REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU

FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

MEMOIRE DE MAGISTER

En **Informatique**
Option : **Ingénierie des Systèmes Informatiques**

Présenté par

M^{lle} SEGHIRI Narimane

Sujet

Détection et Extraction d'Informations Temporelles dans des Entrepôts de Données.

Devant le jury d'examen :

Mr	AHMED OUAMER Rachid	Professeur	U.M.M.T.O	Président.
Mr	RASSOUL Idir	M.C.A	U.M.M.T.O	Rapporteur.
Mme	AMIROUCHE Fatiha	M.C.A	U.M.M.T.O	Examinatrice.
Mme	AOUGHLIS Farida	M.C.A	U.M.M.T.O	Examinatrice.

Année Universitaire : 2013 / 2014.

Table des matières

Liste des tableaux

liste des figures

Résumé

Les entrepôts de données représentent une collection de données. Ces données sont issues de sources différentes, c'est pourquoi elles doivent être intégrées et elles sont représentées dans un même sujet. Le fonctionnement d'un entrepôt de données consiste en une extraction des données sources, puis ces données extraites sont stockées au niveau de l'entrepôt après leur l'intégration. Plusieurs tâches peuvent être effectuées sur les entrepôts de données comme synthèse d'informations spécialisées. La tâche d'extraction de données peut se faire selon différents critères , comme le type de donnée par exemple. L'extraction de données temporelles dans un entrepôt de données, prend en compte le critère de type de données temporelles. Dans le schéma de fonctionnement d'un entrepôt de données, les sources peuvent avoir des documents contenant des informations temporelles. Cependant, lors de l'extraction des données à partir des sources, les informations temporelles peuvent ne pas être prises en compte par la tâche d'extraction, et l'entrepôt de données résultat contiendra des données du types différents, et peut ne pas contenir le type de données souhaité. Pour faire usage de ces informations, des outils temporels sont appliqués aux documents afin d'extraire les données temporelles contenues dans les documents, puis ces données temporelles extraites doivent être regroupées pour pouvoir être exploitées. Dans ce présent travail, on propose d'abord une synthèse sur les entrepôts de données ainsi que sur la détection et l'extraction des informations temporelles. Ensuite une architecture d'un système d'entrepôt de données intégrant un dictionnaire d'annotation de données temporelles proposée. Après, pour vérifier la faisabilité de nos propositions, un prototype d'outil de détection et d'extraction d'informations temporelles dans les entrepôts de données a été développé dans un environnement java intégré où l'accès basé sur les informations temporelles en utilisant le dictionnaire d'annotation que nous proposons de construire.

Mots-Clés : Dictionnaire de données temporelles, Informations temporelles, Entrepôts de données, Base de données, Recherche d'informations, multi modélisation, Reconnaissance et Extraction d'informations Temporelles.

Abstract

Data warehouses represent a collection of data. These data are from different sources. They must be integrated and are represented in the same subject. The operation of a data warehouse consists of an extraction from the source data. Then the extracted data is stored at the warehouse after their integration. Several tasks can be performed on data warehousing as a synthesis of specialized information. The data extraction task can be done by various criteria such as the type of data, for example; Extraction of temporal data in a data warehouse takes into account the criterion of time data type. In the scheme of operation of a data warehouse, the sources may have documents with time information. However, when extracting data from source, time information can't be considered by the extraction task and the result data warehouse will contain different types of data, and may not contain the desired type of data. To make use of this information, time tools are applied documents to extract the time data contained in the documents and the extracted time data should be grouped to be exploited. In this present work, first proposes a synthesis of data warehouses as well as on the detection and extraction of temporal information. Then an architecture of a data warehouse system that integrates a dictionary annotation proposed temporal data. Next, to test the feasibility of our proposals, a prototype tool for detecting and extracting time information in the data warehouse was developed in an integrated Java environment where access based on the time information using the Dictionary annotation we propose to build.

Keywords: Dictionary of temporal data, temporal data, Data warehousing, database, information retrieval, multi modeling, Recognition and Extraction of Temporal Information.

Introduction générale

Contexte et problématique

Les entrepôts de données représentent une collection de données. Ces données sont issues de sources différentes, c'est pourquoi elles doivent être intégrées, et elles sont représentées dans un même sujet. Le fonctionnement d'un entrepôt de données consiste en une extraction de données sources, puis ces données extraites sont stockées au niveau de l'entrepôt après l'intégration de ces données.

Plusieurs tâches peuvent être effectuées sur les entrepôts de données comme l'extraction des données de l'entrepôt. Cependant, la tâche d'extraction peut se faire selon différents critères , comme le type de donnée, la syntaxe, la sémantique...L'extraction de données temporelles dans un entrepôt de données, prend en compte le critère de type de données temporelles.

Le temps joue un rôle central dans tout l'espace de l'information, et il a été pris en compte dans plusieurs domaines tels que l'extraction de l'information, les questions-réponses. Le temps et les mesures temporelles peuvent aider à recréer une période historique particulière ou décrire le contexte chronologique d'un document ou d'une collection de documents [Ling et al., 2010]. Le temps peut être utile pour explorer les résultats de la recherche sur les échéanciers bien définis et de multiples niveaux de granularité de temps en raison des caractéristiques clés de l'information temporelle :

- L'information temporelle est bien définie, en effet si on suppose deux points dans le temps ou deux intervalles, la relation entre eux peut être identifiée, par exemple, la relation peut être de type avant, chevauchement, ou après.

- L'information temporelle peut être normalisée, car quelles que soient les modalités ou la langue utilisée, chaque expression temporelle se référant à la même sémantique utilisée peut être normalisée à la même valeur dans un format standard. Cette propriété rend le terme information temporelle et la langue indépendante.

- L'information temporelle peut être organisée de façon hiérarchique, les expressions temporelles peuvent être de différents niveaux de granularité, par exemple, de type jour ("11 juin 2015") ou de type année ("2015"). En raison du fait que l'année se composent de mois, et les mois se composent de semaines et les semaines se composent de jour.

Grâce à ses caractéristiques clés, les informations temporelles contenues dans les documents peuvent être utilisées pour la recherche d'informations spécifiques au temps et des applications d'exploration. L'information temporelle qui est associée à un document est sa date de création ou la date de sa dernière modification. Ce type d'information, qui est directement accessible par le biais des métadonnées d'un document, peut déjà être utilisé pour plusieurs tâches telles que la recherche ou le regroupement dans le temps.

Dans le schéma de fonctionnement d'un entrepôt de données, les sources peuvent contenir des documents contenant des informations temporelles. Cependant, lors de l'extraction des données à partir des sources, les informations temporelles peuvent ne pas être prises en compte par la tâche d'extraction, et l'entrepôt de données issu contiendra des données du types différents, et peut ne pas contenir le type de données souhaité. Pour faire usage de ces informations, des outils temporels sont appliqués aux documents afin d'extraire les données temporelles contenues dans les documents, puis ces données temporelles extraites doivent être regroupées pour pouvoir être exploitées.

C'est dans ce contexte que s'inscrit notre travail et en se basant sur le fonctionnement d'un entrepôt de données, nous proposons un système d'un entrepôt de données intégrant le dictionnaire d'annotation temporelle.

Organisation du mémoire

Ce mémoire est organisé en trois chapitres :

- Le chapitre I décrit un état de l'art sur les entrepôts de données . Nous commençons par définir l'entrepôt de données en section I.2. Les principaux traitements en section I.3. Définition des données dans l'entrepôt en section I.4. Métadonnées seront présentés en section I.5. La section I.6 est consacrée à l'extraction de données. Entrepôt de données vs base de données en section I.7. Construction d'un entrepôt de données est présentée en section I.8. Les méthodes de conception décrites en section I.9

- Le chapitre II décrit un état de l'art sur la détection et l'extraction d'informations temporelles. Nous débutons le chapitre par l'information temporelle en section II.2. La détection et extraction de l'information temporelle présentée en section II3. La section

II.4 décrit la détection et l'extraction d'informations temporelles dans les entrepôts de données**.**

- Le chapitre III présente notre proposition. La section III.2 décrit l'architecture du système proposé. La section III.3 décrit l'étude de faisabilité .

Nous terminons par une conclusion et des perspectives.

Chapitre I

Entrepôt de Données

I.1. Introduction

Les entreprises manipulent un nombre important de données électroniques qui sont stockées dans leurs systèmes opérationnels sous la forme de bases de données, … l'exploitation de ces données est difficile et fastidieuse, elle est réalisée le plus souvent par les décideurs grâce à des moyens classiques (requêtes SQL, vues, outils graphiques d'interrogation…). La préoccupation des dirigeants et décideurs d'une entreprise est l'exploitation des informations contenues dans les systèmes opérationnels afin de garantir une meilleure prise de décision. Les bases de données des entreprises contiennent une grande quantité de données, ce qui les amène à la recherche de systèmes pouvant prendre en compte leur exploitation. Un type de données particulier, qui est directement accessible par le biais des métadonnées d'un document, peut déjà être utilisé par plusieurs tâches telles que la recherche ou le regroupement par rapport à un critère particulier, afin de permettre une meilleure utilisation des données.

I.2. Caractéristiques des entrepôts de données

I.2.1.Définition de l'entrepôt de données

L'entrepôt de données a été introduit pour décrire une plateforme logique facilitant l'accès aux diverses données d'une organisation, et permettant l'utilisation de ces données à des fins d'analyse et de prise de décision [selma khouri, 2009]. Un entrepôt de données représente une collection de données. Ces données sont issues de sources différentes, c'est pourquoi elles doivent être intégrées afin de rendre homogène les modèles de données, la sémantique des données [Golfarelli, 2009]. Les données présentent dans l'entrepôt de données sont représentées dans un même sujet, et ne peuvent pas être modifiés par un utilisateur, et réorganisées afin de permettre l'utilisation de ces données sans oublier le fait que les données sont soumises à une évolution, ce qui nécessite un historique des données.

I.2.2. Principaux traitements dans l'entrepôt

Pour la réalisation d'un entrepôt de données, deux phases sont nécessaires :

1. L'élaboration d'un schéma global de l'entrepôt, appelé modèle d'entreprise,
2. La transposition des assertions au niveau logique afin de permettre la distribution et la réécriture des requêtes sur les sources.

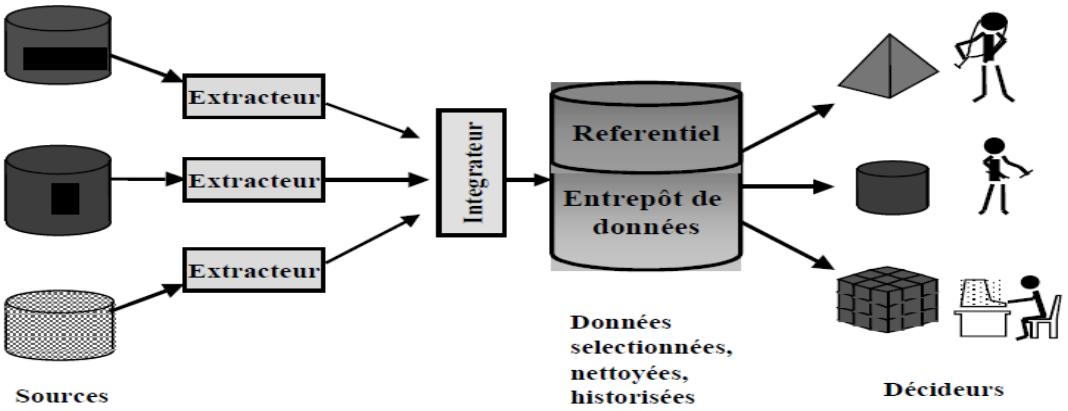


Figure I.1. Schéma de fonctionnement d'un entrepôt de données [Métais et al; 2002]

Une extraction de données de sources hétérogènes est effectuée, ensuite la transformation qui consiste au formatage des données extraites puis leur chargement afin de charger l'entrepôt où le nettoyage et l'homogénéité sont effectués. L'entrepôt de données stocke des données provenant de différentes sources d'information hétérogène et distribuée. Ces sources peuvent être des bases de données, des fichiers de données, des sources externes à l'entreprise. Ce processus vise à construire le schéma de l'entrepôt. Ils stockent les informations des sources internes et externes. Les données peuvent être codées de différentes façons au cours de ses stockages dans les différentes bases de données. L'hétérogénéité des dénominations au niveau des données est traitée dans une phase nommée "nettoyage" permettant de mettre en conformité les données.

Une fois les informations rassemblées, quelques types de logiciel de recherche seront utilisés pour extraire les données de l'entrepôt où elles sont analysées, manipulées et reportées. Ce qui permet d'avoir un accès meilleur et rapide à l'information disponible à tous les niveaux de la corporation.

I.2.3. Les données dans l'entrepôt

La définition des données dans un entrepôt est basée sur des méthodes de vue matérialisée qui sont pris sous plusieurs axes :

- La maintenance incrémentale des vues matérialisées qui se propose de répercuter Immédiatement les mises à jour survenues au niveau des sources de données.

- La configuration de l'entrepôt (sélection des vues à matérialiser) qui se propose de

déterminer un ensemble de vues à matérialiser dans l'entrepôt de telle sorte que le coût de Maintenance soit optimale.

I.2.4. Métadonnées

Rassembler des informations sur les données en plein stockage permet d'éclaircir la définition et la signification des données, en rendant les données plus significatives. Par exemple, est-ce que les données proviennent d'un seul emplacement dans l'organisation ou d'autres sources multiples, et de quand datent-elles, Quand sont-elles rafraîchies : chaque jour, chaque semaine, chaque mois ou moins fréquemment, si les données sont résumées, ou sont stockées. Les métadonnées peuvent être disponibles pour les entreposages de données, afin d'effectuer une analyse de ces données. donc les métadonnées sont décrites comme des données sur les données. Elles permettent une identification des sources ainsi que la relation entre les données.

I.3. Extraction de données

L'extraction peut se faire à travers un outil d'alimentation qui doit travailler de façon native avec les SGBD qui gèrent les données sources. L'inconvénient est le risque de faire des extractions erronées, incomplètes et qui peuvent biaiser l'ED. Il faut gérer les anomalies en les traitant et en gardant une trace.

L'extraction doit se faire conformément aux règles précises du référentiel. Elle ne doit pas non plus perturber les activités de production. Il faut faire attention aux données cycliques. Celles qu'on doit calculer à chaque période, pour pouvoir les prendre en considération. L'extraction peut se faire en interne selon l'horloge interne ou par un planificateur ou par la détection d'une donnée cible (de l'ED) ou en externe par des planificateurs externes. Les données extraites doivent être marquées par " horodatage " afin qu'elles puissent être pistées.

I.4. Entrepôt de données vs Base de données

Un entrepôt de données permet de stocker les données pertinentes aux besoins de prise de décision. Contrairement aux bases de données opérationnelles qui sont conçues pour supporter des opérations journalières, un entrepôt est conçu pour supporter des opérations d'analyse utile à la prise de décision. Le tableau ci-dessous indique la différence entre les bases de données et les entrepôts de données

Caractéristiques	Systèmes opérationnels	Entrepôts de données
But	Exécution d'un processus métier	Evaluation d'un processus métier
Usage	Support à la gestion courante	Support à la prise de décision
Principe de conception	Troisième forme normale	Conception multidimensionnelle
Données	Actuelles, brutes	Historiques, agrégées
Opérations	Lecture et écriture	Lecture et rafraîchissement
Utilisateurs	Employé	Analyste et décideur
Taille	Des giga-octets	Plutôt des téra-octets

Tableau I.1. : Différence entre les bases de données et les entrepôts de données **[**Bellatreche, 2000].

I.5.Construction d'un entrepôt de données

La conception d'un entrepôt de données passe par un cycle de vie qui regroupe les phases suivantes :**[** Golfarelli , 2009].

. La planification : cette phase vise à préparer le terrain pour le développement de l'entrepôt. Elle assure les tâches suivantes :

- Déterminer les buts et objectifs de l'entrepôt à développer,
- Evaluer la faisabilité technique et économique de l'entrepôt, en posant des questions suivant les sources.
- Identifier les futurs utilisateurs de l'entrepôt et leurs rôles.

. Conception et implémentation : Cette phase consiste à développer le schéma de l'entrepôt, et à mettre en place toutes les ressources nécessaires à son implémentation et à son déploiement. Il faut également préparer le déploiement en choisissant une plate-forme logicielle, les ressources, etc. Cette conception comporte cinq principales étapes :

- **Analyse des besoins :** L'analyse des besoins est une étape essentielle dans tout projet de développement de logiciel. Elle permet de réduire les risques d'échec d'un projet. Elle permet l'identification des besoins des utilisateurs et décideurs dans le but de fournir un modèle répondant aux exigences de l'entreprise. pour ce faire, trois étapes sont suivies :

1. **Collecte des besoins** : Afin de comprendre le domaine à modéliser. Deux catégories de méthodes ont été proposées : une catégorie orientée sources qui se fonde uniquement sur les sources de données et une autre catégorie orientée besoins qui inclut les besoins des utilisateurs de l'entrepôt lors de sa conception. Cette collecte se fait par des techniques de réunions, des interviews. Les sources et les besoins ainsi combinés donnent naissance à une méthode hybride.

2. **Analyse des besoins** : Cette étape permet d'analyser les données collectées afin de détecter des besoins conflictuels, contradictoires, complémentaires, etc.

3. **Validation des besoins** : Cette étape permet la validation des modèles initiaux en présence des utilisateurs concernés.

- **La spécification des besoins** : Cette étape permet de déterminer quelles les différentes fonctions de l'entrepôt ainsi que l'ensemble des informations requises que ce dernier doit couvrir, ainsi que les données qui doivent être accessibles.

- **Modélisation des données**

Plusieurs notions peuvent être représentées selon des axes différents, ce qui permet une modélisation multidimensionnelle des données. Dans le cas d'une analyse décisionnelle, l'information est modélisée par une représentation conceptuelle qui est perçue comme une modélisation multidimensionnelle des données. Nous pouvons distinguer plusieurs méthodes de modélisation multidimensionnelles :

- **Modélisation conceptuelle**

Cette modélisation prend en compte deux points importants : le fait et la dimension. Le fait permet de modéliser le sujet qui est formé de mesures numériques afin d'assurer une minimisation d'enregistrement correspondant aux informations de l'activité analysées. Tandis que la dimension permet de modéliser une perspective de l'analyse, cette dimension se comporte des paramètres correspondant aux informations faisant varier les mesures de l'activité, ces paramètres d'une dimension sont organisés selon une hiérarchie en s'appuyant sur la relation "est plus fin". La dimension sert à enregistrer les valeurs pour lesquelles sont analysées les mesures de l'activité.

Une structure de données simple peut en résulter des deux concepts afin de correspondre aux besoins de la modélisation multidimensionnelle. Ces concepts sont représentés de sorte que le fait est au centre et les dimensions autour ce qui se traduit par une modélisation en étoile. Cette dernière peut être décomposée en sous-hiérarchies des dimensions, ce qui se traduit par une émanation de la modélisation en étoile. Une autre technique de modélisation portant le nom de constellation consiste à fusionner des modèles en étoile.

- **Modélisation logique**

La modélisation multidimensionnelle au niveau logique peut être vue de différentes manières :

- Un système de gestion de base de données (SGBD) comme les SGBD relationnels (ROLAP). Dans ce cas, le fait et la dimension sont représentés par des tables, la table de fait est constituée d'attributs représentant les mesures d'activité et les attributs clés étrangères de chacune des tables de dimension. Les tables de dimension contiennent les paramètres et une clé primaire permettant de réaliser des jointures avec la table de fait. Ou bien les SGBD orientées objet (OOLAP) qui s'appuie sur le paradigme objet. Dans ce cas le modèle multidimensionnel se traduit par une classe de fait concernant chaque fait et une classe de dimension concernant chaque dimension.

- Un système de gestion de bases de données multidimensionnelles (MOLAP) qui gère des structures multidimensionnelles natives qui sont des tableaux à n dimensions. Cette approche permet de stocker les données de manière multidimensionnelle dont le but est que les temps d'accès sont optimisés, mais afin de manipuler ces structures multidimensionnelles, des opérations doivent être redéfinies.

