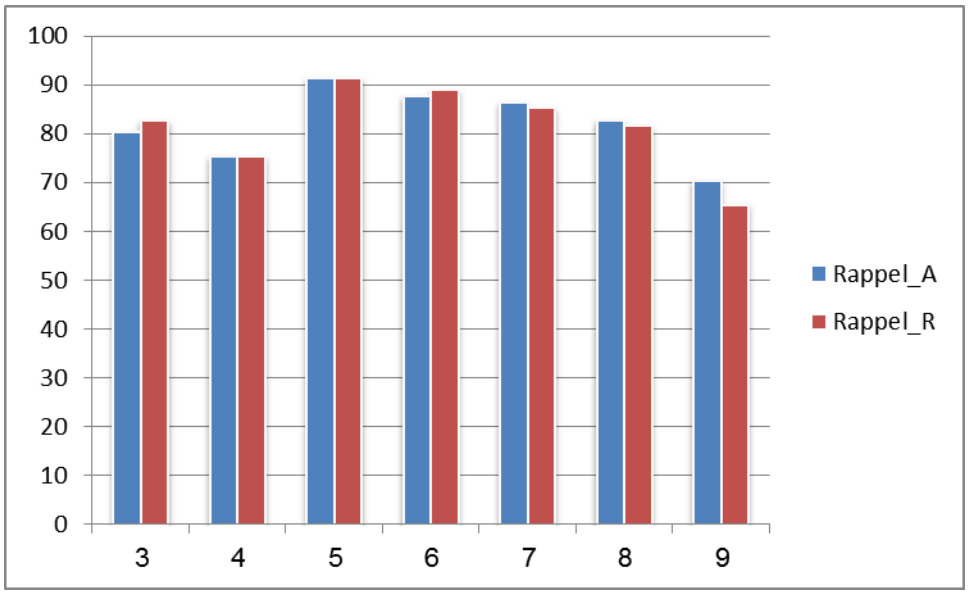


Figure 4.3 représentations graphiques du comparaissant de rappel

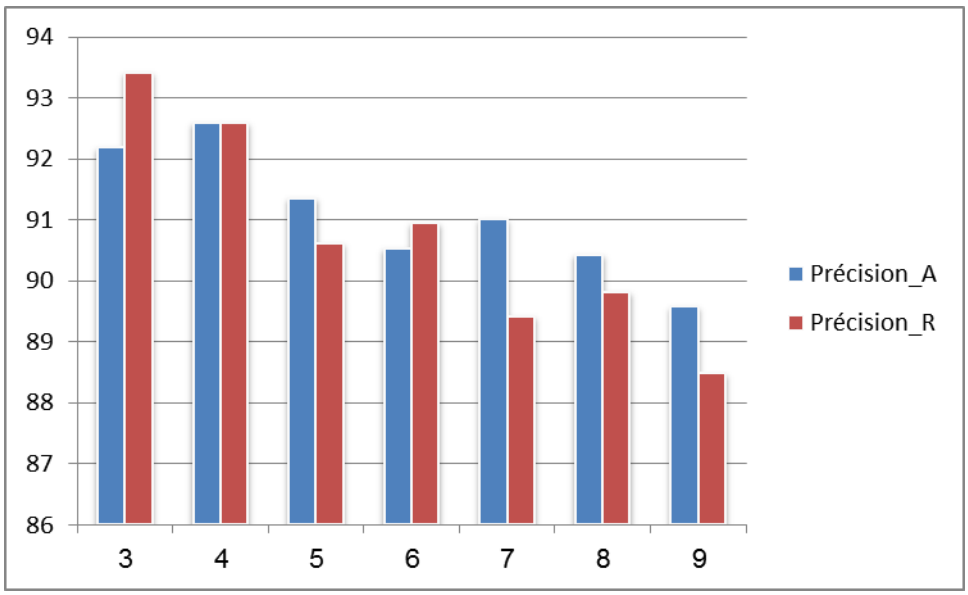


Figure 4.4 représentation graphique du comparaissant de précision

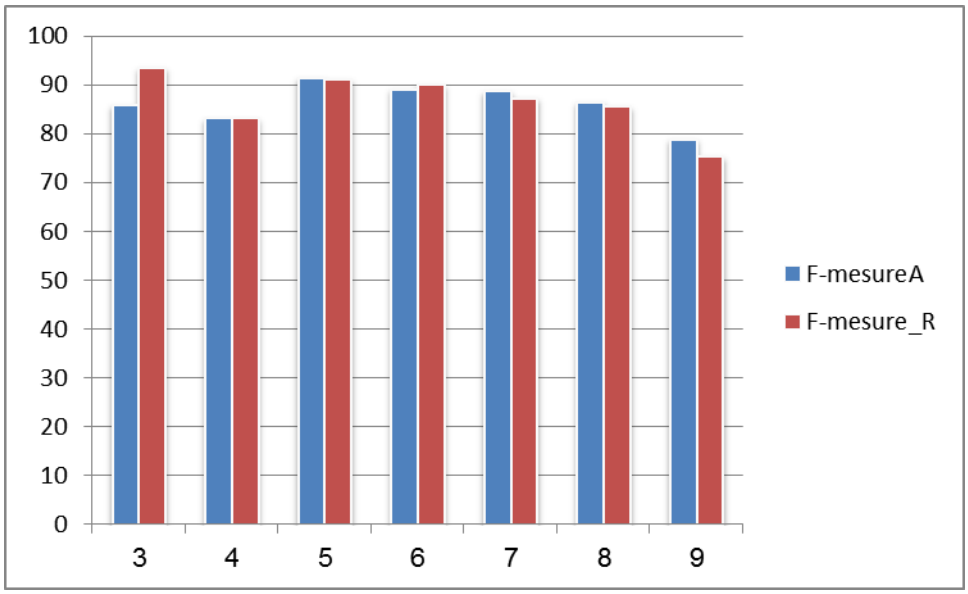


Figure 4.5 représentations graphiques du comparaissant de F_meseure

Dans la deuxième expérience les valeurs de confiance et support sont 0.7 et 0.4.

Le tableau 4.4 résume les différentes mesures que nous avons effectuées pour évaluer les performances des deux approches Apriori et règle d'association.

	Apriori			***R_Association***		
K	***Rappel***	***Précision***	***F-mesure***	***Rappel***	***Précision***	***F-mesure***
3	80.24	92.18	85.80	82.71	93.41	93.41
4	71.6	91.66	80.40	75.30	92.59	83.06
5	91.35	91.35	91.35	91.35	90.61	90.98
6	87.65	90.53	89.07	88.88	90.94	89.90
7	86.41	91.00	88.65	85.18	89.41	87.25
8	82.71	90.43	86.40	81.48	89.81	85.44
9	70.37	89.57	78.81	65.43	88.47	75.22

Tableau 4.4 : les mesures de performance de fréquent Itemset et règle d'association

On remarque que la technique basée sur règles d'association donne de meilleurs résultats de rappel que la technique basée sur les motifs fréquents dans le cas des valeurs de k

inférieur à 7. Pour la précision la technique des règles d'association, reste la meilleure pour des valeurs de k inférieur à 5. Par contre la valeur de F_mesure est meilleure dans le cas des valeurs de k égal à 3, 4 et 6.

Le tableau 4.5 représente les agrégats de deux techniques.

K	***Apriori***	***R_Association***
3	number system data	Number System result
4	number case data system	Number system result data
5	case data number system information	case data number system information
6	application system result data time number	computer data system result number
7	computer data system result case number system	result computer application information number data system
8	application time information system result case number data	data result approach case time number order system
9	Order time work data system case number information computer	order time work data system case number information computer

Tableau 4.5 : les agrégats pour les deux approches

Nous avons représenté graphiquement nos résultats. La figure 4.6 représente le rappel en fonction de K. La figure 4.7 représente la précision en fonction de k. La figure 4.8 représente F_mesure en fonction de k.

