Récupération et traitements linguistiques d'articles de Presse à partir de flux RSS

Premier Année Master Informatique

Liste des rédacteurs du cahier des charges :

- Saber NOUIOUA
- Mohamed wajdi BOUTITI
- Achref BERRICHE
- Fatma MARZOUGUI
- R'Kia INAGHNANE
- Salwa El BZIOUI

Cahier des charges
30/01/2015

TABLE DES MATIERES

TABLE DES FIGURES

INTRODUCTION

Le cahier de charges est un préalable à tout projet informatique. Pour réussir, tout projet doit suivre une logique dans laquelle le cahier des charges tient un rôle particulier, et dont on doit bien maitriser ces aspects : Etude de l'existant, analyse des besoins, spécifications des caractéristiques fonctionnelles, cadre juridique. C'est à partir de ce document, nous pourrons commencer notre projet et qu'on a eu le plaisir de préparer sur le thème : *Récupération et traitements linguistiques d'articles de presse à partir de flux RSS.*

Ce document peut ainsi être subdivisé en plusieurs parties, la première partie est une présentation de groupe, la deuxième partie sera consacré à l'objectif du projet car il est impératif que les objets soient bien établis avant de commencer notre projet, la troisième partie sera dédiée à des modules à développer , ensuite la quatrième partie : développement du projet est la partie la plus importante car elle définit les langages de programmation qu'on va utiliser pour un projet réussi, enfin la répartition des tâches correspond aux dates pour réaliser les activités, identifier les jalons et atteindre les objectifs du projet.

Au cours de ce cahier des charges, nous allons présenter un schéma globale du projet qui explique les modules à développer et chaque partie sera annotée, la démarche que nous aurons suivie et les différents choix effectués seront expliqués, jusqu'au résultat final de notre travail.

I. OBJECTIF DU PROJET :

Un flux **RSS** est un fichier XML répondant à une norme[1], contenant une suite d'informations provenant de sites d'actualités ou de blogs. Les fichiers RSS sont générés automatiquement de manière périodique, en fonction des dernières actualités parues sur le site. Chaque fichier RSS contient, pour chaque information un certain nombre de champs prédéterminés, et notamment: titre de l'information, courte description, lien vers la page contenant l'information complète, sous forme le plus souvent d'un « article de journal ».

Les flux RSS sont majoritairement utilisés par les journaux en ligne, et c'est dans ce cadre que se place notre projet.

L'objectif principal du projet est donc la mise en place d'un crawler de flux RSS pour les quotidiens nationaux et régionaux français.

Plus précisément, le projet comprend trois tâches principales:

- ✓ Récupération des flux RSS à partir d'une liste d'URL fournis par le client et qui pourront être mise à jour par lui;
- ✓ Traitements linguistiques du corpus ainsi récupéré;
- ✓ Stockage et gestion du corpus dans une base de données interrogeable.

1 Trois normes sont actuellement utilisées : RSS 0.91, sortie en 1999, RSS 0.90 et 1.0, sorti en 2000 ; RSS 2.0, sorti en 2002.