- **Processus ETL (Extract-Transform-Load)**

ETL est un processus qui permet l'intégration des données qui proviennent de sources hétérogènes. Le processus est à l'origine de l'intégrité et de l'exactitude des données de l'entrepôt. Cette étape consiste à effectuer les transformations nécessaires au chargement des données des sources au niveau du schéma logique de l'entrepôt. Au début, une extraction de données de sources hétérogènes est effectuée, ensuite la transformation qui consiste au

formatage des données extraites puis leur chargement afin d'alimenter l'entrepôt où le nettoyage et l'homogénéité sont effectués.

- **Modélisation physique**

Cette étape consiste à implémenter physiquement le schéma de l'entrepôt, l'administrateur doit sélectionner des structures d'optimisation comme les index ou les vues matérialisées pour optimiser les accès à l'entrepôt et à spécifier les techniques et schémas d'optimisation de l'entrepôt. Les techniques d'optimisation peuvent être les vues matérialisées, les techniques d'indexation qui permettent d'associer à une clé d'un n-uplet l'adresse relative de ce n-uplet. Nous distinguons les index définis sur une seule table ou vue) et les index de jointures définis sur plusieurs tables dans un schéma en étoile ainsi que la fragmentation qui consistent à partitionner le schéma d'une base en plusieurs sous-schémas pour réduire le temps d'exécution des requêtes.

. Maintenance et évolution

La maintenance de l'entrepôt implique l'optimisation de ses performances périodiquement. L'évolution de l'entrepôt concerne la mise à jour de son schéma en fonction des différents changements survenant au niveau des sources ou des besoins des utilisateurs.

I.6. Méthodes de conception

Nous pouvons distinguer trois approches de conception d'entrepôts de données : [selma khouri, 2009].

I.6.1. Les approches directes

Cette approche engendre le schéma logique d'un entrepôt de données à partir des sources ou des besoins directement, c'est-à-dire en évitant la phase de modélisation conceptuelle. Ce principe s'explique par le fait que les entrepôts soient issus originellement de l'industrie qui s'intéresse aux aspects pratiques et néglige les problèmes de conceptualisation, mais également par le fait que les modèles logiques et physiques d'un entrepôt jouent un rôle important dans l'amélioration et l'optimisation des performances du système décisionnel final. Ce schéma logique est défini à partir des sources de données (selon Inmon) ou des besoins organisationnels ou des deux (selon Kimball). Ces approches impliquent un important travail d'abstraction, et aussi une connaissance préalable du modèle multidimensionnel (faits et dimensions) par les décideurs. De plus, la génération directe du schéma logique présente certaines limites comme le fait de prendre des décisions concernant le niveau physique

(normalisation des tables). Un autre inconvénient concerne hétérogénéité des schémas des sources de données. Le concepteur doit analyser les sources de données afin de construire le schéma logique. Il se trouve ainsi devant une tâche complexe, où il doit gérer les différents conflits (syntaxiques et sémantiques) entre les données des sources.

I.6.2. Les approches traditionnelles

Ces approches s'intéressent à mettre en place une vraie phase de conception lors du développement d'un entrepôt. Nous distinguons deux catégories de méthodes :

- **Les méthodes orientées sources :** qui indiquent que le développement d'un entrepôt repose sur les données des sources par opposition aux méthodes de développement des bases de données traditionnelles qui reposent sur les besoins des utilisateurs. Ces méthodes étendent les schémas E/A par des faits et des dimensions, ou bien dérivent un modèle multidimensionnel de l'entrepôt à partir des schémas E/A ou des schémas relationnels des sources en utilisant des règles de transformation.

- **Les méthodes orientées besoins utilisateurs :** plusieurs travaux reconnaissent la nécessité d'identifier et de modéliser les besoins des utilisateurs pour la conception d'un entrepôt. La modélisation de ses besoins s'effectue soit en utilisant des modèles spécifiques, en utilisant une modélisation objet. Les méthodes fondées sur les besoins permettent de générer un schéma conceptuel multidimensionnel de l'entrepôt en prenant en compte les besoins des décisionnels de l'organisation. Un mapping entre les sources et les besoins doit être effectué. Ce mapping nécessite une étape d'intégration.

I.6.3. Les approches ontologiques

Les ontologies ont été proposées pour résoudre ce problème d'intégration inhérent à la conception d'un ED. Une ontologie peut être définie comme une spécification d'une conceptualisation d'un domaine. Elle permet de décrire explicitement la sémantique des données. une méthode semi-automatique de conception multidimensionnelle d'un ED, fondée sur une ontologie représentant le domaine d'intérêt de l'entrepôt. Cette méthode étudie les multiplicités entre les concepts de l'ontologie pour définir les concepts multidimensionnels „faits" et „dimensions" du schéma conceptuel. Ainsi, un concept de l'ontologie est considéré comme un fait potentiel sil est relié à autant de dimensions et mesures que possible. Les mesures sont désignées comme les attributs numériques permettant l'agrégation des données.

Ils sont reliés au concept représentant le fait par une relation (1,1). Un concept de l'ontologie est considéré comme dimension potentielle sil est rattaché à un fait par une relation (1,n). Les hiérarchies de dimensions sont déterminées en recherchant les relations (n,1) entre les concepts identifiés comme dimensions. Cette identification peut se faire de manière automatique. Les résultats sont ensuite présentés au concepteur qui doit les valider.

I.7. Conclusion

Ce chapitre décrit un état de l'art sur les entrepôts de données, nous avons défini les caractéristiques des entrepôts, les principaux traitements ainsi que le cycle de vie de l'entrepôt et ses modèles de données.

Un aperçu sur les méthodes de conception a permis d'avoir une idée sur la conception des entrepôts de données pour mieux comprendre le fonctionnement ainsi que leur construction qui se fait à partir des données stockées dans les sources opérationnelles. Il est possible de prendre en compte les besoins des décideurs et des utilisateurs du futur ED, et de les confronter aux données des sources afin de développer un entrepôt répondant au mieux aux besoins de l'organisation selon des critères. L'ensemble des informations répondant aux critères peuvent être regroupées afin de permettre une meilleure utilisation. Dans le cas des informations répondants aux critères du type de données temporelles, l'entrepôt de données sera orienté type de données temporelles. La tâche extraction sera orientée vers le type temporel. A ce titre, le chapitre suivant présente la détection et l'extraction des données temporelles dans les entrepôts de données.

Chapitre II

Etat de l'art sur la détection et l'Extraction d’informations temporelles

II.1. Introduction

Le traitement automatique de l'information temporelle est défini pour la connaissance des éléments qui explique ce type d'information dans les textes en langage naturel, à les interpréter afin de leur donner une valeur univoque dans l'espace du temps et à en fournir une représentation normalisée. Le résultat de ce traitement est essentiel pour une exploitation aisée et complète des informations temporelles. Plusieurs applications peuvent recourir à la temporalité dans différents domaines, comme la recherche ou l'extraction d'informations, afin de satisfaire leur objectif.

Les expressions temporelles sont des éléments intéressants pour l'indexation de textes. Si l'aspect thématique représente une dimension très changeante de l'information, la dimension temporelle est plus stable que ce soit du point de vue de son expression qu'en ce qui concerne sa distribution dans les différents textes [Laurent, 2011]. Les informations thématiques demandent une méthode adaptable au domaine exploité tandis que le traitement de l'information temporelle reste une tâche assez peu variable.

II.2. Information temporelle

II.2.1. Information

L'information est un concept ayant plusieurs sens. Il est étroitement lié aux notions de contrainte, communication, contrôle, donnée, formulaire,instruction, connaissance,

Signification, perception et représentation. Au sens étymologique, *l'information* est ce qui donne une *forme* à l'esprit. Elle vient du verbe latin *informer*, qui signifie « donner forme à » ou « se former une idée de ».

L'information désigne à la fois le message à communiquer et les symboles utilisés pour l'écrire, elle utilise un code de signes porteurs de sens tels qu'un alphabet de lettres, une base de chiffres, des idéogrammes ou pictogrammes. Hors contexte, elle représente le véhicule des données comme dans la théorie de l'information et, hors support, elle représente un facteur d'organisation.

II.2.2. La temporalité linguistique

Le temps peut être considéré comme une localisation temporelle, il indique une référence temporelle déictique par rapport au temps de l'énonciation, une référence temporelle anaphorique par rapport à un moment donné dans le discours.

Chaque événement est lié à des informations temporelles, figuré dans un agenda ou un horaire, les informations temporelles sont véhiculées par les temps verbaux dans la datation et la chronologie (classement, ordre dans le temps).

Dans la vie courante, nous manipulons des concepts temporels, qui en résultent d'un système de représentation. Il existe bien alors une relation entre le monde réel et ces concepts temporels.

Beaucoup de travaux de recherche ont abordé la temporalité dans le langage naturel comme l'extraction d'informations ou encore l'ingénierie des connaissances. Vu l'importance du temps, de nombreux moyens qui expriment la temporalité apparaissent. La complexité des problèmes posés par le traitement du temps en langage naturel ne permet pas de l'aborder sous tous les angles.

Le temps peut être exprimé par de nombreux moyens, comme des expressions adverbiales, des connecteurs qui ont une dimension temporelle « et » pour expliquer la suite de deux actions. Enfin, les événements ont aussi une dimension temporelle, implicite, auxquels une granularité naturelle leur est associée.

Un autre moyen qui véhicule la dimension temporelle : les dates dites absolues qui sont invariables et les dates dites relatives. Egalement, on peut recourir à des expressions temporelles imprécises, cependant ce type d'expression peut conduire à une ambigüité, l'exemple suivant illustre cette ambigüité :
-En mars, les prix des véhicules vont baisser, dont la période n'est pas claire.

L'organisation d'un texte dans le domaine temporel, nécessite des moyens véhiculant la dimension temporelle, cependant l'étude de la temporalité dans les textes connait un certain nombre de problèmes en fonction du type du moyen qui véhicule la temporalité. Analyser le

temps revient à comprendre le texte dans sa globalité et plus précisément des événements du texte, elle rencontre souvent des problèmes de modalités et d'anaphore.

Il existe des mécanismes de raisonnement qui permettent de calculer l'extension temporelle des périodes. Ce raisonnement permet d'effectuer un changement des instances selon un modèle linguistique sous forme d'intervalles de temps qui représente un modèle calendaire, pour lesquels un statut d'accessibilité est attribué . [Teissedre, 2012]

II.2.2.1.La localisation temporelle

La localisation temporelle peut être exprimée par des représentations qui peuvent être groupées en trois groupes en fonction de l'information de base considérée pour former des informations temporelles :

- Le point : fait référence à une suite ordonnée d'instants ponctuels, l'associer à un fait revient à lui associer tous les instants pour lesquels le fait est vrai.

- L'intervalle : la valeur d'un événement est connue par un ensemble d'intervalles.

- L'occurrence : dans ce cas, le temps est issu des occurrences qui se produisent, chaque occurrence définit un instant.

II.2.2.2. Traitement de la temporalité

Tout traitement de temporalité prend en compte des étapes permettant l'aboutissement à des objectifs bien déterminés. Avant de faire l'extraction, un prétraitement est nécessaire pour réaliser le traitement, c'est pourquoi l'interprétation des informations temporelles est réalisée. De l'autre, les événements eux aussi véhiculent la temporalité, ce qui est inclus dans le traitement de la temporalité. Les informations temporelles que l'on souhaite extraire doivent être interprétées afin de leur donner une valeur univoque, c'est-à-dire lui attribuer une valeur temporelle relative à un système calendaire.

Mais avant de les interpréter, il faut d'abord les reconnaitre en parcourant l'ensemble des informations temporelles disponibles, permettant ainsi une exploration complète.

Plusieurs auteurs ont consacré leurs efforts à ces opérations dont le but de les approfondir. Nous pouvons citer les travaux de [Bittar, 2009], dont il explique clairement la procédure d'annotation ainsi que la prise en compte des événements qui eux aussi entrainent la temporalité, cependant les événements englobent un large éventail et correspondent à des aspects différents, mais les systèmes d'extraction d'informations temporelles se limitent à un type bien défini selon les besoins ainsi que les objectifs visés par le système.

Parmi les événements considérés par les différents systèmes, nous pourrons distinguer les verbes, ainsi que les événements qui sont véhiculés par des verbes finis, alors que d'autres considèrent les verbes et les groupes nominaux.

Selon le format d'annotation Time Ml (voir annexe B), les événements considérés sont ceux qui engendrent des verbes, des états, des noms événementiels ou encore des adjectifs. Il existe aussi des relations entre les événements et les expressions temporelles, qui consistent à établir des relations entre les événements et le temps calendaire. Nous pouvons remarquer aussi, les relations événement-événement sont elles aussi, inspirées des relations qui ont été proposées par Allen.

Le nombre de relations proposé par Allen est très grand, ce qui pose problème au niveau de l'annotation manuelle, et ce qui amène certains auteurs à utiliser un nombre plus petit de relations.

II.2.3. Le temps et l'information

La spécification de l'information temporelle constitue un enjeu au niveau des systèmes d'annotation automatiques, du fait qu'ils doivent être en mesure de savoir quelles sont les unités qui expriment l'information temporelle, ainsi que les niveaux de représentation qui permettent une appréhension de la sémantique du temps. Les informations temporelles peuvent regrouper des expressions de dates comme le 16 janvier, des expressions d'heure comme midi, des expressions de durée comme 2 minutes, elles peuvent regrouper aussi des événements sous forme de verbes, de noms déverbaux, ou autres noms, ou encore les états sous forme d'adjectifs ou verbes. Ces informations ont une importance qui réside dans le fait qu'elles assurent une bonne compréhension du texte.

Trois aspects importants de l'information temporelle peuvent être dégagés:

- Des groupes de mots véhiculant l'information temporelle qui facilitent la reconnaissance et l'annotation.
- L'interprétation de groupes de mots pour donner une valeur relative.
- Le dernier aspect, la structuration du discours par les valeurs temporelles.

D'après [Turenne , 2004], l'information temporelle peut aussi être vue selon trois aspects:

- Un aspect linguistique dans lequel, l'accent est mis sur la sémantique du fait qu'elle permet de résoudre le problème d'ambigüité qui se trouve dans une expression ou dans une phrase. Concernant la sémantique temporelle, deux approches la traite : une qui se fonde sur une analyse du temps verbal et une autre qui considère des connecteurs.

- Un aspect statistique dans lequel un document contient des champs classiques à savoir le titre, le résumé et la date de création. Ces informations calendaires peuvent ordonner chronologiquement les documents. L'analyse du contenu d'un document électronique par exemple passe par plusieurs étapes :

    - ❖ Traiter des variables statistiques sous forme d'entités extraites à partir d'un texte.

    - ❖ La création d'une matrice de cooccurrences pour la création d'une représentation de vectorielle.

    - ❖ Effectuer des traitements selon l'algorithme choisi.

- Aspect ontologique qui prend en compte les modèles de temps en intervalles, tout en négligeant la notion de temps. [Allen et al., 2010] proposent des intervalles temporels qui ont été pris par l'intelligence artificielle, et même dans le web sémantique.

II.2.3.1. Les expressions qui expriment le temps

Nous distinguons les expressions suivantes :

- **Les adverbiaux temporels**

Les adverbiaux temporels véhiculent des informations importantes concernant les relations qui existent entre les événements, un problème se pose lors de la reconnaissance de l'événement auquel ils sont rattachés et la déduction des informations temporelles qu'ils expriment.

Le temps peut être exprimé par des expressions qui ont une valeur temporelle. Les expressions qui sont regroupées sous le nom d'adverbiaux temporels, qui selon Laurent sont [laurent, 2011] :

- Les adverbiaux de référence temporelle comme aujourd'hui
- Les adverbiaux de durée : en cinq semaines
- Les adverbiaux de fréquence : tous les jours
- Les adverbiaux itératifs : plusieurs fois
- Les adverbiaux de quantification : toujours
- Les adverbiaux présuppositions : encore

[Borillo et al, 2004] distinguent plusieurs formes possibles des adverbiaux temporels :

- Adverbes simples ou composés : demain, plus tard
- Syntagmes prépositionnels : en mars
- Formes nominales : le premier jour, mardi dernier

Au niveau sémantique, les adverbiaux temporels constituent une catégorie homogène, le tableau ci-dessous montre les différents types, soit par rapport au calendrier ou à l'énonciation ou à un procès.

Typologie	Niveau sémantique	Niveau morphosyntaxique
Base calendaire De 1999 à 2015	Adverbiaux Temporels	Syntagmes prépositionnels Avant 2015
Base relative à un procès Jusqu'au début du match		Subordonnées temporelles Depuis qu'il est rentré
Base déictique En ce moment		Syntagmes nominaux La veille
Base anaphorique Ce soir là		

Tableau II.1. Les différents types d'adverbiaux temporels [Teissedre, 2012]

- **Les connecteurs temporels**

Les connecteurs temporels sont des éléments qui possèdent des caractéristiques d'adverbes temporels et de connecteurs de discours. Borillo *et al.* [2004] distinguent trois catégories d'adverbes qui peuvent être considérées comme des connecteurs, qui sont :

- Les adverbes relationnels, dont le rôle est de lier deux événements, et qui, véhiculent la temporalité ou l'ordre chronologique existant entre deux événements.

- Les adverbes de référence anaphorique, dont le rôle est de faire entrer la temporalité concernant une situation connue par rapport au moment de l'énonciation.

- Les adverbes aspectuo-temporels, qui font référence à la modalité, tout en prenant une dimension temporelle.

- **Le procès**

Le procès regroupe des faits, états, un événement ou une action. Il existe différentes typologies de procès. Il y a le type de procès qui regroupe des verbes et des prédicats à partir de leurs caractéristiques temporelles, qui peut lui-même avoir une influence sur l'interprétation temporelle. D'une façon générale, les types de procès sont regroupés selon des classes, le tableau ci-dessous illustre les différentes typologies.

References	Classes			
KENNY[1963]	Etat	Activité	Performance	
Vendler[1957]	Etat	Activité	Accomplissement	Achèvement
Mourelatos[1978	Etat	Processus	Développement	Occurrences ponctuelles

Tableau II.2: Différentes typologies de procès**.** [Laurent, 2011]

- **Le temps**

Il y a plusieurs types ou niveaux de temps, qui est le temps physique « un continu uniforme, infini, linéaire. Il a pour corrélat dans l'homme une durée infiniment variable que chaque individu mesure au gré de ses émotions et au rythme de sa vie intérieure », chronique qui est « le temps des événements, qui englobe aussi notre propre vie en tant que suite d'événements » et grammatical. Les temps verbaux expriment la temporalité d'une manière très directe.

- **L'aspect**

Il fait référence à la structure temporelle des actions ou des situations décrites par les phrases. C'est une interprétation des temps grammaticaux, il ya deux notions d'aspects [laurant, 2011], l'aspect lexical qui exprime le procès tel qu'il est conçu et l'aspect grammatical tel qu'il est montré. Le temps et l'aspect expriment tous les deux la temporalité. Un autre point de vue qui considère la mise en relation avec les types de procès.

- **La structure du discours**

La structure de discours exprime elle aussi la temporalité, elle concerne l'ordre d'apparition des événements qui composent le texte, cet ordre permet une interprétation dans le domaine temporel. Ces événements sont généralement liés par marqueurs, donc, il existe des relations entre ces événements, qui peuvent être exprimées soit par un lien causal, soit par la connaissance des types d'événements et de sous-événements.