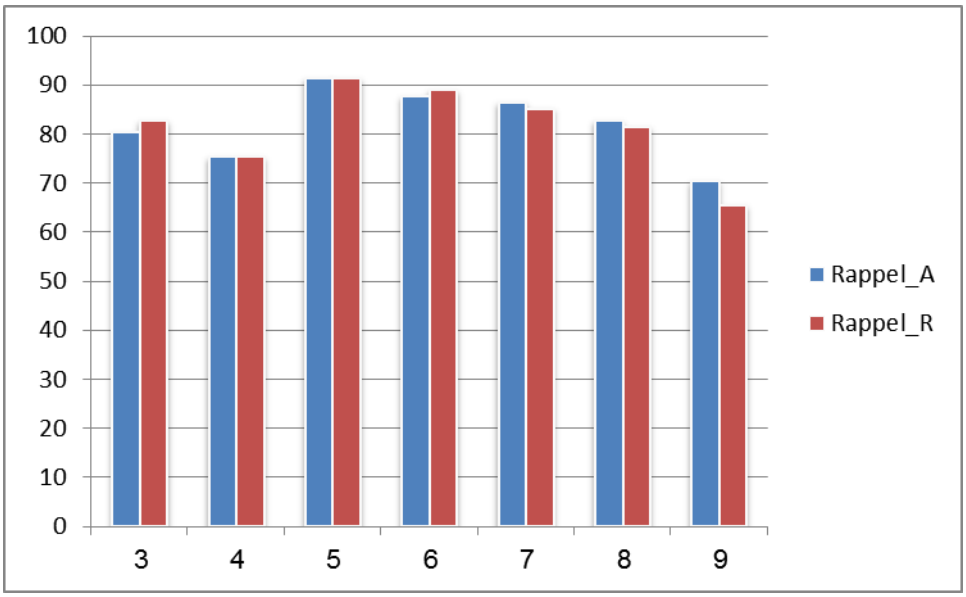


Figure 4.6 représentations graphiques du comparaissant de Rappel

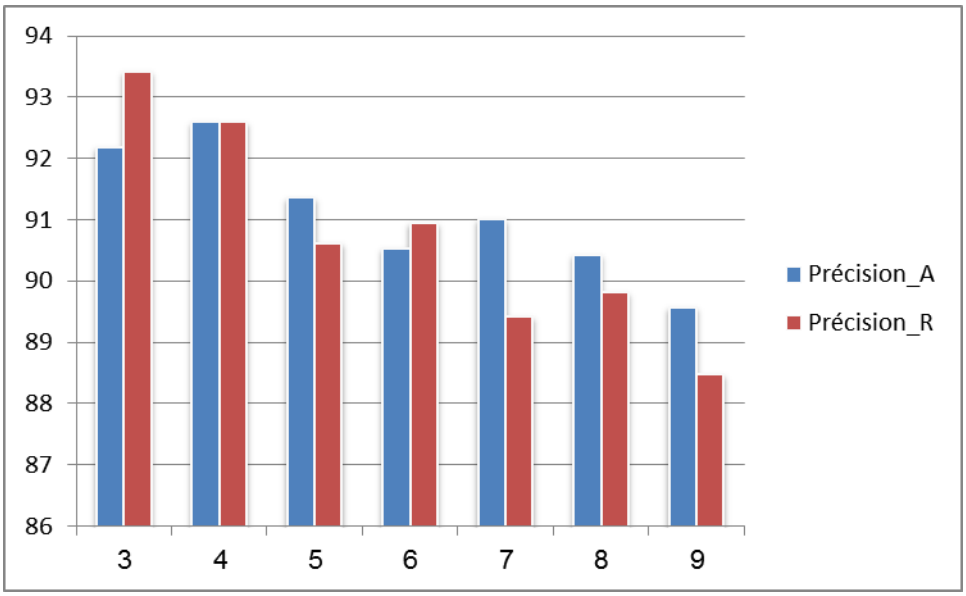


Figure 4.7 : représentations graphiques du comparaissant de la précision

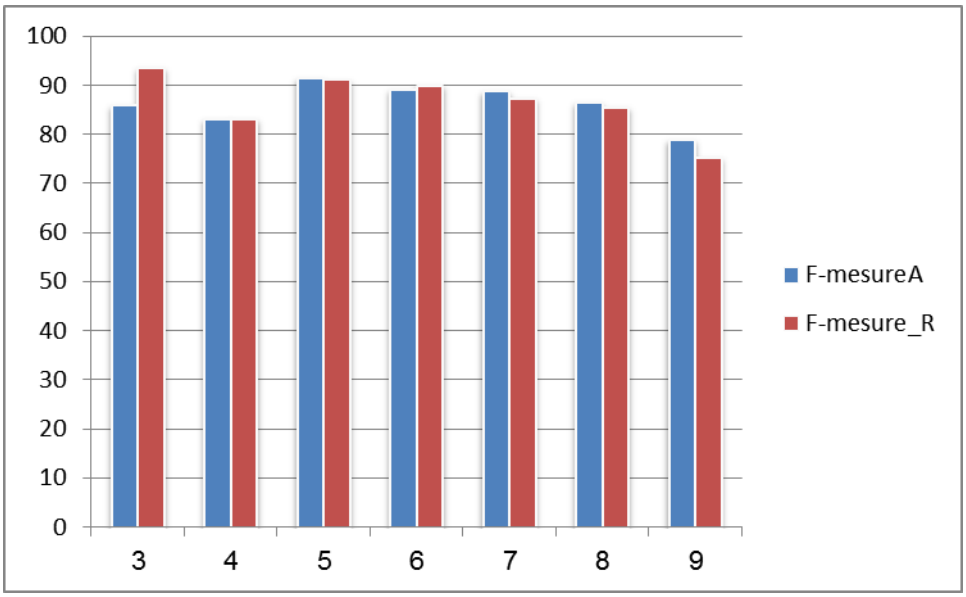


Figure 4. 8 : représentations graphiques du comparaissant du f_mesure

Dans la troisième expérience les valeurs de Confiance et support sont 1, 0.4 respectivement.

Le tableau 4.6 résume les différentes mesures que nous avons effectuées pour évaluer les performances des deux approches Apriori et règle d'association.

	Apriori			***R_Association***		
k	***Rappel***	***Précision***	***F-mesure***	***Rappel***	***Précision***	***F-mesure***
3	80.24	92.18	85.80	69.13	87.65	77.30
4	71.60	91.66	80.40	69.13	90.43	78.36
5	91.35	91.35	91.35	92.59	91.11	91.84
6	87.65	90.53	89.07	88.88	90.94	89.90
7	86.41	91.00	88.65	82.71	90.47	86.42
8	82.71	90.43	86.40	75.30	89.96	81.98
9	70.37	89.57	78.81	65.43	87.37	74.82

Tableau 4.6 : les mesures de performance de fréquent Itemset et règle d'association

Dans cette expérience on remarque que généralement la technique basée sue les motifs fréquents donne de meilleurs résultats que la technique basée sur les règles d'association dans le cas de toutes les mesures.

Le tableau 4.7 représente les agrégats de deux techniques.