II. MODULES A DEVELOPPER :

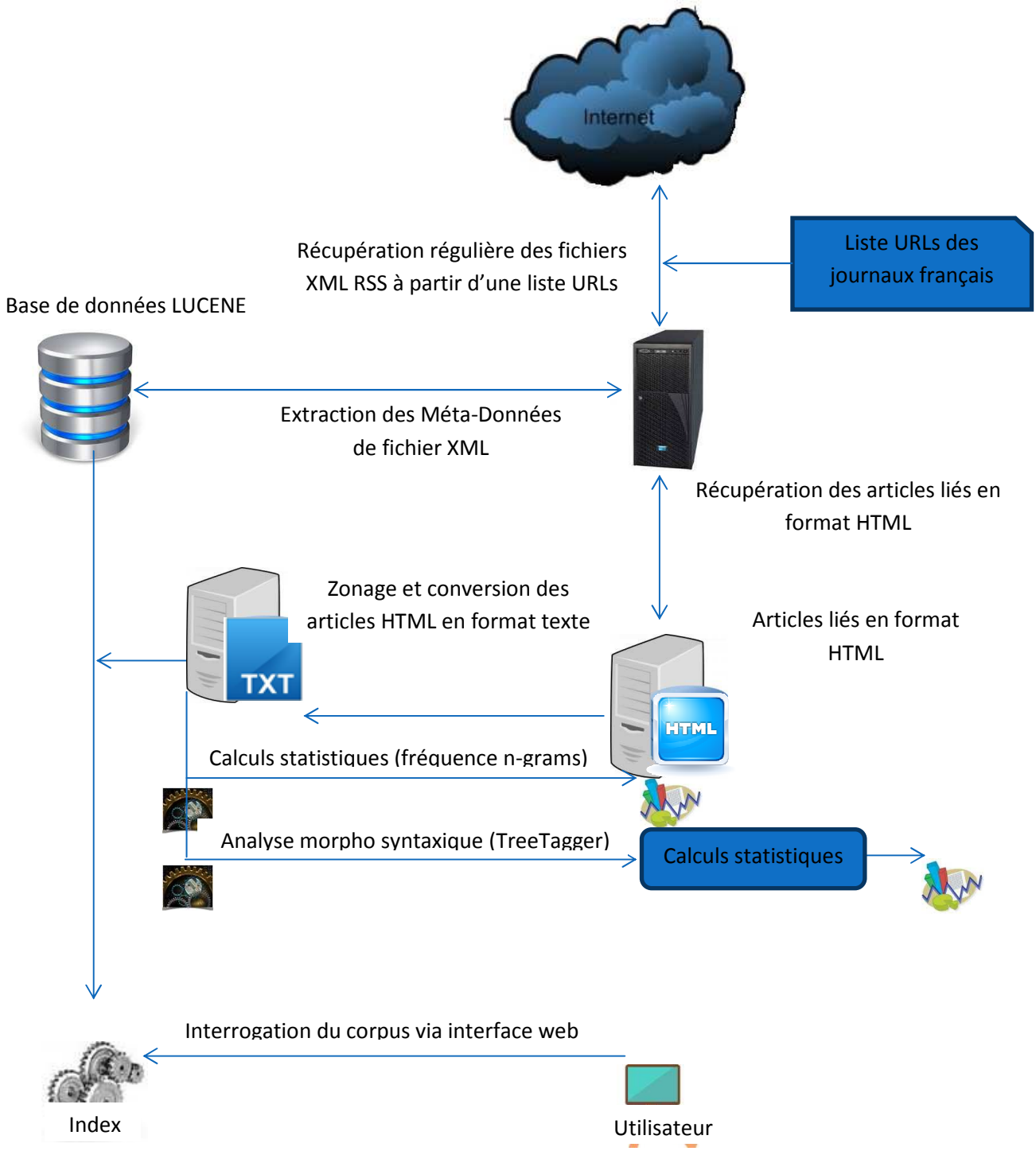

Figure1: *Schéma Global du Projet*

Notre projet est décomposé en trois principaux modules, illustrés comme suit :

A- Module Récupération des Flux RSS des Articles Liés :

L'objectif de ce module est la récupération du flux XML RSS à partir d'une liste d'URL donnée en entrée, l'extraction des Métadonnées pour chaque élément d'information, ainsi que la récupération de l'article lié en format HTML et sa conversion en format texte.

Ci-dessous sont détaillées les différentes tâches liées à ce module:

PROCESSUS	OUTIL UTILISE	ENTREE DU PROCESSUS	SORTIE
1. RECUPERATION DES FICHIERS XML RSS	➢ RSSCORPUSBUILDER	➢ LISTE D'URL	➢ METADONNEES ; ➢ ARTICLE LIE EN FORMAT HTML.
2. EXTRACTION DES METADONNEES A PARTIR DU FICHIER XML RSS; 3. SAUVEGARDE DES METADONNEES DANS LA BASE DE DONNEES LUCENE ;	➢ LUCENE ; ➢ RSSCORPUSBUILDER.	➢ FICHIER XML RSS	➢ METADONNEES EN LUCENE.
4. SAUVEGARDE DE L'ARTICLE LIE EN FORMAT HTML.	➢ RSSCORPUSBUILDER.	➢ FICHIER XML RSS	➢ FICHIER HTML SUR DISQUE.
5. ZONAGE ET CONVERSION DES ARTICLES HTML EN FORMAT TEXTE	➢ JUSTEXT, BOILERBIBE, NCLEANER, CONTENTEXTRACTION	➢ PAGE HTML	➢ FICHIER TEXTE

1. **Récupération des fichiers XML RSS à partir d'une liste URL :**

La première tâche consiste à récupérer les flux XML RSS à partir d'une liste d'URL (quotidiens, journaux français) donnée en paramètre du programme. Cette tâche sera programmée régulièrement deux fois par jours pour récupérer le maximum des informations. Pour ce faire, nous reprendrons le programme RSSCorpusBuilder du client, que nous réécrirons en Java. On pourra utiliser pour le lancement périodique du programme une tâche CRON sous Linux.