- **Les cadres de discours**

Les cadres de discours correspondent à plusieurs types d'univers : spatiaux, de connaissances, de représentation, d'énonciation et temporels. Ce dernier type structure et

organise les expressions temporelles, en effet les introduire en début de phrase, permet une structuration du texte [Laurent, 2011]

II.2.3.2. Représentation d'informations temporelles

La communauté IA distingue deux catégories de formalismes de représentation par rapport au type de raisonnement envisagé:

- Les formalismes algébriques [laurent, 2011] telles que les algèbres d'intervalle. L'algèbre des intervalles proposés par Allen s'adapte à la représentation des phénomènes qui se répètent souvent. Cette algèbre s'adapte à la représentation des phénomènes les plus fréquents. Elle met en évidence les relations qui existent entre ces intervalles sous forme de disjonction des relations de bases. La relation qui existe entre les intervalles temporels se présente sous forme de disjonction entre les différentes relations décrites par Allen. Cette algèbre a été étendue en considérant d'autres types d'intervalles qui expriment des actions sporadiques, et d'autres qui expriment des actions répétées. Les Relations temporelles sont explicitement marquées par des prépositions temporelles (par exemple avant, sur ou par). Les sept relations temporelles suivantes: before, after, incl, at, starts, finishes, excl. La préposition on comme on Friday, désigne la relation incl rapport, alors que la préposition by comme by Friday est représentée comme une relation finishes. Les relations décrites par Allen sont présentées ci-dessous.

Before	{b,m}
After	{bi, mi}
Incl	{d,s f,eq}
At	{di,si,fi,eq}
Starts	{s}
Finishes	{f}
excl	{b,bi,m,mi}

Tableau II.3 : Relations d'allen

- Représentations proposées par la linguistique informatique [laurent, 2011] dans laquelle l'extraction et l'annotation d'informations temporelles peuvent être représentées par des formalismes proposés par des linguistes informaticiens. Parmi les langages d'annotation pour la normalisation des informations temporelles, nous citons le Time ML [Katz et al., 2005] qui est utilisé pour rendre le raisonnement et l'inférence sur le temps plus facile. Ce schéma prend en compte l'annotation des événements, des expressions temporelles, il met en évidence les relations entre les événements et les expressions.

II.3.Détection et extraction de l'information temporelle

II.3.1. Détection des informations temporelles

La détection et la normalisation des informations temporelles telles que les expressions temporelles et les événements, ont été bien exploités dans plusieurs travaux. La compagne d'évaluation tempeval2 s'intéresse à la détection des relations qui existent entre les événements ainsi que les événements et les expressions temporelles, ce qui explique l'utilisation des annotations prédéfinies. [laurent, 2011]

Ces outils utilisent du texte brut, d'où la nécessité de séparer la détection des événements et celles des expressions temporelles. Nous remarquons l'utilisation d'une combinaison d'approche, celles qui utilisent des règles et des ressources linguistiques pour permettre l'extraction des expressions et les événements, ainsi qu'une approche d'apprentissage.

L'outil décrit par [allen et al 2010] Trips, fait une analyse du texte, il se fonde sur la logique de réseau markovienne, utilise aussi une ontologie linguistique pour la description de la sémantique, et un étiqueteur morphosyntaxique, ce qui permet une liaison entre la sémantique et la syntaxe. L'annotation des événements se fonde sur des patrons manuels. Il est utilisé pour produire des formes logiques dans un texte, il n'ya pas de règles grammaticales ni d'entrée lexicale ajoutée. La grammaire trips est lexicalisée indépendamment du contexte. Elle est augmentée par des caractéristiques de structures et d'unification. Elle est motivée par la théorie X-barre.

Un autre outil sous le nom de Trios a été utilisé, qui inclut l'annotateur, l'analyseur et un post traitement. Il s'intéresse aux événements spécifiés dans timeml. Contrairement à

l'outil trios, le corpus doit être annoté. Il permet l'annotation des expressions temporelles, et donne une description des liens entre les événements annotés et les expressions temporelles, il peut même assurer la normalisation des expressions selon la DTC. Cet outil a été testé dans tempeval2 et de bons résultats ont été observés pour les tâches d'annotation et de normalisation, contrairement à l'identification des relations temporelles.

D'autres outils comme Evita, utilisent un analyseur morphosyntaxique, et prennent en compte la tâche d'identification des événements, il a été testé, lui aussi dans tempeval2, et il a fourni un résultat dont le rappel est meilleur que la précision.

Pour le Français, des outils ont vu le jour, comme [Bittar, 2008], qui utilise des automates à état fini pour l'annotation des informations temporelles. La tâche d'extraction fournit un bon résultat contrairement à la tâche de normalisation. Les événements quant à eux sont repérés et typés selon les différents attributs, il utilise des patrons lexicaux.

[weiser et al 2010] décrivent, eux aussi une chaine de traitement linguistique qui permet d'assurer le repérage et l'annotation des expressions temporelles, qui sont réalisés à l'aide des patrons linguistiques, ils s'intéressent particulièrement aux objets touristiques.

Il existe divers moyens de définir l'information temporelle, qui permettent de leur donner une signification plus précise et une sémantique temporelle. Pour ce faire, des formats d'annotations ont été développés dont le but est de fournir des éléments qui ont le même aspect ainsi que des propriétés spécifiques. Parmi les différents formats d'annotation, nous distinguons les plus importants comme Timex, timex 2 et TimeMl.

Afin de permettre une bonne extraction d'informations temporelles, il faut choisir un langage d'annotation qui correspond le mieux, car ce choix influence le résultat d'extraction, et sur tout lors de la reconnaissance, sans oublier l'étape d'interprétation, qui elle aussi utilise les résultats de l'annotation.

III.3.1.1.Formats d'annotation

II.3.1.1.1.Timex

Ce format prend en compte des entités nommées de type temporel, qui peuvent être des expressions absolues ou relatives. Au niveau de ce format, nous distinguons deux types, à savoir le type date et le type time, ces deux types prennent en compte des expressions temporelles de forme intervalle. Le type date prend en compte l'ensemble des expressions temporelles qui expriment des jours, des saisons, des trimestres, des années, des décennies et des siècles. Les expressions temporelles relatives sont aussi prises en compte par le type date, car elles aussi expriment la date. Un autre type d'expression est aussi considéré, qui fait référence aux expressions composées. Les expressions de jours particuliers qui sont référencés par des noms particuliers sont elles aussi, prisent en compte par ce type.

Le type time par contre prend en compte des expressions qui expriment le temps en minutes, qui sont souvent spécifiques, et des heures qui elles aussi sont particulières. Une extension du format Timex, timex2, l'expression annotée dans ce format se voit attribuer une valeur normalisée, d'où l'apparition du champ val qui associe une valeur à l'expression annotée selon une norme spécifiée. Afin de permettre une annotation, les expressions temporelles doivent être composées d'un déclencheur lexical qui véhicule la temporalité, qui peut être un nom, une entité nommée, des patrons temporels, des adjectifs, des adverbes temporels et des nombres. L'étiquette utilisée est timex2. Sous ce format, plusieurs attributs sont utilisés afin de spécifier des valeurs normalisées, des modificateurs …

Nous distinguons des expressions temporelles qui peuvent être précises, c'est-à-dire que nous pouvons les déterminer sur le système calendaire, dans ce cas la date, l'instant ou la durée sont précis. Leur annotation se fait selon le type des expressions, qui peuvent être sous forme de deux expressions, dans ce cas leur annotation se fait pour chaque expression séparément, ou encore une annotation imbriquée. Les expressions temporelles floues sont des expressions qui ne sont pas précises, et dans ce cas l'annotation se fait sur la granularité présente dans l'expression temporelle. Dans ce cas des expressions qui font référence aux temps passé, présent, futur sont incluses, c'est pourquoi le champ val est étendu aux valeurs qui prennent en compte des références liées au passé, futur, présent. Et d'autres expressions qui contiennent des saisons, elles aussi expriment une temporalité floue, puisque la notion de saison est vague, et elle est considérée par rapport à des périodes de l'année ou autre chose, ce

qui a augmenté les valeurs prises par le champ val pour indiquer les autres valeurs possibles prises hors des mois, qui sont spécifiées dans le format ISO.

II.3.1.1.2.TimeMl

Ce format est un langage de spécification, qui facilite le raisonnement sur le temps. Il donne un cadre d'annotation manuelle ou automatique des événements, des expressions temporelles et les liens entre ces éléments. Il s'intéresse aux événements et à leur détection, il est fondé sur Timex2, mais il permet d'identifier en plus les signaux qui interviennent dans l'interprétation des expressions temporelles, les connecteurs temporels comme avant, après, tant que. Ce format opère sur les événements qui se présentent sous forme de verbes, des états, des adjectifs ou des groupes nominaux, il indique aussi les liens existant entre les différents événements et expressions temporelles. Ce format comporte quatre balises : Event, Timex3, Signal, Link.

La balise Timex3 prend en compte les expressions temporelles qui sont spécifiées, sous spécifiées ainsi que les durées. Elle comporte quatre classes :

- Les dates
- Les heures
- Les durées
- Les ensembles

La balise signal s'intéresse à la façon dont les expressions temporelles sont liées, et sont exprimées à l'aide des prépositions temporelles comme « pendant, avant, après», des connecteurs temporels, des conjonctions de subordination, des indicateurs de polarité et de quantification temporelle, des caractères spéciaux.

La balise Link s'intéresse aux liens qui existent entre expressions, il existe trois types : Des liens temporels qui expriment des relations entre des événements ou entre des événements et des expressions temporelles, comme celles décrites par Allen, qui expriment des relations entre deux événements ou entre événement et un signal, et les liens aspectuels qui expriment des relations entre l'événement aspectuel à son événement argument.

Nous décrivons des exemples d'annotation :

- Pour les expressions temporelles avec la balise <timex3>

-Un tremblement de terre a frappé Boumerdes le 21 mai 2004

-Un tremblement de terre a frappé Boumerdes le <TIMEX3 tid="t1" type="DATE" val="20040121"> **21 mai 1993**</TIMEX3>.

-L'examen a duré 3heures et 30minutes

– L'examen a duré <TIMEX3 tid="t2" type="DURATION" val="PT3H30">**3heures et 30 minutes**</TIMEX3>.

-Le voyage a lieu 1fois par an.

–Le voyage a lieu <TIMEX3 tid="t3" type="SET" freq="1X" val="P1Y">**1 fois par an**</TIMEX3>.

-Il sera disponible entre samedi et mardi.

– Il sera disponible entre <TIMEX3 tid="t4" type="DATE" val="WXX1">**samedi**</TIMEX3> et <TIMEX3 tid="t5" type="DATE" val="WXX3">**mardi**</TIMEX3>.

Les valeurs sont celles accordées par la norme ISO 8601.

- Pour les événements avec la balise <EVENT>

-Une tempête de froid a touché la ville.

-Une tempête de froid <EVENT eid="e1" tense="PAST" aspect="PERFECTIVE">**a touchée**</EVENT> la ville.

-Les précautions ont été prises après l'attentat.

-Les précautions ont été prises après l'<EVENT eid="e2" tense="NONE" aspect="NONE">**attentat**</EVENT>.

Cependant, ce format s'adapte le mieux pour une annotation manuelle, mais ne prend pas en compte des expressions de la forme 4eme samedi du mois.

Ces différents formats décrits représentent des formats standards utilisés par de nombreux travaux, cependant certains considèrent d'autres expressions ayant plus de

caractéristiques, ce qui amène à voir naître d'autres formats pour prendre en compte ses caractéristiques.

II.3.1.2. Evaluation du traitement de la temporalité

Afin d'évaluer la temporalité, plusieurs compagnes ont vu le jour pour y répondre à certaines questions, et donner un point de vue sur les tâches effectuées aux moments présent, ainsi que les systèmes qui les prennent en compte. Ces compagnes prennent en compte des tâches particulières.

Nous pouvons citer la compagne tempeval2, qui permet de faire une évaluation des systèmes d'annotation des expressions temporelles, se basant sur le langage TimeMl. Autres conférences comme MUC s'intéressent à l'évaluation des taux de précision et de rappel. Elle s'intéresse particulièrement aux traitements automatiques de messages, mais avec l'évolution des travaux, elle s'intéresse aux tâches d'extraction, de reconnaissance d'événements, qui sont limités aux messages concernant des domaines particuliers. Le langage TimeMl permet d'évaluer l'annotation des événements, dans la compagne TempEval, ce qui a élargit les informations visées par les différents systèmes.

Parmi les tâches réalisées dans tempeval2, la détection des informations temporelles selon le format timeml, qui utilise la balise Timex3. Cette tâche est accompagnée de la tâche de normalisation qui consiste à attribuer une valeur calendaire. La détection des événements selon le même format utilise la balise Event.

Une autre tâche consiste à détecter la relation temporelle qui existe entre les événements et l'expression temporelle, entre l'événement et la date de création du document et entre deux événements.

II.3.2. Extraction d'informations temporelles

L'extraction d'informations consiste à l'analyse du texte de manière partielle, permettant ainsi une extraction d'information spécifique. Généralement la tâche d'extraction regroupe trois sous-tâches [laurent, 2011] :

- La reconnaissance des entités nommées.
- La reconnaissance des relations entre les entités nommées.
- Et la reconnaissance d'événements.

La tâche de reconnaissance des entités nommées est une tâche assez complexe et difficile, cependant la reconnaissance des relations qui existent entre les entités et les événements représente une tâche souvent prise en compte par les outils d'extraction, sans oublier la tâche d'extraction des événements qui peut être considérée comme un sous ensemble de relation. Ces tâches sont importantes pour le développement d'outils d'extraction, d'où l'apparition de diverses compagnes d'évaluations, qui ont fait émerger plusieurs approches qui font appel à des prétraitements.

L'extraction d'informations temporelles est une tâche qui prend en compte des caractéristiques selon le type d'information que l'on souhaite extraire.

Plusieurs méthodes ont été développées sur ce sens, afin de permettre une extraction complète de l'information temporelle. Ces méthodes regroupent généralement des prétraitements qui assurent le traitement de la temporalité.

La reconnaissance des informations temporelles est un traitement permettant de parcourir un texte en détectant ces informations. Ce traitement est suivi de l'annotation qui lui attribue une valeur aux informations extraites. Ces valeurs sont données selon un système calendaire.

L'extraction d'informations temporelles considère deux tâches essentielles, la tâche de repérage des informations ainsi que la tâche d'interprétation. L'extraction d'informations temporelles a été élaborée selon plusieurs approches, qui peuvent être regroupées en deux classes : une approche symbolique et une approche fondée sur l'apprentissage automatique.

II.3.2.1. Approche symbolique

L'extraction est fondée sur les techniques linguistiques . Le principe de cette approche est d'établir des règles manuelles, afin d'assurer le repérage et l'extraction des informations souhaitées. Elle se base sur les automates pour la création des patrons associés aux règles.

Cette approche se focalise sur les systèmes de règles et les ressources linguistiques. Elle vise à atteindre un bon niveau de précision, tant dit que le rappel est lié aux ressources développées. Plusieurs auteurs se sont penchés vers cette approche dont le but de décrire les étapes de reconnaissance et d'interprétation. Des automates à état fini ont été utilisés pour

décrire ces étapes qui prennent en compte des types d'informations bien précises comme les dates et les adverbes.

Cette approche fondée sur les automates à état fini représente une base pour d'autres travaux, comme pour Hagege et Tannier. Le logiciel Xip de Xerox utilise, lui aussi, cette approche.

Dans cette approche, l'étape d'interprétation n'est pas aussi précise à cause des automates, car elle nécessite des ressources indispensables à cette interprétation, c'est pourquoi des règles sont manipulées pour la reconnaissance, ainsi que l'interprétation qui peut être mise en œuvre de plusieurs façons.

Nous pouvons citer quelques travaux qui abordent les règles pour la reconnaissance, comme [Habel et al, 2001].

II.3.2.2. Approche d'apprentissage

Cette approche considère un texte annoté, et utilise des caractéristiques plus ou moins linguistiques. Les informations extraites par cette méthode doivent être mises en relation. Elle permet de mettre en évidence l'étape de reconnaissance d'informations temporelles, contrairement à l'interprétation de celles-ci, qui s'annonce moins efficace, et nécessite des règles pour une meilleure interprétation. En effet, à une expression temporelle peut correspondre plusieurs classes selon une classification, ce qui rend son interprétation difficile.

L'extraction d'informations temporelles s'est basée aussi sur les méthodes d'apprentissage, ces méthodes sont aussi utilisées pour la détection des informations temporelles, en se focalisant sur des caractéristiques bien déterminées de ces informations, ses caractéristiques varient selon les divers systèmes existants. Sur ceux, nous pouvons avoir une liste de mots comme caractéristique, cette liste comporte les mots qui sont fréquemment exploités par les expressions temporelles. Son utilisation a permis d'augmenter le rappel. Nous retrouvons cette caractéristique dans la méthode conditional random fiel, décrite par [Adafre et al, 2005], les « Support Victor Machine ».

D'autres travaux considèrent des systèmes de classifications qui prennent en compte les expressions temporelles selon des classes.

La combinaison des règles et des méthodes d'apprentissage peut être utilisée, elle aussi, par des systèmes, pour faire face aux problèmes liés aux deux approches, ce qui a donné naissance à des systèmes hybrides.

II.3.2.3.Annotation de l'information temporelle

Ancrer des informations temporelles obtenues à partir des textes revient à augmenter l'information temporelle en cas de descriptions temporelles indexicales et vagues.

- **Le système granulaire des entités temporelles**

L'information temporelle obtenue à partir des textes de presse est organisée dans un système granulaire des entités temporelles qui prennent en compte des niveaux de granularité comme le jour, la semaine, le mois et l'année. Les jours sont ancrés par une date, exemple date (1962, 7, 5) sur la ligne du temps. Pour plus d'informations, par exemple, le jour de la semaine, peut également être inclus en ajoutant une information supplémentaire de l'entité de temps. Les Entités de Temps dont le niveau de granularité, par exemple semaines, est représentée sur la base d'intervalle, qui peut être déterminé par un début, qui est une entité de jour et une durée spécifique.

La notion de granularité temporelle se traduit sur le plan linguistique, par exemple, dans l'utilisation des démonstratifs comme déterminants d'expressions temporelles, comme le cas de langue allemande par exemple, un jour précis de semaine fait référence à un jour dans une semaine en cours, donc, il correspond à un niveau de granularité plus vaste. Il existe une relation fonctionnelle unique entre le jour et la semaine, cependant le jour fait partie d'une seule semaine qui peut temporellement chevaucher un ou deux mois. Néanmoins, la semaine est plus fine que le mois dans le système de granularité.

- **Annotation d'informations temporelles selon un format**

Afin d'annoter les informations temporelles, un langage d'annotation est utilisé, nous citons le langage TimeML, qui prend en compte les événements qui n'ont pas été pris en compte par d'autres. Il est aussi utilisé dans l'évaluation des systèmes d'annotation des informations temporelles [laurent, 2011]. Vu les différentes fonctionnalités qu'il offre.

- La détection des expressions temporelles et des événements.
- La normalisation des références datatives.
- Associer des dates aux événements.
- La description des liens entre les événements.

La description de ces fonctionnalités est assurée grâce aux différentes balises spécifiées par le langage.

Les systèmes d'annotation prennent en compte trois classes d'informations temporelles, celles qui dénotent des expressions temporelles comme les dates et les durées, celles qui indiquent les événements et celles qui montrent les relations temporelles.

❖ **Les expressions temporelles**

Les expressions temporelles contenues dans un texte se présentent souvent sous forme d'expressions calendaires, car elles sont ancrées au niveau du calendrier, ce qui conduit à les détecter ensuite leur associer une valeur selon la ligne de temps, ce qui fait émerger des systèmes sémantiques, qui sont utilisés dans le résumé multidocuments, pour gérer l'ordonnancement temporel des données extraites afin de faciliter la compréhension.

Elles sont définies comme étant un élément dans un langage qui lexicalise le concept du temps en matière de reconnaissance et généralement unités temporelles quantifiables. Les expressions temporelles se distinguent par rapport au moment de l'énonciation comme demain, par rapport à un moment de référence comme 5 jours avant, par une date relative comme le 5 juillet, et aussi plus souvent.

Il existe plusieurs catégories d'expressions temporelles qui sont les dates, les heures, les durées, les fréquences et des quantifications temporelles.