K	***Apriori***	***R_Association***
3	case method system	case value system
4	number case data system	case work number system
5	case work data number system	result case work number system
6	data case order number system	Result case work time number system
7	information order case time number data system	Result order case time number data system
8	result order case number time information data system	result order case time number information data system
9	result application order information case time number data system	result approach order problem case time number data system

Tableau 4.7 : les agrégats pour les deux approches

Comme les expériences précèdent nous avons représenté graphiquement nos résultats. La figure 4.9 représente le rappel en fonction de K. La figure 4.10 représente la précision en fonction de K. La figure 4.11 représente F_mesure en fonction de K.

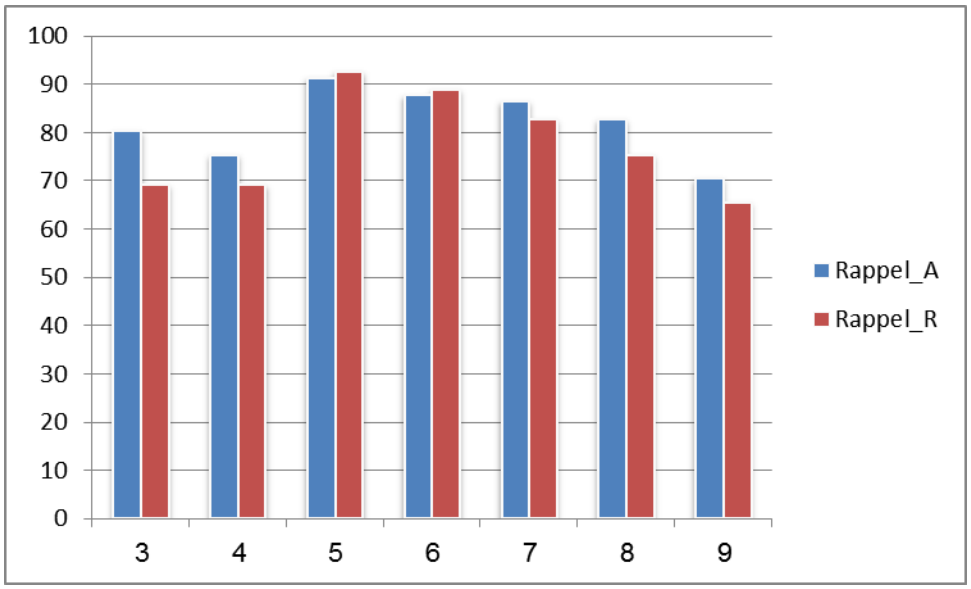


Figure 4.9 : représentations graphiques du comparaissant du rappel

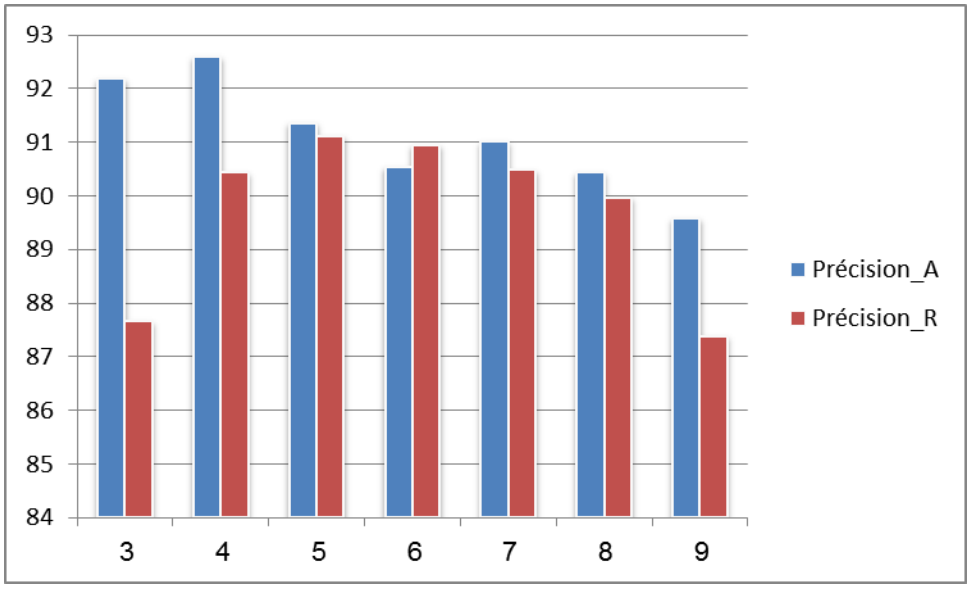


Figure 4.10 : représentations graphiques de comparaissant de la précision

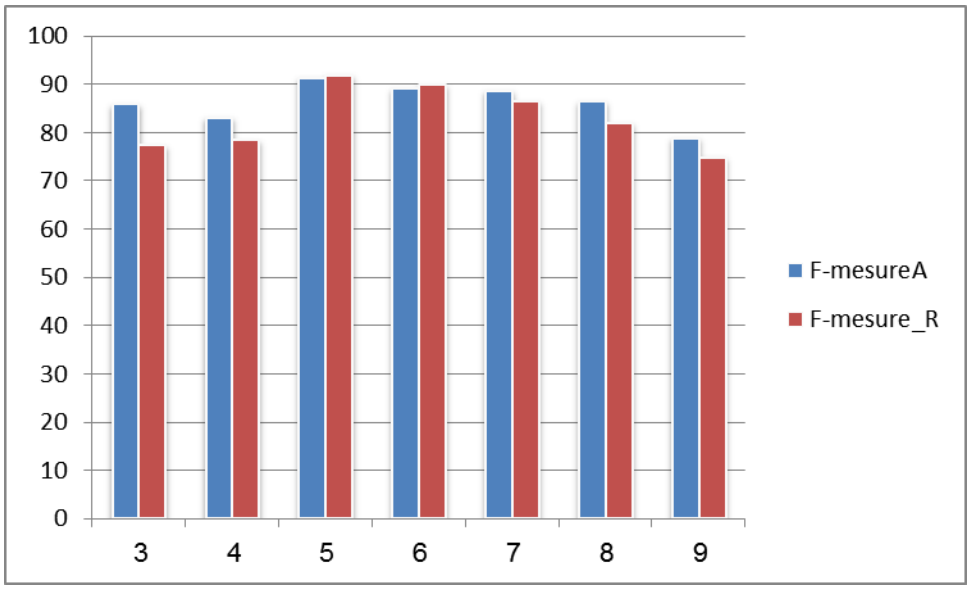


Figure 4.11 représentations graphiques du comparaissant du f_mesure

Enfin nous pouvons dire que la performance de la technique basé des règles d'association donne de bons résultats par rapport à la technique basée sue les motifs fréquents dans les cas où les valeurs de k sont inferieures a 5 pour toutes les mesures. Pour cela elle reste toujours une des méthodes acceptable dans le domaine d'agrégation textuelle. Car tous les résultats de cette technique donne des bonnes valeurs (généralement supérieur à 65 %) pour toutes les mesure de performance (rappel. précision. F_mesure).

Conclusion

Conclusion

Dans le domaine de text mining il existe plusieurs méthodes pour l'agrégation textuelle tel que (Top keyword, Topic, Biencube, GOTA, et TAG. Le but de notre travail est de tester le concept des règles d'association et leur effet dans l'agrégation textuelle.

Pour atteindre les objectifs tracés, on a choisi la méthode Apriori pour l'extraction des motifs fréquents, et nous avons utilisé déférents outils orienté au domaine de fouille de données comme l'outil ORANGE et WEKA pour l'extraction des règles d'association et nous avons implémenté les parties préparation des données, extraction des agrégats et la partie du calcul des mesure des performances en langage JAVA.