1. **Dates :** elles font référence au calendrier grégorien, nous pouvons citer les exemples suivants:
    - Le vendredi 16 janvier 2015 est une date calendaire complète.
    - Mardi est une date partielle
    - Le deuxième semestre est un sous-intervalle calendaire
    - Lundi prochain est une expression déictique
    - Le lendemain est une expression anaphorique.

2. **Heures :** subdivision particulière de la journée, par exemple :
    - 15H est le temps alphanumérique
    - Le matin est la partie de la journée
    - Cette nuit est une expression déictique
    - Le lendemain matin est une expression anaphorique

3. **Durée :** c'est une période prolongée de temps, par exemple :
    - 5ans est une durée unitaire calendaire
    - Une demi-heure est une durée unitaire d'horloge.

4. **Fréquence :** la régularité ou la réapparition d'une éventualité par exemple :
    - Une fois, qui est une fréquence simple
    - Une fois par an, qui est une fréquence pendant les temps.

5. **Quantification temporelles :** c'est la quantification d'une unité temporelle (période ou durée) par exemple :
    - Tous les jours, qui est une expression simple.
    - Certains mois de l'année, qui est une expression complexe.

Les expressions temporelles décrites par time ML sont les dates, les durées et agrégats.

- **Les expressions absolues**

Elles correspondent à des dates explicites ou concrètes, elles désignent clairement une zone du calendrier, leur identification est simple contrairement à leur délimitation. L'interprétation des expressions absolues se fait généralement en les positionnant sur un calendrier. Les exemples suivants illustrent ce concept :

- Le 2 janvier, le bureau sera ouvert.
- Le bureau sera ouvert en janvier.

Ces deux exemples font référence à un jour précis et à un mois entier, qui peuvent être placés sur un calendrier.

- **Les expressions relatives**

Une expression relative est normalisée en calculant la valeur effective de sa référence calendaire, par exemple, normaliser une expression telle qu'aujourd'hui consiste à lui attribuer une valeur calendaire, par exemple le 16 janvier 2015. Généralement un format est utilisé pour associer une valeur calendaire, par exemple, le format ISO 8601. Elles sont dites aussi des expressions déictiques, qui représentent des unités linguistiques. Nous prenons les deux exemples suivants pour illustrer les expressions temporelles déictiques :

- Demain, le bureau sera ouvert.
- Le bureau fermera le 30.

L'interprétation des deux expressions nécessite de savoir le moment de l'énonciation, pour le premier exemple, nous devons connaitre la date d'aujourd'hui, et pour le deuxième le mois et l'année en cours.

La différence entre les expressions absolues et les expressions relatives permet de faire une classification des expressions de type calendaire uniquement, et non les autres types d'expressions.

- **Les durées**

L'annotation des durées se fait en les considérant comme des entités nommées telles que "durant une semaine", et elles sont composées d'une valeur numérique et une unité calendaire cardinale qui permet une appréhension facile de ce type de durée souligné par Time ML, et inversement la notion de cohérence est perdue.

- **Les itératifs**

Les itératifs sont considérées comme des fréquences telles que "Une fois par semaine", un autre point de vue les considère comme un ensemble d'expressions ayant un ancrage multiple sur le calendrier. Nous observons quatre classes d'itérateurs :

- Les itérateurs calendaires (tous les samedi)
- Les itérateurs fréquentiels (souvent)
- Les itérateurs quantificationnels (4 fois)
- Les itérateurs événementiels (Lorsque Said travaille, il stationne d'abord la voiture au garage).

- **Les expressions composées**

Les expressions composées sont de deux natures, les expressions complexes et les expressions juxtaposées, qui peuvent être elles-mêmes unifiées ou imbriquées. La séparation de ces expressions repose sur deux critères, la sémantique et la syntaxe.

- **Les événements**

Les événements véhiculés par des verbes sont plus faciles à détecter et à être relié aux informations temporelles, alors que les événements véhiculés par des noms indiquent son importance. Les événements peuvent être :

- Duratifs ou ponctuels
- Accomplis ou en cours
- Factuels ou non

Ils marquent ainsi une influence sur l'ordre des événements. Les événements véhiculés par des noms peuvent être classés en trois groupes :

- Un verbe peut indiquer l'action de l'événement, ce qui est désigné par des noms déverbaux.

- D'autres événements sont construits à partir des noms qui ne sont pas déverbaux mais ambigus.
- Un autre moyen d'exprimer les événements est les syntagmes nominaux, qui n'ont pas de valeurs événementielles, mais font référence à un événement qui mentionné par un lieu ou une date.

Les événements peuvent être duratifs ou ponctuels, accomplis ou en cours, factuels ou non (ou peut-être). Time ML estime l'annotation des verbes conjugués, des adjectifs et des noms concernant des événements ou des états, en leur attribuant des propriétés constitue l'événement.

❖ **les relations entre événements**

Il existe des relations entre les événements qui sont introduit par des verbes, leur annotation peut être faite à l'aide de la démarche décrite par [katz et al, 2005] ; [setzer,2001], qui adopte une méthode symbolique et utilise une annotation humaine souvent difficile à réaliser suite aux problèmes rencontrés. Des auteurs ont proposé une automatisation de cette annotation pour diminuer certains problèmes, car les annotateurs humains peuvent négliger certaines relations, en ne les prenant pas en compte, c'est pourquoi ils proposent une annotation automatique en considérant un lien entre les paires d'événements. Mais ce qui conduira à une tâche lourde, ce qui les amène à faire un choix sur les relations prise en compte, et appliquer le formalisme proposé par Allen.

II.3.2.4. Méthodes d'extraction

a. Méthode à base de règle

Cette méthode utilise des règles et des ressources linguistiques [strotgen et al, 2010] afin d'assurer les taches de repérage et d'extraction des informations spécifiques. Elle est basée généralement sur des patrons d'expressions régulières afin d'assurer l'extraction. Un prétraitement est effectué qui consiste à annoter les informations temporelles selon un langage d'annotation et permettant ainsi une attribution de valeurs qui correspond à chaque type d'information considéré.

Dans cette méthode, la tâche de normalisation est supervisée d'une manière assez simple. La tâche d'extraction est accompagnée d'une tâche de normalisation, qui, les deux assurent un bon traitement.

a.1.Description du fonctionnement

Dans ce qui suit, une description du fonctionnement de la méthode est détaillée, ainsi que les différentes tâches de traitement.

Le traitement pipeline de documents existants peut être en mesure d'intégrer un annotateur temporel. C'est une extension de l'annotateur temporel utilisée pour l'extraction et l'exploration de l'information spatio-temporelle dans le document.

Le traitement des contenus non structurés tels que l'audio, les images ou les textes est souvent manipulé par UIMA (Unstructured Information Management Architecture) (voir annexe A). Un pipeline d'outils modulaire peut être créé en combinant les différents éléments. La structure d'analyse commune (CAS) est utilisée par tous les composants. En général, un pipeline UIMA est composé de trois types de composants :

- Une collection pour les lecteurs, afin d'accéder aux documents, provenant d'une source et initialiser un objet CAS pour chaque document.

- Les moteurs d'analyse qui assurent une analyse des documents et ajoutent des annotations aux objets CAS.

- Les consommateurs CAS qui assurent le traitement final, comme la mémorisation de l'information annotée dans une base de données.

Le processus de conception contient le lecteur tempeval2, pour une lecture des données spécifiques, initialiser un objet CAS pour chaque document textuel ainsi que l'ajout des données annotées au CAS. Cette information spécifique aux phrases, token et étiquette attribuée par POS (part of speech) (voir annexe C)est utilisée afin d'assurer la tâche d'extraction et la tâche de normalisation des expressions temporelles présentes dans les documents. Un autre fichier est utilisé, le fichier d'écriture tempoeval2 du consommateur CAS. Il permet de créer les fichiers nécessaires qui seront utilisés dans l'étape d'évaluation.

Lors de l'élaboration des règles, l'évaluateur tempeval2 du consommateur CAS est utilisé, il s'agit de faire une comparaison entre les informations annotées selon le standard

timex3 et celles qui sont extraites. Cette comparaison donne naissance à deux listes qui seront utilisées pour la gestion des règles.

a.2.Les tâches d'extraction et de normalisation

Dans cette approche, chaque expression temporelle peut être représentée par un tuple de trois composants sous la forme suivante : te=<ei ;ti ;vi> avec :

- Ei : représente l'expression temporelle.
- Ti : représente le type de l'expression, il y a quatre types possibles à savoir la date, l'heure, la durée et les ensembles.
- Vi : représente la valeur normalisée, cette valeur représente une sémantique temporelle associée à une expression, comme celle décrite par le langage de spécification timeml.

Le processus consiste alors à extraire chaque expression temporelle ei, et lui affecter le type de valeur correspondant ainsi que la valeur normalisée adéquate. Pour ce faire l'utilisation des règles construites manuellement s'avère nécessaire. Ces règles sont regroupées selon le type d'expression temporelle considérée. Une règle dans ce cas est un triplet composé de :

- Une règle d'extraction.
- Une fonction de normalisation.
- Type d'information.

Les règles d'extraction représentent des modèles d'expressions régulières. Des ressources sont utilisées pour assurer le bon fonctionnement des tâches d'extraction et de normalisation.

Un exemple de ressources d'extraction et de normalisation concernant le mois, la saison est donnée ci-dessous. Des ressources associées aux jours, mois, saison sous forme d'expression régulière et des ressources de connaissances pour la normalisation des expressions est aussi utilisée.

Expression	reMonth= ‘’(…|june|july…) ‘’
Ressources	reSeason= ‘’(…|summer|…) ‘’

Normalisation	normMonth(‘’june’’)=’’06’’
Functions	normSeason(‘’summer’’)=’’SU’’

L’algorithme décrit permet d’illustrer la façon dont les règles sont utilisées. Au début du traitement, les règles sont appliquées à l’ensemble des phrases du document, ensuite après l'extraction, les informations correspondants au format timex3 sont ajoutées à l’objet CAS.

Une autre procédure est appliquée, celle qui consiste à lever l’ambigüité des valeurs sous spécifiées ainsi que la suppression d’expressions temporelles invalides de CAS.

L’algorithme est décrit ci-dessous

Foreach sentence **in** document
AddDates To CAS(date rules, CAS);
AddTimes To CAS(time rules, CAS);
AddDurations To CAS(dur rules, CAS);
AddSets To CAS(set rules, CAS);
End foreach
Foreach timex3 **in** CAS
DisambiguateValues(CAS);
End foreach
RemoveInvalidsFromCAS(CAS);

a.3. Description du processus

Les expressions temporelles peuvent être décrites soit explicitement, implicitement ou relativement selon leur description textuelle. La tâche d’extraction se déroule de la même façon pour les expressions temporelles, contrairement à la tâche de normalisation qui consiste à attribuer des valeurs de manières différentes, par exemple pour une expression temporelle explicite, l’attribution de valeur se fait directement en utilisant la fonction de normalisation correspondante à la règle.

Prenons l’exemple d’une expression explicite : 15 février 2010. Cette expression est extraite avec la règle date r1, qui contient les ressources concernant le mois, le jour et l’année, qui se présentent respectivement sous forme : remonth, reday et re fullyea

Expression temporelle explicite

Date-r1=$(reMonth)_{g1}$ $(reDay)_{g2}$, $(refullyear)_{g3}$

Norm-r1(g1,g2,g3)=g3-normMonth(g1)-normDay(g2)

Les groupes sont utilisés pour consulter les tokens associés aux mois, jours et année.

Pour une expression temporelle implicite, la sémantique temporelle est connue, ce qui amène ensuite à attribuer une valeur à cette expression implicite. Un exemple est donné ci-dessous:

Expression temporelle implicite

Date r2 = (reHoliday)g1 (ref ullyear)g2

Norm r2(g1,g2) = g2-normHoliday(g1)

L'expression holiday peut être extraite par la règle date r2, en utilisant la ressource re holiday. La normalisation de cette expression est indiquée par une ressource de connaissance. Par exemple "Independence de l'Algérie", la valeur 05-07-1962 est affectée.

Pour une expression temporelle relative, la tâche de normalisation est beaucoup plus difficile, dans ce cas, des valeurs dans un format sous spécifié sont assignées aux expressions, en fonction du temps de référence. Afin de lever l'ambigüité des attributs sous spécifiés, un post-traitement est fait.

Pour une valeur qui commence par undef-ref, la date mentionnée précédemment est utilisée pour la désambiguïsation, ou bien le temps de création de document. Pour désambiguïser une valeur de la forme undef-last march, il faut calculer le mois avant la date de création du document.

Une autre valeur plus complexe, c'est la valeur sous spécifiée de forme undef-march. Dans ce cas la connaissance linguistique est utilisée pour lever l'ambigüité.

a.4.Evaluation

Afin d'assurer une évaluation de cette méthode, deux ensembles de règles ont été construits. Le premier ensemble se compose de 43 règles, 25 pour les dates, 6 pour les temps,

les durées et les ensembles. Des meilleurs de résultats sont obtenus pour la précision par rapport au rappel.

b. La méthode basée sur les intervalles de temps

Cette méthode [Ling et al., 2010] permet l'identification des relations telles que : événements-temps et événement-événement. Elle utilise un ensemble restreint d'intervalles de comparaison d'Allen.

b.1. Description du processus

Le processus considère un corpus de texte de langage naturel, un ensemble de sortie d'éléments temporels E et des contraintes temporelles C. Le processus considère que chaque élément temporel doit désigner un événement ou un temps, qui les regroupe sous un même nom.

Une référence temporelle comme 1880 peut être vue comme un événement, ce qui dénote un intervalle de temps. Pour chaque élément temporel, un point de début et un point de fin sont associés. Le point de début est marqué par $^{\triangleleft}e$ et le point de fin est marqué par $e^{\triangleright}$.

L'objectif est de produire un ensemble maximal d'événement ayant le plus petit ensemble de contraintes temporelles. Concernant leur évaluation, elle se fait sur la base de l'exactitude des contraintes temporelles, et le nombre des événements ayant des bornes.

Pour la tâche de reconnaissance des relations temporelles, deux étapes successives sont suivies :

- L'extraction des événements et l'identification des expressions temporelles. L'utilisation d'un analyseur syntaxique et un rôle sémantique afin de créer un ensemble de caractéristiques.

- L'inférence sur un modèle probabiliste, afin de localiser les relations d'inégalités entre les points d'extrémités des éléments extraits.

b.2. La tâche d'extraction

Associer un temps à l'événement représente le traitement de beaucoup de systèmes d'extraction d'informations temporelles. Cette approche est simple et pratique, cependant le

rappel est négligé. La prise en compte des relations événement-événement et en appliquant une transitivité peut engendrer un nombre plus important de contraintes sur les temps des événements.

Le raisonnement de transitivité est appliqué, et une extraction d'un grand nombre d'événements est spécifiée, en utilisant des phrases verbales et des phrases nominales datées.

Dans d'autres cas, les événements sont désignés comme une entrée. Un traitement permet d'assurer la correspondance des pages web, en utilisant des modèles simples pour l'extraction des dates, en se basant sur le raisonnement temporel flou pour l'extraction.

Afin d'assurer la tâche d'extraction, une procédure est suivie, qui consiste à considérer une séquence de phrases T, ici le résultat souhaité est un ensemble d'éléments temporels E et des contraintes temporelles C.

- Les éléments temporels désignent un événement ou une référence temporelle.
- Les contraintes représentent des inégalités linéaires de la forme p1<=p2+d ou d est la durée et p1 et p2 représentent soit le début d'un élément temporel ou le point final de cet élément.

A la fin de la procédure, un ensemble maximal d'événements est produit, avec un plus petit ensemble de contraintes temporelles.

b.3.Description du traitement

Afin d'analyser les phrases du corpus, l'analyse syntaxique de standford est utilisée. Les phrases du corpus véhiculent aussi des rôles sémantiques qu' ils doivent être détectés.

Afin de trouver tous les événements et les temps dans une phrase, EVITA [Knippen et al., 2005] et GUTIME ont été utilisés. Et pour chaque élément ei, il faut générer des caractéristiques syntaxiques sous forme de relation. Pour ce faire, un modèle probabiliste est utilisé avec les règles de transitivités afin de classer les relations point par point.

Pour une identification des événements ainsi que les expressions du temps dans chaque phrase, l'utilisation d'EVITA et GUTIME s'avère nécessaire. L'analyseur de

standford permet la création d'une analyse dépendante de la phrase. Pour localiser les arguments temporels de verbes, un système d'étiquetage est utilisé. A la fin des caractéristiques émergent :

- Les événements et temps : selon la norme timeml, les événements et le temps reconnus expriment des aspects importants comme des verbes…

- Caractéristiques de dépendance: en plus des caractéristiques sur chaque élément temporel individuel, il est également essentiel d'avoir une bonne caractéristique pour les paires d'éléments (xi;xj). Sachant que les dépendances syntaxiques indiquent fortement les relations temporelles. Par exemple, dans la phrase « Australia has been independent since 1901 ».l'analyseur génère **prep since**(independent, 1901) . La dépendance **prep since** (l'un des quelques 80 jetons produits par l'analyseur) indique que l'indépendance arrive à un certain moment, en 190, et continue d'être vrai par la suite.

Après analyse de chaque phrase du texte, l'arbre d'analyse syntaxique est obtenu ainsi que les dépendances de la phrase dep (w_1,w_2) qui est considérée à son tour. Si w_1 et w_2 font parties des expressions textuelles xi et xj, alors la fonction dep (xi,xj) est construite pour identifier la relation entre xi et xj et elles sont utiles pour prédire l'ordre temporel entre xi et xj. Dans le cas où il n y a pas de dépendance entre les événements, une fonction proximity (e,x) est créé où x représente l'élément le plus proche dans l'arbre d'analyse, pour éviter le cas où un élément temporel ne peut être lié.

SRL considère un argument spécifique comme événement pour chaque verbe identifié. Un ensemble de règles des réseaux logiques markoviens assure l'interprétation en se basant sur la préposition initiale. En effet, un argument commençant par « avant » suppose que le verbe se produit avant l'heure dans l'argument.

Afin d'identifier l'ordre des relations temporelles qui existent entre les éléments, les réseaux logiques de Markov sont utilisés. Ils représentent une extension probabiliste de la logique du premier ordre formelle, donc un ensemble de formules pondérées du premier ordre.

Chaque formule représente un élément dans le réseau avec les poids correspondants. La probabilité associée est :

$P(x)=1/Z \exp(\sum_i w_i n_i(x))$ où

- Z est la constante de normalisation.
- wi est le poids de la ième formule et
- ni (x) est le nombre de fondement satisfaisant de la ieme formule

Pour une classification des relations, le modèle de formule est utilisé, il influence les dépendances syntaxiques sur les relations temporelles entre les arguments.

dep(x, y) →after(point(x), point(y))

srl after(p, q) →after(p, q)

after(p, q) ^ after(q, r) →after(p, q)

- La première formule est au second ordre, car dep représente la proximité ou l'une des 80 dépendances de Standford. Le point désigne soit le point de départ ou celui d'arrivé.

- La deuxième formule inclut l'information temporelle fournie.

- La troisième formule permet de réduire la probabilité d'interprétation non compatible avec la transitivité.

b.4. Evaluation

Afin d'évaluer la méthode, une prise en compte de la précision et le rappel a été observée. Puisque des contraintes sont prises en compte, donc, ces deux mesures sont calculées en effectuant des comparaisons avec les contraintes observées à la sortie. En effet, la précision est calculée de la façon suivante :

$P = \sum_i c_i / = \sum_i p_i$ où

- Ci est le nombre de contraintes correctement prédites pour la iéme phrase.
- Pi est le nombre de prédiction, et la somme est sur toutes les données d'essai.

L'inférence probabiliste est utilisée par cette méthode pour extraire les contraintes sur les paramètres de l'événement-intervalles, en prenant avantage de la transitivité, ainsi que la

notion de l'entropie temporelle, qui est introduite en tant que moyen permettant de visualiser le rappel d'un extracteur temporel avec des limites temporelles induites.

II.4. Détection et extraction d'informations temporelles dans les entrepôts de données

L'extraction d'informations prend en compte des caractéristiques selon l'information que l'on souhaite extraire, ainsi que les contraintes concernant le type de ces informations, des objectifs bien déterminés et les méthodes utilisées. Cette extraction se fait selon des étapes qui rendent l'extraction efficace et complète, ces étapes ont été mentionnées dans plusieurs conférences autour de l'extraction, qui mettent en évidence donc à chaque extraction, la prise en compte des informations à extraire, et pour cela, un marquage des informations temporelles est indispensable qui consiste à annoter et à reconnaître des expressions temporelles afin de baliser l'ensemble des expressions temporelles présentes dans le document, et d'attribuer une valeur aux expressions temporelles, cette étape est conjointement liée à la première. Elle consiste à donner un sens à la référence temporelle. Après avoir délimité les informations que l'on souhaite extraire, il faut associer à ces informations une valeur qui permettra de leur donner une signification, et pour cela un format est utilisé.

La détection et l'extraction d'informations temporelles permettent de cerner les informations temporelles disponibles dans les textes. Cependant, pour une prise de décision sur ces informations extraites, une extraction dans les entrepôts de données est meilleure, elle représente le processus par lequel les données opérationnelles sont transférées vers l'entrepôt. Le schéma de l'entrepôt peut être considéré comme une vue sur les données opérationnelles, on parle de vue matérialisée puisque les relations correspondants à la vue sont effectivement créées. Il est intéressant, de pouvoir mettre à jour les données de l'entrepôt sans régénérer la totalité de celui-ci , ce qui nous ramène à une mise à jour incrémentale.

L'extraction des informations d'un entrepôt se fait selon deux approches principales :

- Les requêtes en SQL, ce qui se traduit par ROLAP (relational on-line analytical processing).

- L'entrepôt considéré comme une base de données multidimensionnelle, dont le modèle est un cube à n-dimensions, ce qui se traduit par MOLAP (multidimensional on-line analytical processing).

II.5. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur la détection et l'extraction d'informations temporelles, en mettant l'accent sur les méthodes et techniques d'extraction. Nous avons présenté au cours de ce chapitre le fonctionnement de deux méthodes d'extraction d'informations temporelles, l'une basée sur les règles et l'autre basée sur les intervalles de temps, ainsi que les résultats donnés par chacune de ces deux méthodes.

Cependant, l'extraction d'informations temporelles s'articule généralement en deux phases: le repérage ou l'annotation des expressions et leur interprétation et reformulation en une valeur normalisée. Ces deux étapes présentent chacune à sa manière les expressions temporelles.

Dans le schéma de fonctionnement d'un entrepôt de données, les sources peuvent contenir des documents contenant des informations temporelles. Cependant, lors de l'extraction des données à partir des sources, les informations temporelles peuvent ne pas être prise en compte par la tâche d'extraction, et l'entrepôt de données issu contiendra des données de types différents, et peut ne pas contenir le type de données souhaité. Pour faire usage de ces informations, des outils temporels sont appliqués aux documents afin d'extraire les données temporelles contenues dans les documents, puis ces données temporelles extraites doivent être regroupées pour pouvoir être exploitées. C'est à ce niveau que se situe notre problématique, qui consiste à proposer une architecture d'un entrepôt de données intégrant le dictionnaire d'annotation de données temporelles, qui sera présentée dans le chapitre suivant.

Chapitre III

Proposition d'un système d'entrepôt de données intégrant un dictionnaire d'annotation temporelle

III.1. Introduction

La construction d'un entrepôt de données peut être considérée comme une intégration matérialisée par des concepts multidimensionnels. Ces données sont issues de sources différentes, c'est pourquoi elles doivent être intégrées. Cependant, afin d'extraire des données, des critères sont pris en compte d'où la nécessité de définir un format d'annotation pour assurer la reconnaissance de ce type d'information.

La recherche d'un type de donnée au sein d'un document ou plusieurs documents, consiste à parcourir le document et à extraire le type de données considéré. Afin de faciliter la recherche de l'information temporelle, la construction du dictionnaire d'annotation des données temporelles est utile. En effet, la recherche de l'information ne se fera plus au niveau du document mais plutôt au niveau du dictionnaire, et de retourner aussi le texte ou le document correspondant.

Nous proposons un système d'un entrepôt de données intégrant le dictionnaire d'annotation de données temporelles.

III.2. Architecture du système

Nous nous sommes basés sur une architecture de fonctionnement d'un entrepôt de données, auquel nous proposons des modules afin d'assurer l'extraction d'informations temporelles au sein de cet entrepôt. Nous proposons l'ajout de :

- Un extracteur de données temporelles vers le dictionnaire d'annotation : qui assure l'extraction des informations temporelles repérées. La détection consiste à repérer les informations temporelles et les représenter sous le format timeml. Nous obtiendrons des informations temporelles balisées, en vue de construire le dictionnaire d'annotation, après avoir parcouru le texte, afin de repérer les différents types d'informations temporelles considérés.

- Dictionnaire d'annotation de données temporelles sous le format timeml : qui comporte les différents types d'informations temporelles annotées selon le format considéré timeml.

- Un moteur de recherche : qui assure la recherche de l'information temporelle dans le dictionnaire d'annotation et permet de retourner le texte ou le document correspondant et d'accéder aux sources.

Nous proposons l'architecture suivante :

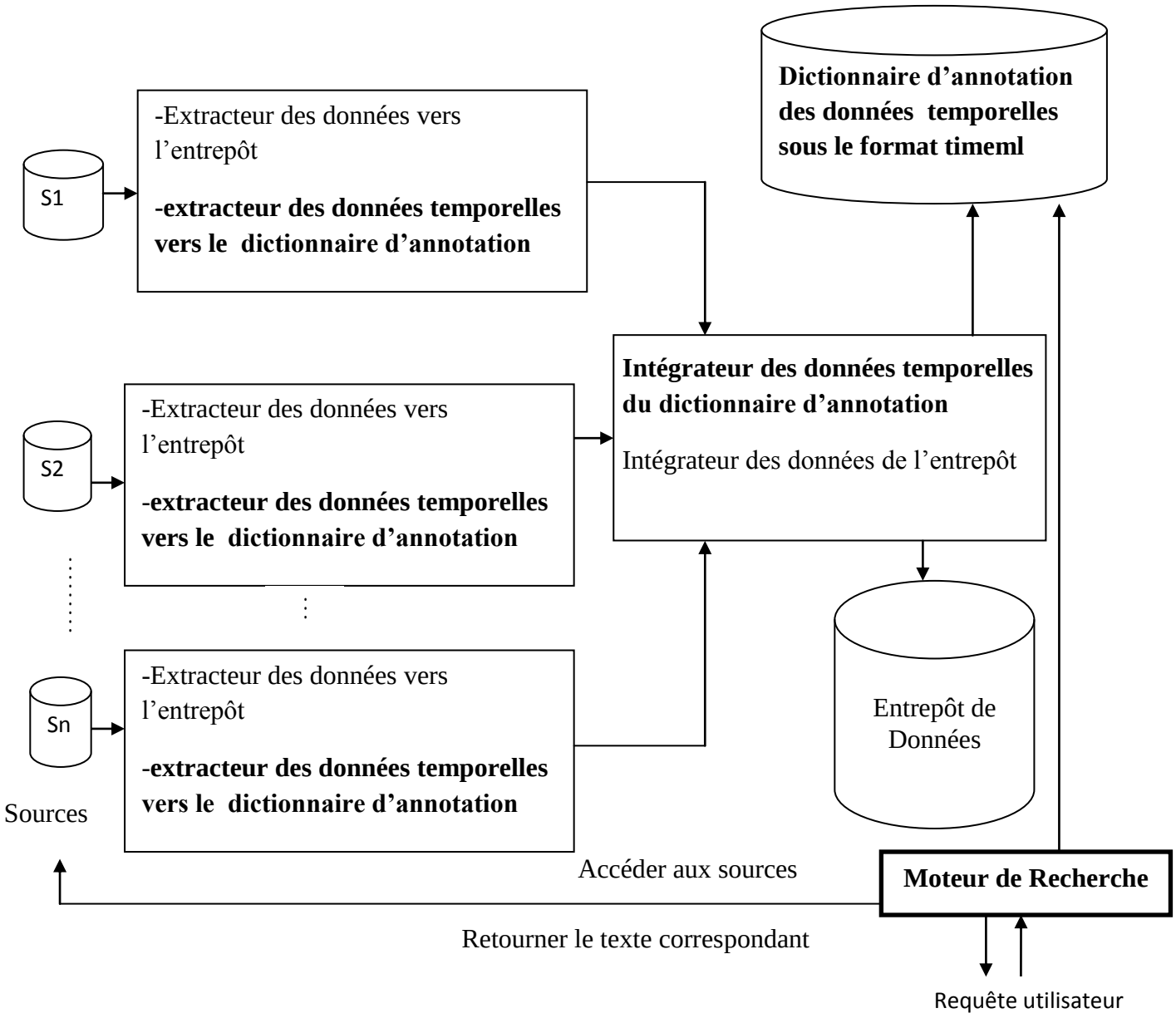


Figure III.1. Architecture du système proposé

L'extracteur des données temporelles vers le dictionnaire d'annotation assure l'extraction des informations temporelles, détectées et annotées sous le format timeml. Pour ce faire, nous nous sommes basés sur des textes annotés, générés par l'outil heideltime [strotgen et al, 2010], qui à partir d'un texte, génère un texte annoté après avoir appliqué les traitements nécessaires à base d'expressions régulières, qui expriment chacune d'elles les différents types d'informations temporelles. Le texte issu de l'outil heideltime contient des balises timex3, des différents types d'informations temporelles présentes dans le texte.

A partir du texte annoté sous le format timeml, la tâche d'extraction est effectuée afin de construire le dictionnaire d'annotation des données temporelles sous le format timeml. Cette tâche consiste à extraire les informations temporelles annotées.

Nous proposons l'algorithme suivant :

Pour chaque texte annoté du document faire
Pour chaque phrase du texte annoté faire
Pour chaque balise de la phrase faire
Si le début de la balise est **<timex3** alors
Récupérer la suite de la balise jusqu'à la fin de la balise </**timex3**>
Fin si
Fin pour
Fin pour
Fin pour

Cet algorithme décrit l'extraction des informations temporelles annotées, issues de la tâche de détection.

Le dictionnaire d'annotation des données temporelles est issu de la détection et l'extraction des informations temporelles présentes dans les entrepôts de données qui est assurée grâce l'extracteur des données temporelles vers le dictionnaire d'annotation. Il consiste à extraire les balises représentées sous le format timeml, prenant en compte les quatre types d'informations temporelles : les dates, les durées, les heures et les ensembles.

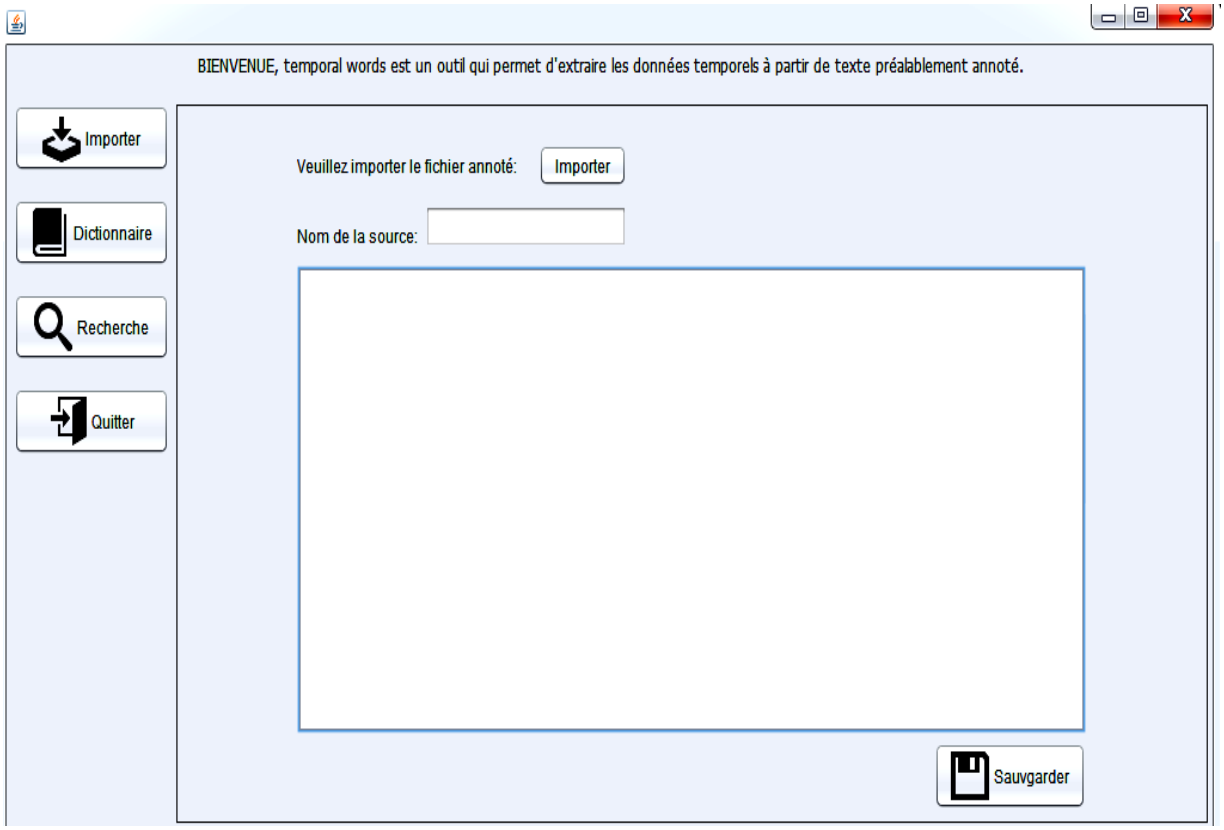
Ce dictionnaire représente la base de données temporelle, qui contient toutes les informations temporelles extraites des textes annotés. Il sera utilisé pour effectuer la recherche d'une information temporelle, et de retourner le texte ou le document correspondant.

III.3. Etude de faisabilité

L'objectif est d'étudier la faisabilité de notre proposition décrite ci-dessus, par la réalisation d'un prototype implémentant un dictionnaire d'annotation des données temporelles, et son utilisation.

Pour ce faire, nous avons utilisé l'environnement de développement intégré NetBeans IDE 8.2.0 pour la création de notre interface. Nous avons utilisé aussi mysql pour la création et la gestion de la base de données.

L'interface de notre outil est décrite ci-dessous :



FigureIII.2. Interface de l'outil proposé

L'interface de notre outil comporte les boutons suivants :

- Le bouton importer : qui consiste à importer le fichier annoté.
- Le bouton sauvegarder : qui consiste à sauvegarder le ficher annoté après l'avoir importé.
- Le bouton dictionnaire : qui consiste à afficher le dictionnaire d'annotation.
- Le bouton recherche : qui consiste à rechercher une information temporelle présente dans le dictionnaire d'annotation.

- Le bouton quitter : qui consiste à fermer l'outil.

Le fonctionnement de notre outil se fait comme suit :

Nous commençons par importer le fichier annoté, nous prenons l'exemple du texte suivant :

Texte 1

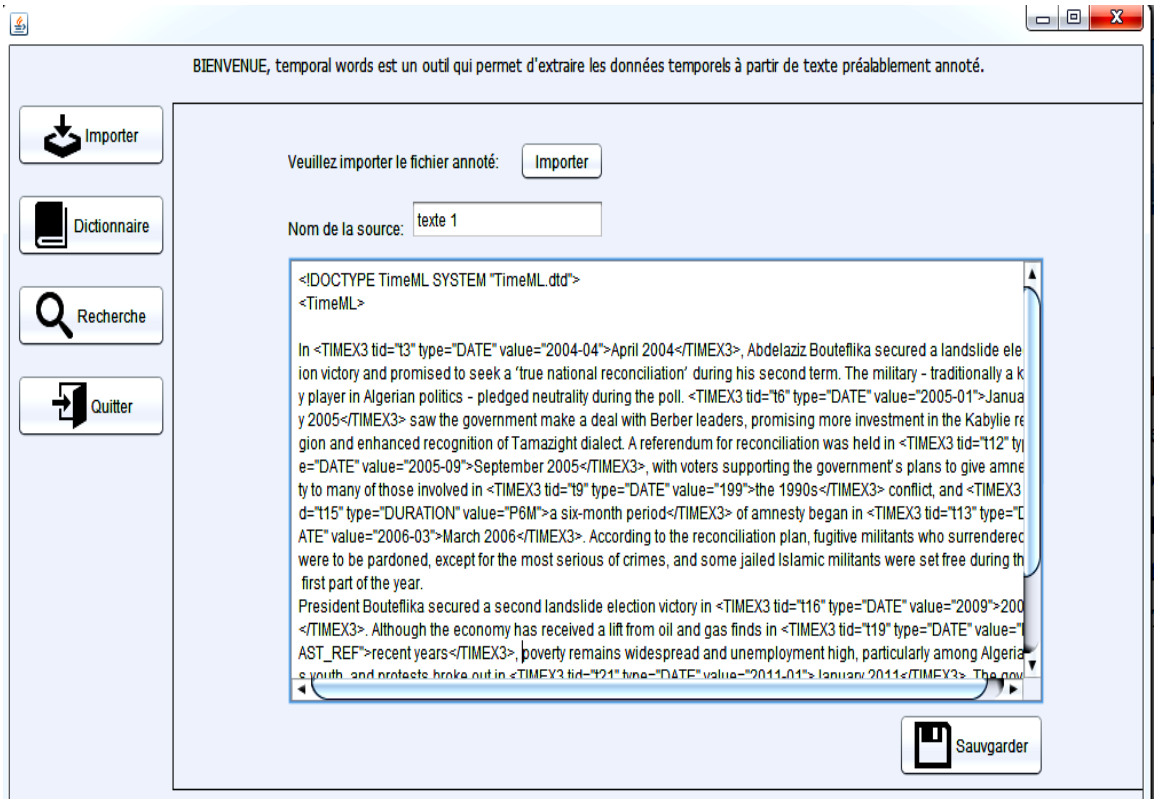
In April 2004, Abdelaziz Bouteflika secured a landslide election victory and promised to seek a 'true national reconciliation' during his second term. The military – traditionally a key player in Algerian politics – pledged neutrality during the poll. January 2005 saw the government make a deal with Berber leaders, promising more investment in the Kabylie region and enhanced recognition of Tamazight dialect. A referendum for reconciliation was held in September 2005, with voters supporting the government's plans to give amnesty to many of those involved in the 1990s conflict, and a six-month period of amnesty began in March 2006. According to the reconciliation plan, fugitive militants who surrendered were to be pardoned, except for the most serious of crimes, and some jailed Islamic militants were set free during the first part of the year.
President Bouteflika secured a second landslide election victory in 2009. Although the economy has received a lift from oil and gas finds in recent years, poverty remains widespread and unemployment high, particularly among Algeria's youth, and protests broke out in January 2011. The government responded by ordering a reduction in the price of basic foodstuffs, and repealed the 1992 state of emergency law.