On a Aussi choisi une fonction d'agrégation textuelle basée sur les motifs fréquents pour effectuer une étude comparative, On a appliqué les deux fonctions sur un corpus de données textuelles réelles. Il s'agit d'un ensemble d'articles scientifiques de la conférence Innovations organisée à Dubaï en 2011, utilisé dans [22][7].

Après analyse des résultats nous sommes arrivé à la conclusion que les fonctions basées sur les motifs fréquents et celle basée sur les règles d'association donne des résultats presque similaire, mais la technique basée sur les règles d'association donne de meilleurs résultats dans le cas des valeurs de k inferieures a 5 pour les différentes expériences que nous avons effectuées. Pour cela nous pouvons dire que cette technique reste valide comme une fonction d'agrégation textuelle efficace.

Enfin ce travail a été une expérience enrichissante et très utile, nous avons découvert plusieurs aspects pour une meilleure programmation avec java, et pour une meilleure utilisation des outils de fouille de donnée, en plus nous avons acquis des nouvelles connaissances dans le domaine de l'analyse des données textuelles.

Bibliographie

Bibliographie:

[1] A. Baccini, S. Déjean, and J. Mothe. Analyse des critères d'évaluation des systèmes de recherche d'information. Technique et Science Informatiques, 29(3), 2010.

[2] Abdelali, M. and O. Hicham . Création de règles d'association. Caen, Ensicaen 2003.

[3] Abdelhamid DJEFFAL : Cours Fouille de données avancée 2015.

[4] Aby, K. and A. EL Kourri ,Post traitement de règles d'association. Caen, ISMRA ENSI Caen 2003.

[5] Agrawal, R. and Srikant, R. (1994) Fast Algorithms for Mining Association Rules in Large Databases. Proceedings of the 20th International Conference on Very Large Data Bases, Santiago de Chile, 12-15 September 1994 .

[6] Amini Massih-Reza, Gaussier Eric : Recherche d'information, Applications, modèles et algorithmes, Eyrolles, 2013.

[7] Ben Abderrahmane Nadjat L'utilisation de technique data mining pour l'agrégation textuelle, mémoire de master Université de Laghouat 2015.

[8] Ben Yahia, S. and E. Mephu Nguifo . Approches d'extraction de règles d'association basées sur la correspondance de Galois. Lens, Centre de Recherche en Informatique de Lens 2004.

[9] Benzian Khadidja and Cherif Asma : ÉTUDE COMPARATIVE ENTRE DEUX FONCTIONS D'AGREGATION DES DONNEES TEXTUELLES, mémoire de master Université de Laghouat 2013.