Nous avons appliqué l'outil heideltime [strotgen et al, 2010] pour la génération du texte annoté. Nous obtenons le résultat suivant :

<?xml version="1.0"?>
<!DOCTYPE TimeML SYSTEM "TimeML.dtd">
<TimeML>

In <TIMEX3 tid="t3" type="DATE" value="2004-04">April 2004</TIMEX3>, Abdelaziz Bouteflika secured a landslide election victory and promised to seek a 'true national reconciliation' during his second term. The military – traditionally a key player in Algerian politics – pledged neutrality during the poll. <TIMEX3 tid="t6" type="DATE" value="2005-01">January 2005</TIMEX3> saw the government make a deal with Berber leaders, promising more investment in the Kabylie region and enhanced recognition of Tamazight dialect. A referendum for reconciliation was held in <TIMEX3 tid="t12" type="DATE" value="2005-09">September 2005</TIMEX3>, with voters supporting the government's plans to give amnesty to many of those involved in <TIMEX3 tid="t9" type="DATE" value="199">the 1990s</TIMEX3> conflict, and <TIMEX3 tid="t15" type="DURATION" value="P6M">a six-month period</TIMEX3> of amnesty began in <TIMEX3 tid="t13" type="DATE" value="2006-03">March 2006</TIMEX3>. According to the reconciliation plan, fugitive militants who surrendered were to be pardoned, except for the most serious of crimes, and some jailed Islamic militants were set free during the first part of the year.
President Bouteflika secured a second landslide election victory in <TIMEX3 tid="t16" type="DATE" value="2009">2009</TIMEX3>. Although the economy has received a lift from oil and gas finds in <TIMEX3 tid="t19" type="DATE" value="PAST_REF">recent years</TIMEX3>, poverty remains widespread and unemployment high, particularly among Algeria's youth, and protests broke out in <TIMEX3 tid="t21" type="DATE" value="2011-01">January 2011</TIMEX3>. The government responded by ordering a reduction in the price of basic foodstuffs, and repealed the <TIMEX3 tid="t23" type="DATE" value="1992">1992</TIMEX3> state of emergency law.
</TimeML>

Ensuite en indiquant le nom de la source, nous obtenons la fenêtre suivante :



FigureIII.3. Chargement du texte annoté

Après avoir importé le texte annoté, nous allons le sauvegarder. Nous obtenons la fenêtre suivante :

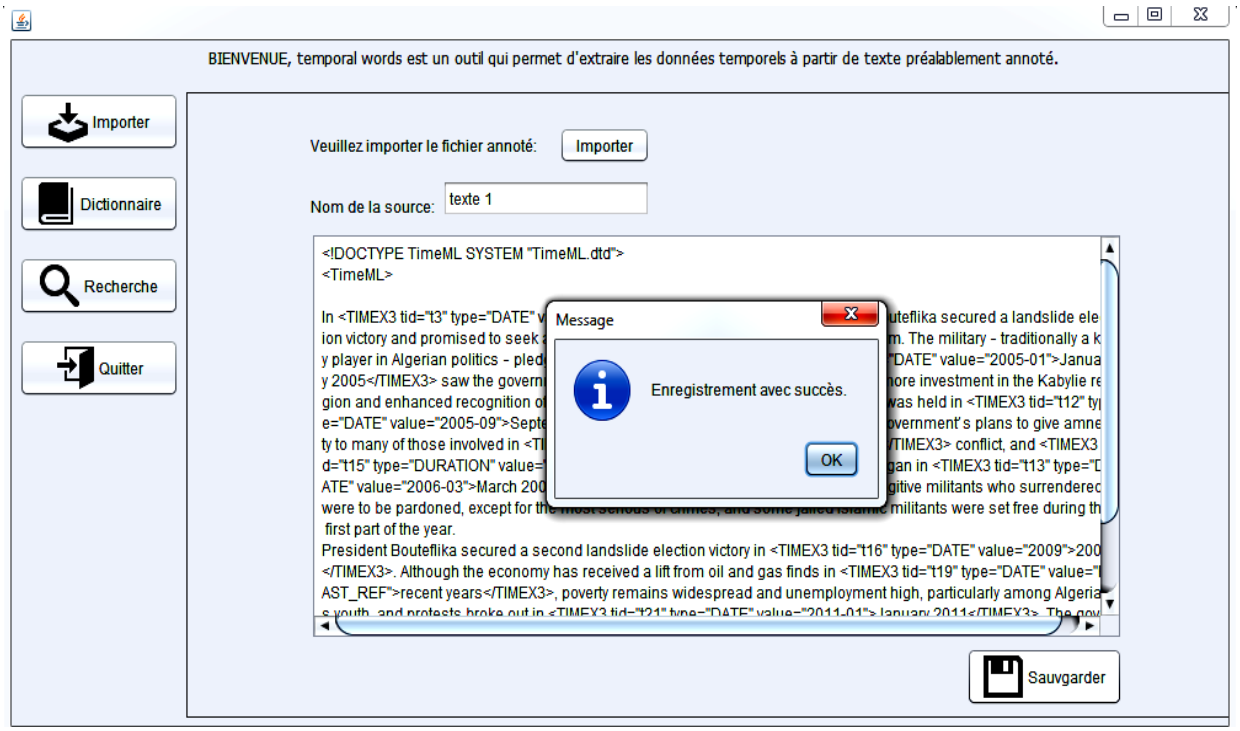


figure III.4. Enregistrement du texte annoté

Un message d'enregistrement avec succès est alors affiché.

Afin de générer le dictionnaire d'annotation de données temporelles, nous utilisons le bouton dictionnaire, on obtient le résultat suivant :

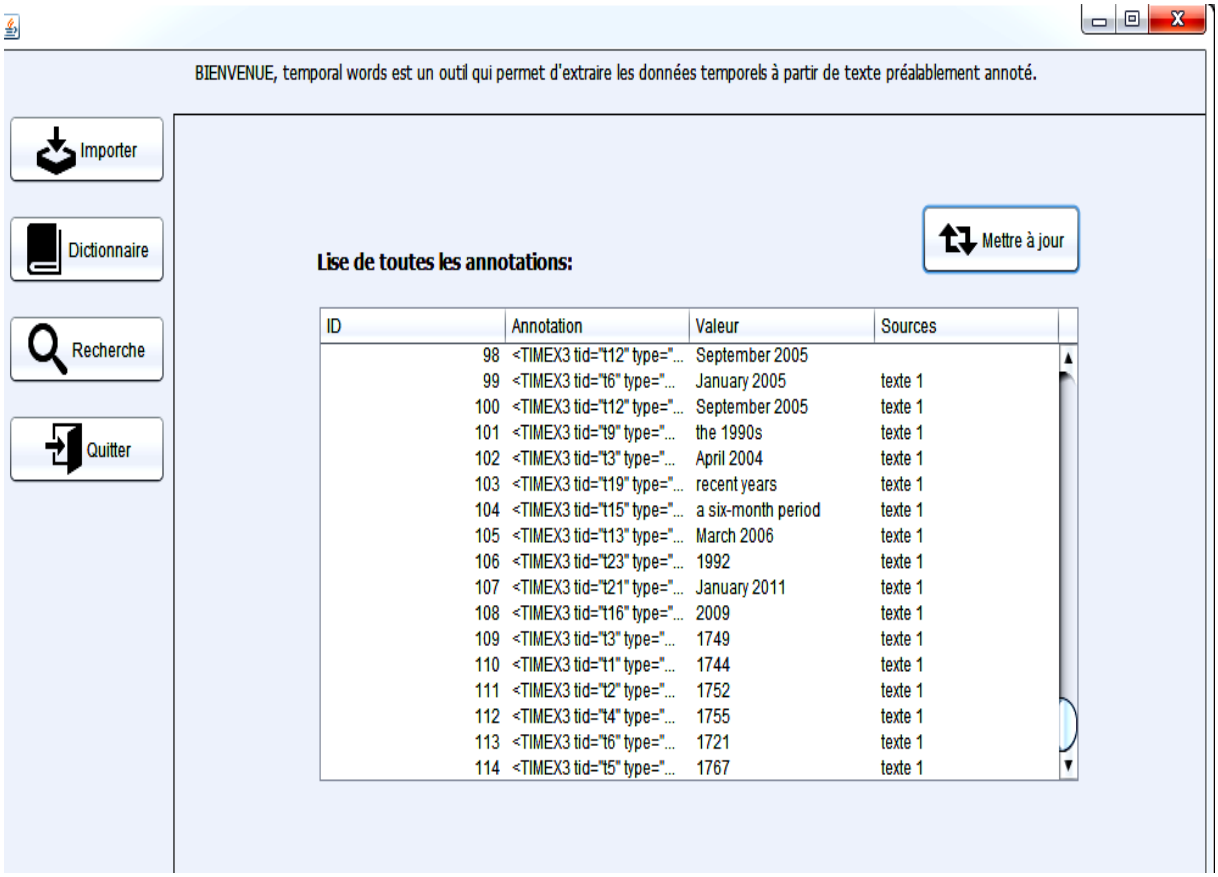


Figure III.5. Dictionnaire d'annotation

Notre dictionnaire d'annotation comporte toutes les informations temporelles décrites par un identifiant, l'annotation selon le format timeml, ainsi que la source utilisée. Le bouton « mettre à jour », consiste à mettre à jour le dictionnaire d'annotation, afin de prendre en compte les nouvelles extractions.

La procédure qui est chargée de cette tâche est donnée ci-dessous:

```
public Dictionnaire() {
    initComponents();
    List<Annotation> annotations = service.getAllAnnotation();
    DefaultTableModel dtm = (DefaultTableModel) tableAnn.getModel()
   annotations.stream().forEach((a) -> {
   dtm.addRow(new Object[]{a.getId(), a.getAnnotation(), a.getValeur(), afficheSources(a.getSources())});
    });
  }
```

Pour effectuer une recherche au niveau du dictionnaire, nous utilisons le bouton « rechercher », qui consiste à rechercher l'information temporelle et à retourner la source correspondante. Nous obtenons la fenêtre suivante :

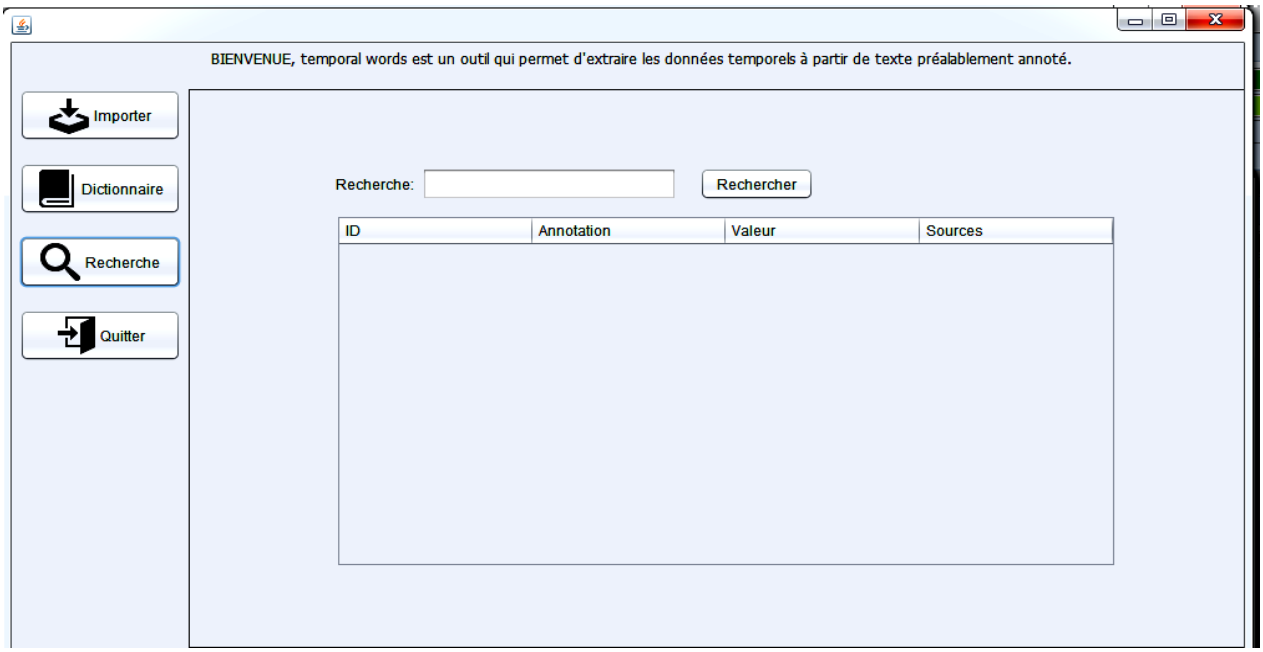


Figure III.6. La recherche dans le dictionnaire

Nous utilisons le champ « recherche » pour lancer notre recherche, dans notre exemple, si l'on veut faire une recherche de l'information temporelle « january 2011 », et de retourner la ou les sources correspondantes, nous obtiendrons le résultat suivant :

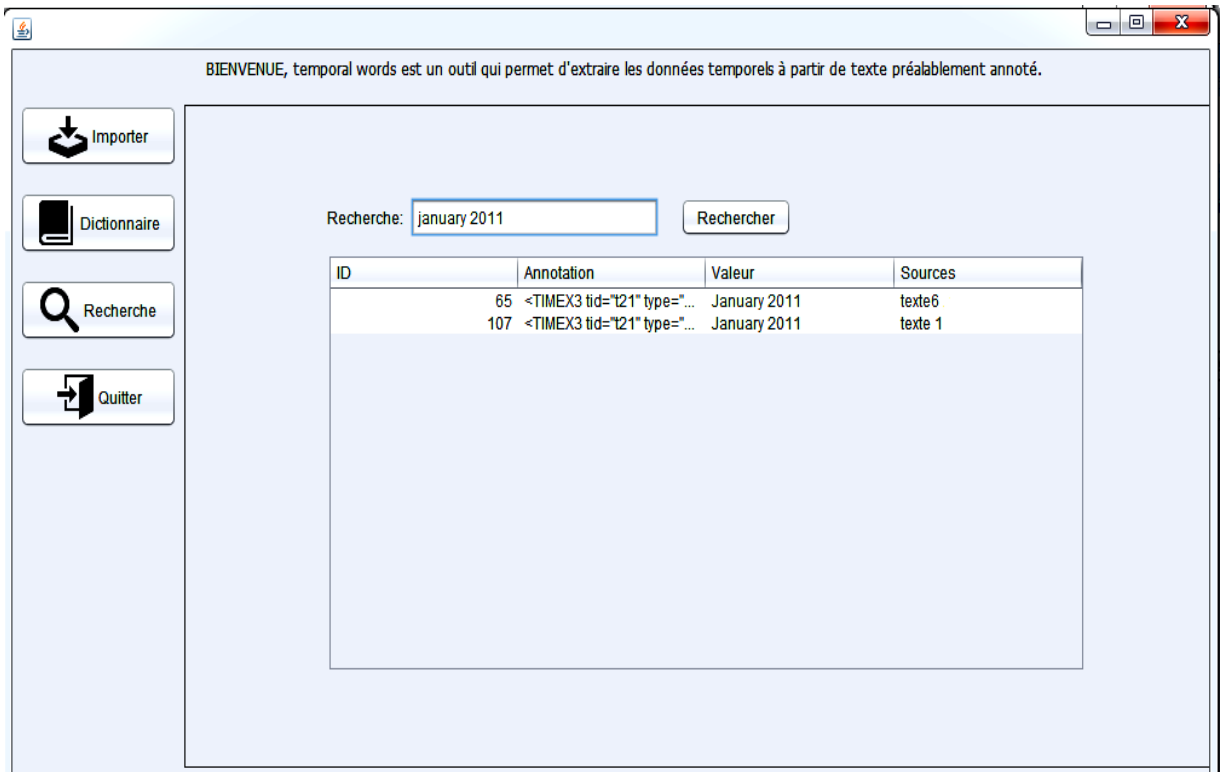


Figure III.7. Résultat de la recherche

Le résultat de la recherche décrit l'information temporelle recherchée ainsi que la source correspondante.

III.4. Conclusion

Nous avons proposé une architecture d'un entrepôt de données intégrant le dictionnaire d'annotations des données temporelles, inspirée de l'architecture de l'entrepôt. Nous avons évalué la faisabilité de notre proposition par la réalisation d'un prototype où:

- La détection des informations temporelles se fait en utilisant l'outil heideltime proposé dans [strotgen et al,2010].
- L'extraction vers le dictionnaire d'annotation.
- L'accès basé sur les informations temporelles en utilisant le dictionnaire d'annotation.

Conclusion générale

Conclusion Générale

Les travaux présentés dans ce mémoire s'inscrivent dans le contexte de l'extraction de l'information temporelle au sein des entrepôts de données. Les informations temporelles intégrées dans les documents sous la forme d'expressions temporelles offrent un moyen intéressant pour faire avancer et améliorer la fonctionnalité des applications de la recherche de l'information. Le traitement de la temporalité est crucial pour la compréhension de textes en langue naturelle, ce qui a motivé le développement d'outils pour le repérage et la normalisation de ce type de donnée ainsi que l'extraction d'informations afin de permettre une tâche complète de l'extraction d'informations temporelles.

Le schéma général du fonctionnement d'un entrepôt de données génère un entrepôt de données qui contient les données issues de la tâche extraction, effectuée sur des sources. Ces données sont de types différents. Afin de générer un entrepôt contenant uniquement un type d'information particulier, la tâche d'extraction doit porter uniquement sur ce type particulier, afin de permettre une meilleure utilisation. C'est dans ce contexte que se situe notre travail, qui consiste à construire le dictionnaire d'annotation de données temporelles qui représente un outil issu de la détection et l'extraction de données temporelles dans les entrepôts de données.

Dans notre travail, nous avons présenté un état de l'art sur les entrepôts de données en mettant l'accent sur les tâches d'extraction ainsi que la construction des entrepôts de données. Nous avons également présenté un état de l'art sur la détection et l'extraction des informations temporelles, en mettant l'accent sur la détection et l'extraction des informations temporelles dans les entrepôts de données. Ensuite, nous avons proposé une architecture d'un entrepôt de données intégrant le dictionnaire d'annotation de données temporelles. Afin d'évaluer la faisabilité de notre proposition, nous avons développé un prototype d'outil de détection et d'extraction d'informations temporelles dans les entrepôts de données. Pour ce faire, nous avons utilisé l'environnement de développement intégré NetBeans IDE 8.2.0 et mysql pour la création et la gestion de la base de données.

Comme perspective, nous pouvons prendre en considération un intervalle plus large des informations temporelles, ce qui nous ramène à une extension des types d'informations temporelles pris en compte dans l'étape d'annotation . Nous pouvons considérer par exemple

les événements qui eux aussi expriment la temporalité, annotés sous le format timeml. Utiliser aussi le dictionnaire d'annotation dans des cadres applicatifs.

Références Bibliographiques

[Adafre et al., 2005] S-F.Adafre, D. AHN, M. de Rijke. *Extracting temporal information from open domain text : A comparative exploration*. Journal of Digital Information Management, 3(1):14–20, 2005.

[Allen et al., 2010] J. Allen, N. Uzzaman. *TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text*. In Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval '10), pages 276–283, 2010.

[Bellatreche, 2000]. L. Bellatreche. « Utilisation des index et de la fragmentation dans la conception logique et physique d"un entrepôt de données » .Thèse pour obtenir le grade de Docteur en Informatique, université de Clermond Ferrand II, 2000.

[Bittar, 2008] A .Bittar. *Annotation des informations temporelles dans des textes en français*. In Actes de la 12e édition de RECITAL, pages 11–20, Avignon, France, 2008.

[Bittar, 2009] A. Bittar. *Annotation temporelle de textes en français*. Séminaire de linguistique de l'Université de Marne-la-Vallée, 2009.

[borillo et al,. 2004] A.borillo., M.Bras, A.Ledraoulec, L.Vieu, A.Molendijk, H. deswart, H.Verkuyl, C.Vet, C.Vetters. *Tense, connectives and discourse structure*. *In* CORBLIN, F. de SWART, H., éditeurs : *Handbook of French Semantics*, CSLI Lecture Notes, pages 309–348. CSLI Publications, Standford, 2004.

[Golfarelli , 2009]. Golfarelli. "From user requirements to conceptual design in data warehouse design – a survey", 2009.

[Habel et al, 2001] C. Habel et H. Schilder. *From temporal expressions to temporal information : Semantic tagging of news message.* In Proceedings of ACL'01 workshop on temporal and spatial information processing, pages 65–72, Toulouse, France. Association for Computational Linguistics, 2001.

[Katz et al., 2005] G.Katz et I.Mani. *The specification language TimeML*. In I. Mani, J. Pustejovsky et R. Gaizauskas, Eds., The Language of Time : A Reader. Oxford University Press, 2005.