[10] C. Poudat and G. Cleuziou. Catégorisation de textes en domaines et genres. Complémentarité des indexations lexicale et morphosyntaxique, 1998.

[11] D. Bourigault and N. Aussenac-Gilles. La syntaxe comme marche pied de l'acquisition des connaissances : Bilan critique d'une expérience. Université Paul Sabatier, 2003.

[12] Diop, C. T., M. Lo, et al. . Intégration de règles d' association pour améliorer la recherche d' informations XML. Quatrième conférence francophone en Recherche d'Information et Applications. École Nationale Supérieure des Mines de Saint Étienne 2007.

[13] F. Archetti and E. Fersini. A hierarchical document clustering environment based on the induced k-means. 1998.

[14] F. Ravat and O. Teste. Top-keyword : agrégation de mots-clefs dans un environnement d'analyse en ligne(olap). Toulouse CEDEX 9 (France),2008.

[15] Fatiha Boubekeur. Contribution à la définition de modèles de recherche d'information flexibles basés sur les CP-Nets. PhD thèses ,Université Paul Sabatier-Toulouse III, 2008.

[16] Habert B and Nazarenko A. La syntaxe comme marche pied de l'acquisition des connaissances : Bilan critique d'une expérience. Actes des septièmes Journées Acquisition des Connaissances, 1996.

[17] Hacène Cherfi. Etude et réalisation d'un système d'extraction de connaissance à partir de textes. Thèse de Doctorat, INRIA, France, 2004.

[18] Harrathi Farah. Extraction de concepts et de relations entre concepts à partir des documents multilingues : Approche statistique et ontologique. Thèse de Doctorat, L'Institut Nationale des Sciences Appliquées de Lyon, 2009.

[19] Ibekwe-SanJuan Fidelia. : Fouille de textes : méthodes, outils et applications Hermès, 2007.

[20] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation. Data Mining and Knowledge Discovery , 2004.

[21] Mohammed J. Zaki. Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering 12(3):372-390, May/June 2000.

[22] Mustapha Bouakkaz Aggrégation sémantique OLAP , Thèse de Doctorat Université de Laghouat 2017 .

[23] N. BECHET. Extraction et regroupement de descripteurs morphosyntaxiques pour des processus de fouille de textes. Thèse de Doctorat d'Université Montpellier II, 2009.

[24] N. Lavrac, P. Flach, and B Zupan. Rule evaluation measures: A unifying view. In Proceedings of the 9th International Workshop on Inductive Logic Programming Slovenia, 1999. Springer Verlag.

[25] Nicolas Pasquier. Data Mining : algorithmes d'extraction et de réduction des règles d'association dans les bases de données, Universit´e Blaise Pascal - ClermontFerrand II, 2000.

[26] Preux Philippe : Fouille de données (notes de cours) 2011.

[27] R. Abbes and J. Dichy. Extraction automatique de fréquences lexicales en arabe et analyse d'un corpus journalistique avec le logiciel araconc et la base de connaissances diinar.1. Presses Universitaires de Lyon, 2008.

[28] R. Brussee and C. Wartena. Topic detection by clustering keywords.19th International Conference on Database and Expert Systems Applications, 2008.

[29] R. Tournier. Analyse en ligne (olap) de documents. Thèse de doctorat d'Université Toulouse III- Paul Sabatier (France), 2007.

[30] R., Vinot. Classification automatique de textes dans des catégories non thématiques, Thèse de Doctorat de l'école nationale supérieure des télécommunications, France, 2004.

[31] Robert Neches, Richard E Fikes, Tim Finin, Thomas Gruber, RameshPatil, Ted Senator, and William R Swartout. Enabling technology for knowledge sharing. AI magazine, 12(3) :36, 1991.

[32] S. Bringay, A. Laurent, P. Poncelet, M. Roche, and M. Teisseire. Bien cube, les données textuelles peuvent s'agréger ! Univ. Montpellier 3, 2010.

[33] S. Réhel, Catégorisation automatique de textes et cooccurrence de mots provenant de documents non étiquetés, Mémoire de maitrise de l'université Laval, Quebec, Canada, 2005.

[34] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. AI magazine, 17(3) :37, 1996.

[35] Vincent Claveau. Vectorisation, okapi et calcul de similarité pour le Tal : pour oublier enfin le tf-idf. In TALN-Traitement Automatique des Langues Naturelles, 2012.

[36] Yannick Toussaint. Fouille de textes : des méthodes symboliques pour la construction d'ontologies et l'annotation sémantique guidée par les connaissances, Traitement du texte et du document. Université Henri Poincaré - Nancy I, 2011.

Annexe

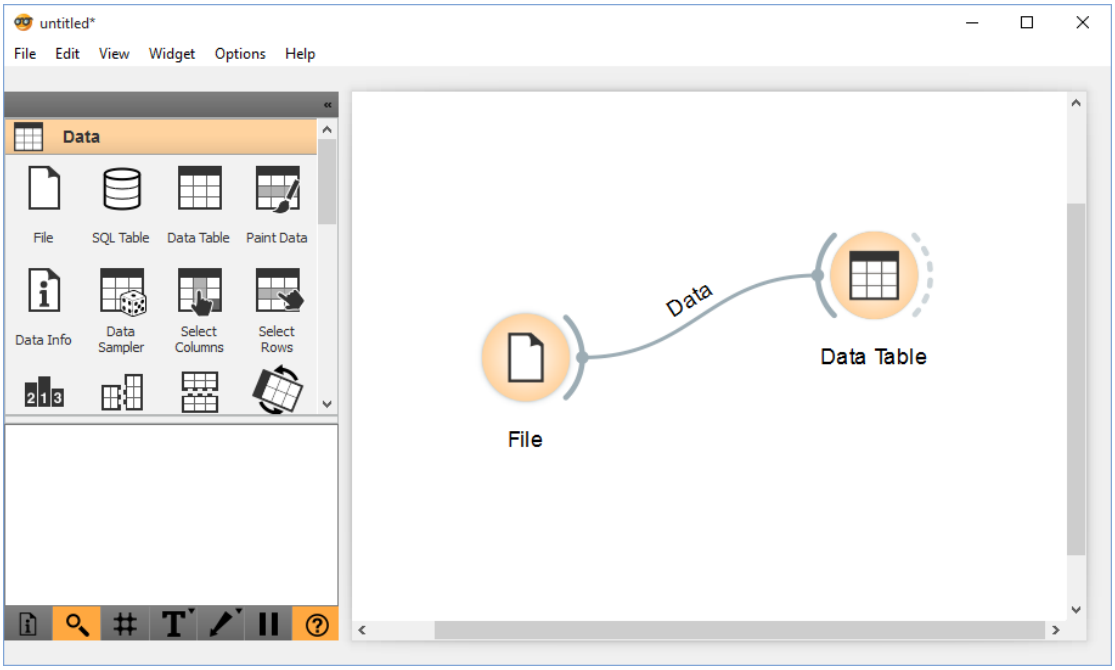
Annexe 1 : Présentation de l'outil Orange :

Orange est un logiciel libre de fouille de données (data mining) , Il a été créé en 1997 par Janez Demšar et Blaž Zupan à l'université de Ljubljana en Slovénie. Il est développé en C++ et en Python, Chaque algorithme se présente sous la forme de widgets pouvant avoir une entrée et une sortie ; ils sont agencés dans une fenêtre. Il existe des versions Windows, Mac et Linux.

Orange propose donc une interface graphique qui repose sur le concept de « widgets ». Chacun de ces widgets accepte un ou plusieurs types d'entrées et génère un ou plusieurs types de sorties.

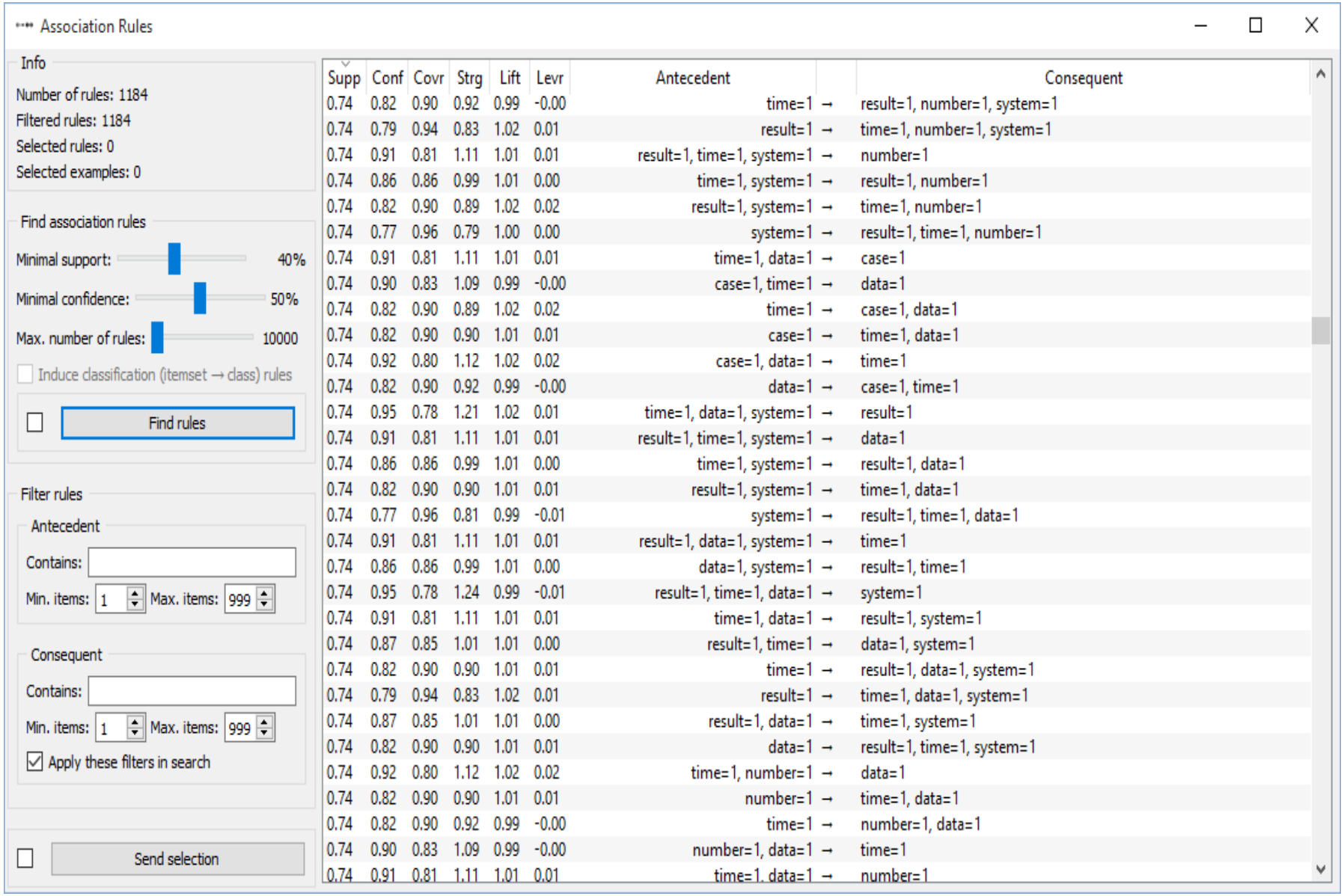
L'interface permet de façon très simple de créer ces widgets et de les connecter entre eux. A condition bien évidemment que leurs entrées/sorties soient compatibles. On construit ainsi graphiquement un enchaînement des traitements (autrement dit, un workflow) à réaliser.