`

[Knippen et al., 2005] R. Knippen, R.Saurí., M. Verhagen and J. Pustejovsky. *Evita : A Robust Event Recognizer for QA Systems.* In Proceedings of HLT/EMNLP 2005, p. 700–707, 2005.

[Laurent, 2011] Laurent Kevers. *Accès sémantique aux bases de données documentaires :Techniques symboliques de traitement automatique du langage pour l'indexation thématique et l'extraction d'informations temporelles*. Thèse de doctorat en en Langues et lettres, université catholique de louvain, louvain-la-neuve, 2011.

[Ling et al., 2010] X.Ling et D-S.Weld. *Temporal Information Extraction*. Proc. the Twenty-Fourth Conference on Artificial Intelligence , AAAI 2010.

[Métais et al; 2002] E. Métais, F.Sédes. appariement d'informations dans les entrepôts de données: quelques approches pour lefiltrage flexible.

[Pustejovsky et al., 2009] J. Pustejovsky et M.Verhagen. *SemEval-2010 Task 13: Evaluating Events, Time Expressions, and Temporal Relations (TempEval-2).* In Proceedings of the Workshop on Semantic Evaluations, pages 112–116. ACL, 2009.

[Schmid, 1994] H.Schmid. *Probabilistic Part-of-Speech tagging using decision trees*. In International Conference on New Methods in Language Processing, Manchester, UK. 1994

[selma khouri, 2009]. "modélisation conceptuelle à base ontologique d'un entrepôt de données". thése pour obtenir le grade de magister en informatique, ecole nationale superieure d'informatique, 2009

[Setzer, 2001] A.Setzer. *Temporal Information in Newswire Articles : An Annotation Scheme and Corpus Study*. PhD thesis, University of Sheffield , 2001.

[strotgen et al,2010] J. Strotgen et M. Gertz. HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions. Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 321–324, Uppsala, Sweden, 15-16 July 2010

`

[Teissedre, 2012] C. Teissedre. *Analyse sémantique automatique des adverbiaux de localisation temporelle : application à la recherche d'information et à l'acquisition de connaissances.* Thèse de doctorat en sciences du langage et traitement automatique des langues, université paris ouest-nanterre la défense, 2012.

[Turenne, 2004] N. Turenne. *Apport de l'extraction d'information temporelle à la modélisation des réseaux biologiques*. INRA - Unité Mathématique, Informatique et Génome (MIG), 2004.

[WEISER, 2010] S. WEISER. *Repérage et typage d'expressions temporelles pour l'annotation sémantique automatique de pages Web*. Application au e-tourisme. Thèse de doctorat, Université Paris Ouest Nanterre La Défense, Paris, France, 2010.

Annexes

Annexe A

Unstructured Information Management Architecture (UIMA)
Traitement Automatique du Langage Naturel
(TALN)
Outils d'analyse de données textuelles
Laurent Audibert (LIPN - UMR CNRS 7030)

Apache UIMA :

UIMA n'est pas une plateforme mais une librairie dédiée au développement de chaînes de traitement de l'information non structurée, implémentation initiée par IBM, devenu un projet *Open Source* en incubation à la fondation apache en 2006, Projet Apache de niveau supérieur (*Top level project*) depuis 2010.

Sa spécification en cours de normalisation à l'OASIS 6 (*Organization for the Advancement of Structured Information Standards*), il a pour ambition de s'imposer en tant que norme et standard industriels, *sa* documentation importante et existence de tutoriaux :

L'Apache UIMA propose un framework contenant :
-Ensemble de librairies permettant le développement de composants UIMA
-Ensemble d'outils permettant le déploiement des composants

Il Facilite l'intégration et le déploiement de composant d'annotation. Les différents composants (*Analysis Engine*) s'échangent un ensemble d'annotations déportées (CAS) modélisé par un système de types (*Type Systeme*)

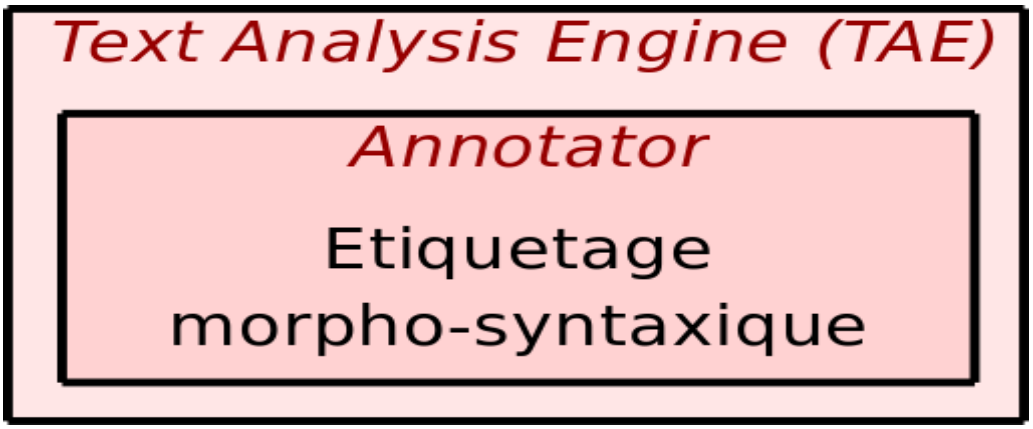
UIMA - *Analysis Engine* (AE)

Analysis Engine (AE) : Composant fondamental de traitement

AE primitif :

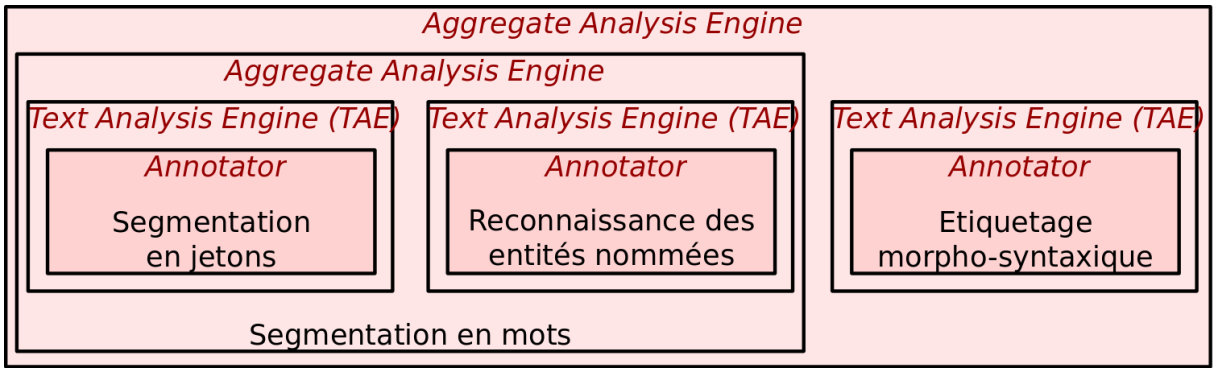
- Partie déclarative (spécifications) en XML
- *Annotator* : implémentation en Java, en C++ ou sous forme de service Internet

Appelé TAE (*Text Analysis Engine*) quand il manipule des documents textuels.



UIMA favorise la réutilisation et l'agrégation de composants en s'appuyant sur leur description XML.

AE complexe : composition ordonnée d'un ensemble d'AE complexes ou primitifs (*Aggregate Analysis Engine*)

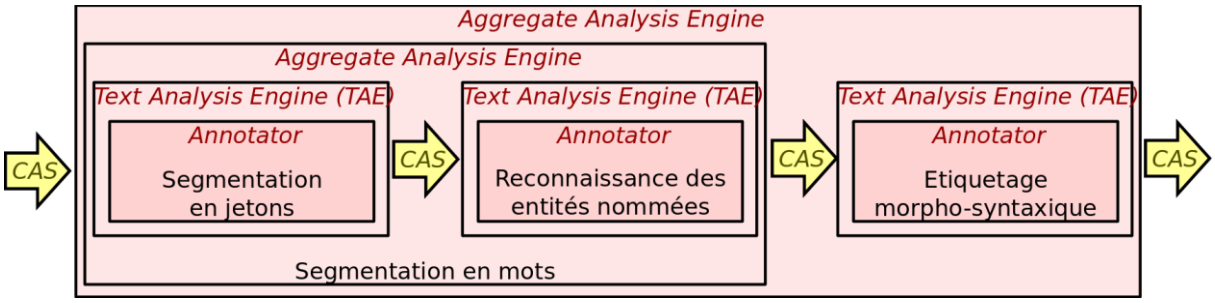


UIMA favorise la réutilisation et l'agrégation de composants en s'appuyant sur leur description XML

UIMA - Common Analysis System (CAS)

Common Analysis System (CAS)

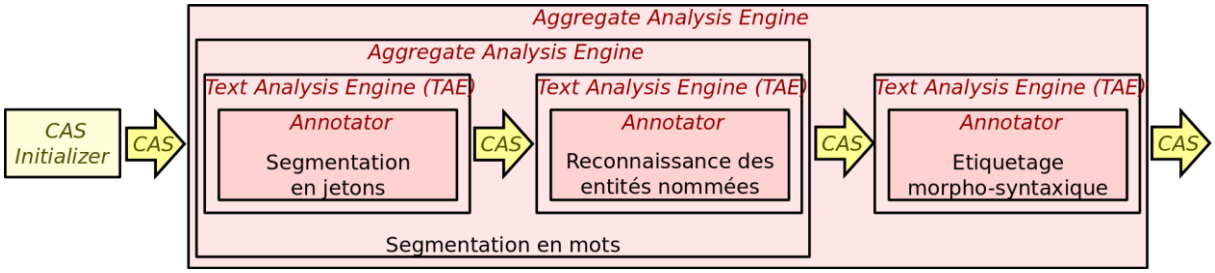
Objet commun aux différents composants contenant le document original, ses méta-données (annotations) et une ou plusieurs interfaces pour accéder aux données, Un TAE ne fait que compléter un CAS, Pour plus de flexibilité, les annotations sont déportées



UIMA - *Common Analysis System Initialiser* (CAS)

Common Analysis System Initialiser (CAS)

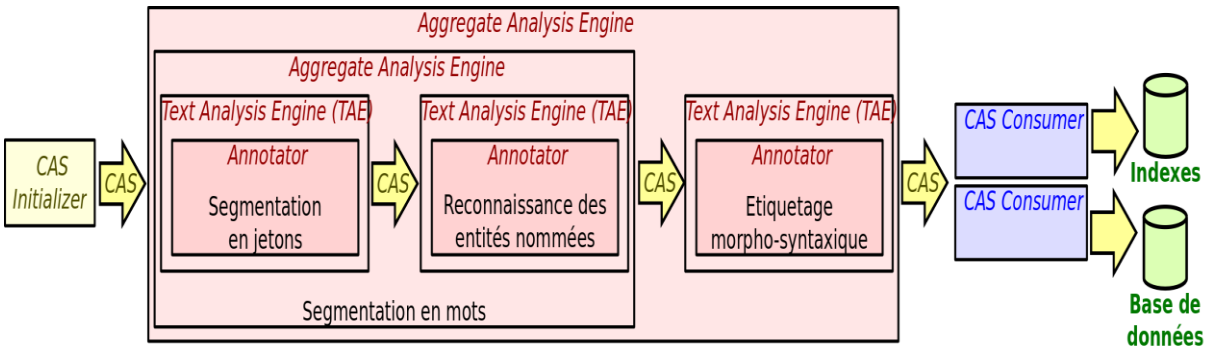
Un *CAS Initialiser* est propre à un format de document source et a pour tâche de produire un objet CAS



UIMA - Common Analysis System Consumer (CAS Consumer)

CAS Consumer :

Intervient à la fin de la chaîne des différents AE pour produire, à partir des CAS, une ressource exploitable par une autre application (index, base de données…), consomment des CAS mais n'en produisent pas, Le rôle peut aller de la simple mémorisation des CAS à des inférences portant sur la totalité des CAS consommés.



UIMA - *Collection Reader*

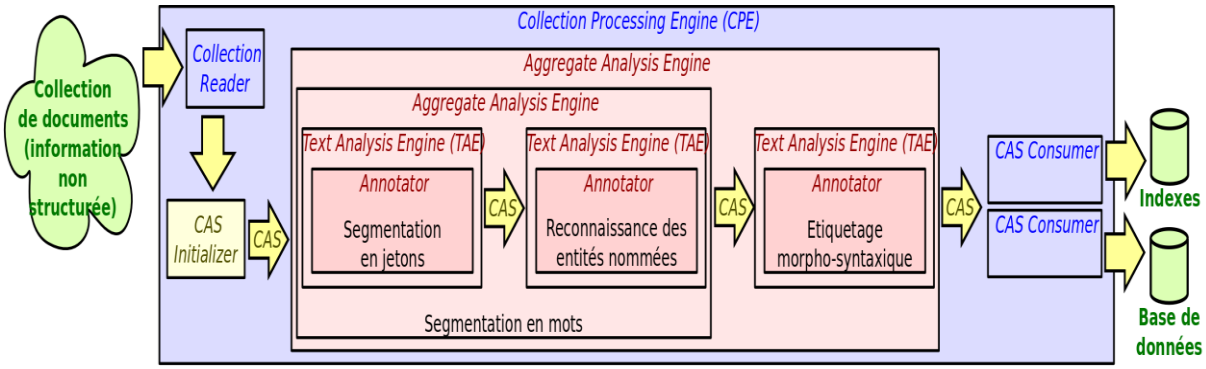
Collection Reader

Itère sur la collection des documents pour alimenter les *CAS Initialiser*. La seule méthode d'un composant *Collection Reader* est « *passer au document suivant* »

UIMA - *Collection Processing Engine* (CPE)

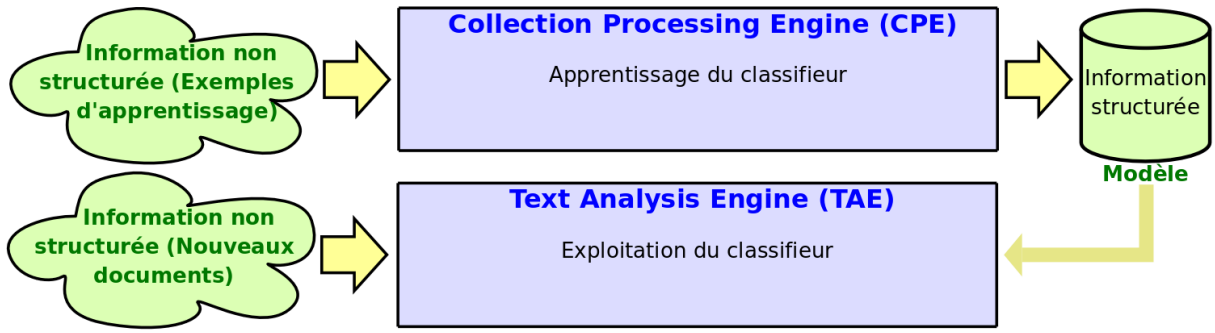
Collection Processing Engine (CPE)

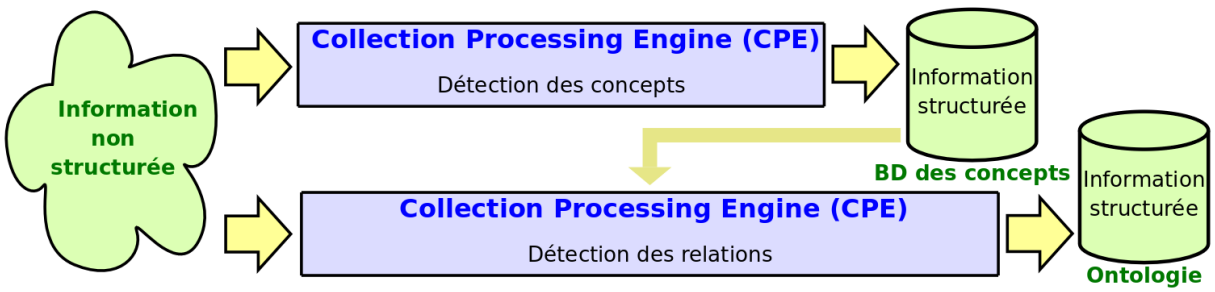
Composant complexe rassemblant tous les composants participant au traitement du *Collection Reader* jusqu'aux *CAS Consumer*, contrôle le flux entre ses différents composants.



UIMA - Utilisation de ressources d'informations structurées :

UIMA permet à un AE d'accéder à une ressource d'informations structurées,





UIMA - *Collection Processing Management* (CPM)

Collection Processing .. Management (CPM)

Composant permettant de déployer et d'exécuter un CPE dans un environnement UIMA

Le CPM permet :

- Le démarrage, la pause et la reprise des traitements
- L'exécution d'un sous-ensemble des TAE en respectant les contraintes inhérentes aux méta-données du CAS
- La définition d'une stratégie concernant la gestion des documents provoquant des erreurs
- Le monitoring des performances (temps, mémoire…)
- La parallélisation des traitements sur différents documents

Annexe B

ISO-TimeML
La norme et les balises

La norme TimeML

La norme TimeML est le fruit d'un groupe de travail dirigé par James Pustejovsky (Pustejovsky *et al.*, 2005) qui vise à standardiser les annotations sémantiques reliées à la temporalité dans un texte en langage naturel. Le guide d'annotation est tourné vers l'anglais, mais les éléments du standard sont à priori indépendants de la langue considérée. La norme traite trois sortes d'entités : les adverbiaux (objet TIMEX3), les événements (EVENT), et les signaux (SIGNAL). Elle introduit aussi plusieurs sortes de relations entre ces entités : liens d'ordonnancement (TLINK), de modalité (SLINK), ou bien aspectuels (ALINK), couvrant ainsi l'essentiel des informations temporelles que l'on peut associer à un texte.

.

Le schéma d'annotation :

TimeML (Pustejovsky et al., 2009) est un langage de spécification issu de l'atelier TERQAS1 dans le cadre du projet AQUAINT. Ce projet vise à améliorer les systèmes de questions-réponses en leur permettant de répondre à des questions de nature temporelle sur les entités et les événements. Dans ce cadre, TimeML a été élaboré comme langage d'annotation pour faciliter le raisonnement et l'inférence sur le temps.

Le schéma d'annotation prévoit les fonctionnalités suivantes : l'annotation des événements, l'étiquetage des expressions temporelles et la normalisation de leur valeur ainsi que la mise en évidence des relations qui existent entre ces deux types d'entités temporelles. Les traits de temps verbaux, la polarité et la modalité ainsi que la classe d'événement peuvent aussi être annotés.

Contrairement à des notions traditionnelles, TimeML adopte une conception large des événements qui regroupe les événements et certains états (ce qui correspond plus à la notion d'éventualité. En plus de la plupart des verbes, cette définition comprend des noms événementiels comme destruction et guerre, ainsi que des adjectifs (malade) et les groupes prépositionnels (à bord) qui désignent typiquement des états. TimeML compte 7 classes d'événements différents : ASPECTUAL, I_ACTION, I_STATE, OCCURRENCE,

PERCEPTION, REPORTING et STATE. Les événements sont annotés avec la balise EVENT. Le schéma d'annotation TimeML précise que c'est la tête lexicale du chunk événementiel qui doit être annotée. Ce choix est fait afin de simplifier l'annotation (le processus et le résultat), notamment en vue des difficultés présentées par les propositions enchâssées ou celles contenant plusieurs verbes. Cette simplification constitue une première étape dans le repérage des événements. (Pustejovsky et al., 2009) ont proposé une extension du schéma d'annotation qui consiste en l'ajout de balises pour capturer les arguments des événements, mais cette proposition n'a pas encore été intégrée.

Les informations sur la polarité (attribut polarity), l'aspect (aspect) et la modalité (modality) sont également représentées à l'intérieur de la balise EVENT.
Les TIMEX ont plusieurs attributs, comme le type (date, heure, durée), la valeur absolue correspondant à la localisation temporelle, le temps d'ancrage, pour les localisations qui sont calculées relativement à une autre (deux jours après *son départ*) et la quantité ou la fréquence, pour des localisations qui correspondent à plus d'un moment dans le temps (tous les jeudi d'avril). Les éventualités ont pour attribut une classe parmi : occurrence (la plupart des événements), état, action ou état intensionnel (ceux-ci, comme dans *je* **voudrais** *qu'il vienne*, suppose un second événement qui ne se réalise pas nécessairement), action d'énonciation ou de perception (dire, révéler, entendre, voir, etc.) et finalement, les événements aspectuels (commencer, finir de, continuer, etc.). Les autres attributs sont principalement : le temps grammatical, l'aspect progressif ou non (s'il est déterminable), la forme (nom, verbe, adjectif). Finalement, la balise `<SIGNAL>` sert à identifier un item (généralement une préposition) marquant une relation temporelle entre événements et localisations temporelles : *Il viendra* **après** *le 3 avril*.

Nous nous concentrons ici sur les entités à repérer, les relations qu'elles entretiennent étant un problème à part. Les expressions temporelles sont marquées par la balise TIMEX3. Elles se divisent en 4 classes :
Les dates (type DATE, le 15 janvier, 15.01.2008), les heures (type TIME, 15h20, l'après-midi), les durées (type DURATION, 5 jours, deux ans) et les ensembles (type SET, tous les jours, chaque année). TimeML permet aussi le raisonnement avec des expressions temporelles sousspécifiées, comme lundi prochain et l'année précédente, dont la valeur doit être déterminée par rapport à un point temporel de référence.

Les événements et les expressions temporelles sont mis en relation par trois sortes de liens (links) : liens temporels (TLINK), aspectuels (ALINK) et de subordination (SLINK). La première sorte capture des relations temporelles entre deux entités (EVENT-EVENT, TIMEX3-TIMEX3, ou EVENT-TIMEX3), la deuxième capture les phases dans le déroulement d'un événement et le dernier est essentiel pour les raisonnements qui dépendent de la véracité ou la certitude des propositions qui dénotent des événements. Les mots fonctionnels qui signalent explicitement un de ces liens sont annotés avec la balise SIGNAL. Le plus souvent ce sont des prépositions comme avant, après, pendant ou lors de.

Ci-dessous figure un exemple simplifié d'annotation avec les balises principales y compris un lien (TLINK) pour la relation temporelle entre l'événement et l'expression de date, pour la phrase Jean est arrivé avant le 11 février 2004 :

Jean est <EVENT id="e1" class="OCCURRENCE" pos="VERB" tense="PAST" polarity="POS">arrivé</EVENT> <SIGNAL>avant</SIGNAL> le <TIMEX3 id="t1" val="2004-02-11">11 février 2004</TIMEX3>.
<TLINK relType="BEFORE" event="e1" time="t1"/>

<TIMEX3> :

Type	Forme	Valeur
Dates	*le 3 janvier*	XXXX-01-03
	19/12/2006	2006-12-19
Heures	*17h30*	T1730
	5h du matin	T0500
Durées	*3 jours*	P3D
	100 ans	P100Y
Expressions déictiques	*l'année dernière*	2008
	Lundi prochain	2009-01-26

Annexe C

Le tagger OpenNLP partie du discours

Annotation des expressions temporelles

L'annotateur est composé de deux programmes: le programme de formation crée un fichier de paramètres à partir d'un lexique complet et un corpus tagué manuellement. Le programme de tagger lit le fichier de paramètre et annote le texte avec la partie du discours et l'information lemme. Les deux programmes donnent l'information sur leur utilisation quand ils sont appelés sans arguments.

Les formes de mots sont souvent ambiguës dans leur partie du discours (POS). Le mot anglais de forme store par exemple, peut être un nom, un verbe fini ou un infini. Dans un énoncé, cette ambiguïté est normalement résolue par le contexte d'un mot: par exemple dans la phrase « the 1978 PCs could store three pages of data » store ne peut qu'être à l'infinitif. La prévisibilité de la partie du discours du contexte est utilisée par un annotateur automatique de la partie du discours.

Plusieurs méthodes ont été proposées pour annoter automatiquement les mots avec un annotateur parti du discours. Certains chercheurs ont utilisé des systèmes à base de règles. les modèles de réseaux de neurones ont également été testés dans l'annotation POS [schmid,1994] et le problème connexe de la prévision POS.

Toutes les méthodes probabilistes citées ci-dessus sont basées sur le premier ordre ou le deuxième ordre du modèle Markov. Parce que le grand nombre de paramètres (en particulier le cas des trigrammes) de ces méthodes ont des difficultés à estimer avec précision les faibles probabilités.

Une nouvelle technique qui est présentée et qui évite ce problème de données en utilisant un arbre de décision pour obtenir des estimations fiables des probabilités de la transition. L'arbre de décision détermine automatiquement la taille appropriée du contexte qui est utilisée pour estimer la probabilité de transition.

Annotation probabiliste

L'annotateur arbre a beaucoup en commun avec un annotateur ngram classique. Dans les deux modèles, la probabilité d'une séquence marquée de mots (dans le cas d'un modèle de

Markov du second ordre) de manière récursive :

$P(w_1w_2w_3,\ldots w_n,t_1t_2t_3\ldots t_n) := p(t_n|t_{n-2}\,t_{n-1})\; P(w_n|t_n)\; P(w_1w_2w_3,\ldots w_{n-1},t_1t_2t_3\ldots t_{n-1})$(1).

Les méthodes diffèrent, toutefois, de la probabilité de transition

$p(t_n|t_{n-2}\,t_{n-1})$ est estimé.

Les annotateurs ngram estiment souvent la formule suivante basée sur l'estimation du maximum de vraisemblance (EMV) de principe:

$p(t_n|t_{n-2}\,t_{n-1}) = F(t_{n-2}\,t_{n-1}t_n)/\,F(t_{n-2}\,t_{n-1})$ (2) où $F(t_{n-2}\,t_{n-1}t_n)$ est le nombre d'occurrences du trigram $t_{n-2}\,t_{n-1}t_n$ dans le corpus et $F(t_{n-2}\,t_{n-1})$ est le nombre d'occurrences du bigram

$t_{n-2}\,t_{n-1}$.

Cette méthode d'estimation est problématique puisque de nombreuses fréquences sont petites de sorte que les probabilités correspondantes ne peuvent être estimées de manière fiable, des difficultés particulières sont posées par zéro fréquences: il est difficile de décider si le trigramme correspondant est syntaxiquement incorrecte ou juste rare .

Par conséquent, la formule ci-dessus est souvent modifiée par le remplacement des probabilités nulles avec une petite valeur et renormaliser les probabilités, de sorte que leur somme est égale à 1. Un choix approprié de la valeur de remplacement est essentielle pour la qualité du résultat d'annotation.

L'annotateur arbre

En contraste à un annotateur ngram, l'annotateur arbre estime les probabilités de transition avec un arbre de décision binaire. La figure 1 représente un arbre de décision de l'échantillon. La probabilité d'un trigramme donné est déterminée en suivant le chemin correspondant à travers l'arbre jusqu'à ce qu'une feuille soit atteinte. Si l'on regarde par exemple la probabilité d'un nom qui est précédé par un déterminant et un adjectif p (NN | DET, ADJ), nous devons d'abord répondre au test au niveau du nœud racine. Depuis l'étiquette du mot précédent est ADJ. Nous suivons le chemin. Le prochain test (tag-2 = DET) est vrai aussi et l'on finit par le niveau d'un nœud de feuille. Maintenant, nous avons juste à regarder pour la probabilité de l'étiquette NN dans le tableau qui est attaché à ce nœud.

Construction de l'arbre de décision

L'arbre de décision est construit de manière récursive à partir d'un ensemble de trigrammes en utilisant une version modifiée de l'algorithme ID3. À chaque pas de récurrence, un test est créé, qui divise l'ensemble d'échantillons de trigrammes en deux sous-ensembles avec netteté maximale en ce qui concerne la distribution de probabilité de la troisième (prédite) étiquette. Le test examine l'une des deux balises et contrôle les précédentes si elles sont identiques à une étiquette t. le test à la forme suivante:

$$Tag_{-i} = t \; ; \; i \in \{1,2\} \; ; \; t \in T, \qquad (3)$$

Où T est un ensemble d'étiquettes.

À chaque étape récursive, tous les tests possibles sont comparés et pour le meilleur rendement une information est attachée au nœud courant de l'arbre de décision. Puis ce nœud est développé récursivement sur chacun des deux sous-ensembles de l'ensemble qui est défini par le test. Les sous-arbres qui en résultent sont attachés au nœud actuel comme le sous-arbre yes et le sous-arbre no.

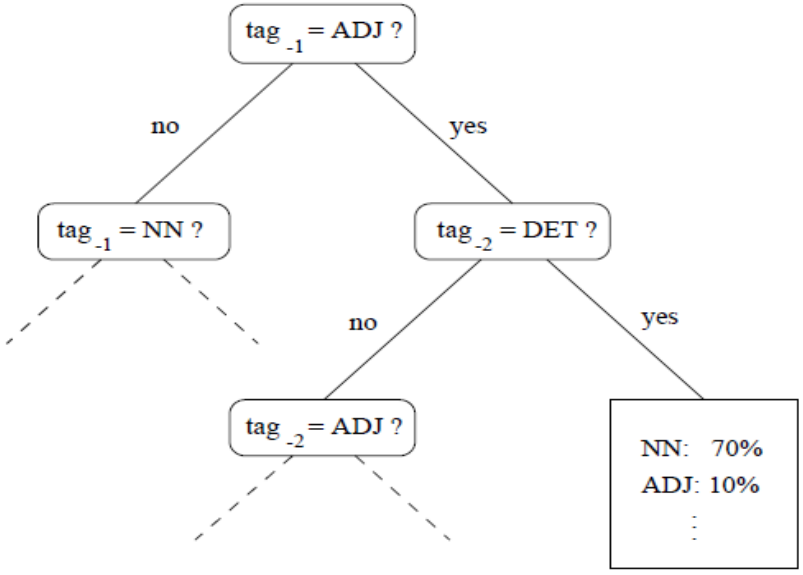


Figure : exemple d'arbre de décision

Le critère qui est utilisé pour comparer tous les tests possibles, q est la quantité d'informations qui est gagnée sur la troisième étiquette de chaque test. Maximiser le gain d'information est équivalent à minimiser la quantité moyenne de QI de l'information qui est encore nécessaire pour identifier la troisième balise après que le résultat du test q est connu.

$$I_q = - p(C+ \mid C) \sum_{t \in T} p(t \mid C+) \log_2 p(t \mid C+) - p(C- \mid C) \sum_{t \in T} p(t \mid C-) \log_2 p(t \mid C\_) \quad (4)$$

C est ici le contexte qui correspond au nœud courant et C + (C-) est égal à C, en plus la condition pour laquelle le test q réussit (échoue). p (C + | C) (p (C-| C)) est la probabilité que le test q réussit (échoue) et p (t | C +) (p (t | C-)) est la probabilité de la troisième étiquette t si le test réussi. Ces probabilités sont estimées à partir des fréquences avec MLE:

$$p(C_+ \mid C) = \frac{f(C+)}{f(C)} \quad (5)$$

$$p(C_- \mid C) = \frac{f(C-)}{f(C)} \quad (6)$$

$$p(t \mid C+) = \frac{f(t,C+)}{f(C+)} \quad (7)$$

$$p(t \mid C-) = \frac{f(t,C-)}{f(C-)} \quad (8)$$

L'expansion récursive de l'arbre de décision s'arrête si le prochain test génère au moins un sous-ensemble de trigrammes dont la taille est inférieure à un certain seuil prédéfini, par exemple 2 (c'est à dire f (c +) <2 ou f (c-) <2. Les probabilités p (t | c) pour la troisième balise sont alors estimées pour tous les trigrammes qui ont été transmis à cette étape de la récursivité et elles sont stockées au niveau du nœud courant

$$p(t \mid C) = \frac{f(t,C)}{f(C)} \quad (9)$$

La taille de l'arbre de décision :

Après la version initiale de l'arbre de décision qui a été construit, l'arbre est élagué. Si les deux sous-nœuds d'un nœud sont des feuilles, et le gain d'information pondérée au niveau du nœud est inférieur à un certain seuil, les sous-nœuds sont supprimés et le nœud devient une feuille elle-même. L'information gagné pondérée G est définie comme:

$$G = f(c)\,(I_0 - Iq) \quad (10)$$

$$I_0 = \sum_{t \in T} p(t \mid C) \log_2 p(t \mid C) \quad (11)$$

Où I_0 est la quantité d'informations qui est nécessaire pour lever l'ambiguïté au niveau du nœud courant et Iq, comme ci-dessus, est la quantité d'informations qui est encore nécessaire après que le résultat du test q est connu. Ce critère de gain de l'information ne doit pas être utilisé lors de la construction de l'arbre de décision, car il est possible qu'un nœud ne parvient pas à répondre, bien que tous ses sous-nœuds le fassent. Cette partie de l'arbre ne serait pas

construite si nous voulons utiliser le critère de gain de l'information au premier. Comme d'autres annotateurs probabilistes le font, l'annotateur arbre détermine la meilleure séquence de balise pour une séquence donnée de mots.

Lexique

Le lexique contient les étiquettes probables à priori pour chaque, il comporte trois parties: un lexique complet, un lexique de suffixes et une entrée par défaut. Au cours de la recherche d'un mot dans le lexique de l'étiqueteur arbre, le lexique de forme complète est recherché en premier. Si le mot est trouvé là, le vecteur étiquette de probabilité correspondante est retourné. Sinon les lettres majuscules du mot sont tournées vers minuscules, et la recherche dans le lexique de forme complète est répétée. Si elle échoue à nouveau, le lexique de suffixe est recherché prochainement. Si aucune des étapes précédentes n'a été réussie, l'entrée du lexique par défaut est renvoyée.

Au cours de la recherche d'un mot dans le lexique de l'annotateur arbre, le lexique de forme complète est recherché en premier. Si le mot est trouvé là, le vecteur étiquette de probabilité correspondante est retourné. Sinon les lettres majuscules du mot sont tournées vers minuscules, et la recherche dans le lexique de forme complète est répétée. Si elle échoue à nouveau, le lexique de suffixe est recherché prochainement. Si aucune des étapes précédentes n'a été réussie, l'entrée du lexique par défaut est renvoyée.

Le lexique complet a été créé à partir d'un corpus d'apprentissage étiqueté. Le nombre d'occurrences de chaque paire mot / étiquettes a été pris en compte et ces étiquettes de chaque mot avec une fréquence relative de moins de 1 pour cent ont été supprimées car elles étaient dans la plupart des cas, le résultat d'erreurs de marquage dans le corpus d'origine.

La deuxième partie du lexique, le lexique de suffixe, est organisée comme un arbre. Chaque nœud de l'arbre est marqué avec un caractère. Au niveau des nœuds des feuilles, des vecteurs étiquettes de probabilité sont attachés. Lors d'une recherche, l'arbre de suffixes est recherché à partir d'un nœud racine. À chaque étape, la branche qui est marquée avec le prochain caractère de la fin du mot suffixe est suivie.

Supposer par exemple que nous voulons chercher le mot de marquage dans le lexique de suffixe qui est montré dans la figure. Nous commençons à la racine (marqué #) et suivre la branche qui mène au nœud marqué g. A partir de là, nous nous dirigeons vers le nœud

étiqueté n, et finalement nous nous retrouvons dans le nœud marqué i. Ce nœud est une feuille et le vecteur étiquette de probabilité associée (qui n'est pas représenté sur la figure) est retourné. Le lexique de suffixe a été construit automatiquement à partir du corpus d'apprentissage. Un arbre de suffixe a été construit à partir des suffixes de la longueur 5 de tous les mots qui ont été annotés avec une classe partie de discours et les étiquettes fréquences ont été comptés pour tous les suffixes et stockées dans les nœuds de l'arbre correspondant. Alors une mesure d'information I (S) a été calculée pour chaque nœud de l'arbre :

$$I(S)= -\sum P(pos|S) \log_2 P(pos|S) \qquad (12)$$

Ici, s est le suffixe qui correspond au nœud courant et p (pos | s) est la probabilité de l'étiquette pos donne un mot avec le suffixe S.

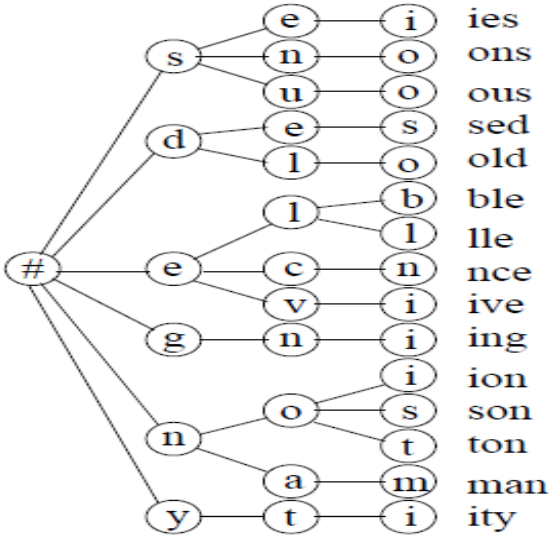


Figure : Echantillon d'un arbre de suffixe de longueur 3

En utilisant cette mesure de l'information, l'arbre de suffixe a été élagué. Pour chaque feuille, le gain d'information pondérée G (aS) a été calculé.

$$G(as)=F(aS)\ (I(S)-I(aS)) \qquad (13)$$

où S est le suffixe du nœud parent, comme le suffixe du nœud actuel et F(aS) est la fréquence de suffixe aS.

Si le gain de l'information à une certaine feuille de l'arbre de suffixes est inférieur à un seuil donné, la feuille est retirée. Les fréquences d'étiquette de tous les sous-nœuds supprimés d'un nœud parent sont recueillies au niveau du nœud par défaut du nœud parent. Si le nœud par défaut est le seul restant sous-nœud, il est supprimé aussi. Dans ce cas, le nœud

parent est une feuille. Pour illustrer ce processus nous considérons l'exemple suivant, où ess est le nœud parent, less est le suffixe d'un nœud enfant et Ness est le suffixe de l'autre nœud enfant.Un échantillon d'étiquettes de fréquences des nœuds est donné dans le tableau :

tag	suffix *ess*	suffix *ness*	suffix *less*
JJ	86	1	85
NN	10	2	8
NP	45	45	0
RB	2	0	2
total	143	48	95

Tableau: échantillon fréquence d'étiquette à un nœud de l'arbre et de ses deux nœuds enfants

La mesure de l'information pour le nœud parent est :

$$I(ess)=-\frac{86}{143}\log_2\frac{86}{143}-\frac{10}{143}\log_2\frac{10}{143}-\ldots 1.32 \qquad (14)$$

Les valeurs correspondantes pour les nœuds enfants sont 0,39 pour ness et 0,56 pour less.

Maintenant, nous pouvons déterminer le gain d'information pondérée à chacun des nœuds enfants. Nous obtenons:

$$G(ness)=48(1.32-0.39)=44.64 \qquad (15)$$

$$G(less)=95(1.32-0.56)=72.20 \qquad (16)$$

Les deux valeurs sont bien au-dessus d'un seuil de 10, et donc aucun d'entre eux ne devrait être supprimé.

Comme expliqué précédemment, l'arbre de suffixe est recherché lors d'une recherche le long du chemin, où les nœuds sont annotés avec les lettres du mot suffixe dans l'ordre inverse. Si une feuille est atteinte à la fin de la trajectoire, le vecteur de probabilité de mot-clé correspondant est retourné. Si aucune correspondance de sous-nœud peut être trouvée à un nœud sur le chemin, le nœud par défaut est suivi s'il existe. Si aucun nœud par default n'existe, la recherche dans le lexique de suffixe échoue et l'entrée par défaut est renvoyée. L'entrée par défaut est construite en soustrayant les fréquences d'étiquettes à toutes les feuilles de l'arbre de suffixes taillé à partir des fréquences d'étiquettes au niveau du nœud racine et en normalisant les fréquences résultantes.