Pour la définition des données il suffit soit de « glisser / déposer » l'icône du widget FILE depuis la barre située en gauche de fenêtre,



L'interface graphique de l'outil Orange

Annexe 2 : Les règles obtenues par l'outil Orange :



Supp	Conf	Covr	Strg	Lift	Levr	Antecedent		Consequent
0.74	0.82	0.90	0.92	0.99	-0.00	time=1	→	result=1, number=1, system=1
0.74	0.79	0.94	0.83	1.02	0.01	result=1	→	time=1, number=1, system=1
0.74	0.91	0.81	1.11	1.01	0.01	result=1, time=1, system=1	→	number=1
0.74	0.86	0.86	0.99	1.01	0.00	time=1, system=1	→	result=1, number=1
0.74	0.82	0.90	0.89	1.02	0.02	result=1, system=1	→	time=1, number=1
0.74	0.77	0.96	0.79	1.00	0.00	system=1	→	result=1, time=1, number=1
0.74	0.91	0.81	1.11	1.01	0.01	time=1, data=1	→	case=1
0.74	0.90	0.83	1.09	0.99	-0.00	case=1, time=1	→	data=1
0.74	0.82	0.90	0.89	1.02	0.02	time=1	→	case=1, data=1
0.74	0.82	0.90	0.90	1.01	0.01	case=1	→	time=1, data=1
0.74	0.92	0.80	1.12	1.02	0.02	case=1, data=1	→	time=1
0.74	0.82	0.90	0.92	0.99	-0.00	data=1	→	case=1, time=1
0.74	0.95	0.78	1.21	1.02	0.01	time=1, data=1, system=1	→	result=1
0.74	0.91	0.81	1.11	1.01	0.01	result=1, time=1, system=1	→	data=1
0.74	0.86	0.86	0.99	1.01	0.00	time=1, system=1	→	result=1, data=1
0.74	0.82	0.90	0.90	1.01	0.01	result=1, system=1	→	time=1, data=1
0.74	0.77	0.96	0.81	0.99	-0.01	system=1	→	result=1, time=1, data=1
0.74	0.91	0.81	1.11	1.01	0.01	result=1, data=1, system=1	→	time=1
0.74	0.86	0.86	0.99	1.01	0.00	data=1, system=1	→	result=1, time=1
0.74	0.95	0.78	1.24	0.99	-0.01	result=1, time=1, data=1	→	system=1
0.74	0.91	0.81	1.11	1.01	0.01	time=1, data=1	→	result=1, system=1
0.74	0.87	0.85	1.01	1.01	0.00	result=1, time=1	→	data=1, system=1
0.74	0.82	0.90	0.90	1.01	0.01	time=1	→	result=1, data=1, system=1
0.74	0.79	0.94	0.83	1.02	0.01	result=1	→	time=1, data=1, system=1
0.74	0.87	0.85	1.01	1.01	0.00	result=1, data=1	→	time=1, system=1
0.74	0.82	0.90	0.90	1.01	0.01	data=1	→	result=1, time=1, system=1
0.74	0.92	0.80	1.12	1.02	0.02	time=1, number=1	→	data=1
0.74	0.82	0.90	0.90	1.01	0.01	number=1	→	time=1, data=1
0.74	0.82	0.90	0.92	0.99	-0.00	time=1	→	number=1, data=1
0.74	0.90	0.83	1.09	0.99	-0.00	number=1, data=1	→	time=1
0.74	0.91	0.81	1.11	1.01	0.01	time=1, data=1	→	number=1

Interface des règles obtenue par Orange