2. **Extraction des Méta-Données de fichier XML et stockage dans LUCENE :**

La deuxième tâche consiste à extraire les Métadonnées (title, pubDate, description, …) de chaque nouvel Item, et les sauvegarder dans la base de données LUCENE. L'algorithme sera repris du programme RSSCorpusBuilder.

3. **Récupération des articles liés et sauvegarde sur disque :**

La troisième tâche consiste à récupérer les articles liés pour chaque nouvel **Item** du flux XML RSS, et de les sauvegarder sur disque. L'algorithme sera repris du programme RSSCorpusBuilder.

4. **Zonage et conversion des articles HTML en format texte :**

La quatrième tâche consiste à convertir les pages HTML récupérées en format texte en utilisant la technique « BoilerPlate Removal ».

❖ La technique de BoilerPlate Removal

Une page HTML contient plusieurs informations hétérogènes (styles variés, bandeaux, publicités, etc.). Dans notre projet, l'objectif est de ne récupérer que le contenu principal correspondant à la partie textuelle de l'article. Il sera également souhaitable de distinguer entre les différents types de contenu, comme le corps de l'article, les commentaires des utilisateurs, publicités, etc.

L'objectif est donc d'extraire seulement le contenu de l'article et de le convertir en format texte, et d'exclure tous les autres composants de la page HTML (Header, Footer, Side Bar) comme le montre le schéma suivant :

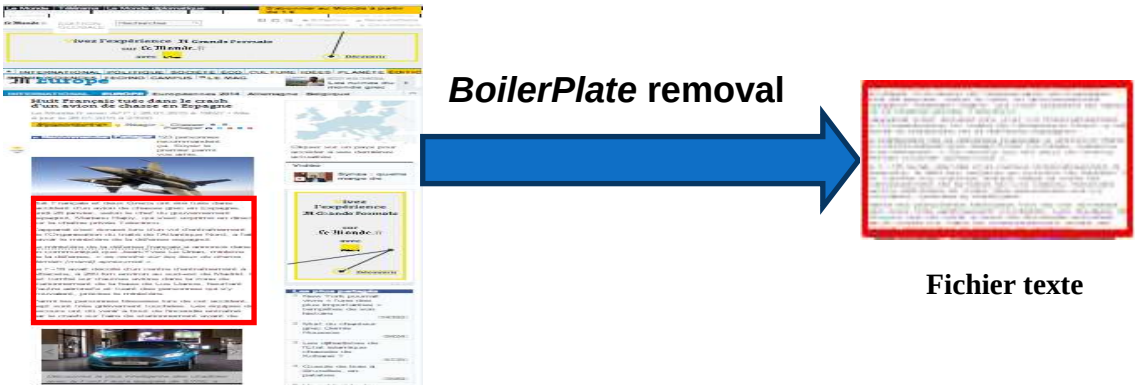

Figure 2: *Schéma explique la technique de BoilerPlate Removal*

Il existe plusieurs librairies gratuites (Open Source) disponible sur internet permettant l'extraction du contenu principal d'une page HTML:

1- BoilerPipe :

BoilerPipe est une bibliothèque Java open source écrit par **Christian Kohlschütter** sous la licence Apache 2.0. Elle fournit des algorithmes, pour détecter et supprimer le surplus autour de l'article principal d'une page HTML. Il a été écrit pour analyser les pages Web, dans le but d'extraire le contenu principal.

2- JusText :

- JusText est une **API Python** gratuit, permettant de supprimer les liens de navigation, en-têtes et pieds de page de pages HTML. Il est conçu pour préserver l'article contenant des phrases complètes et il est donc bien adapté pour la création des Corpus linguistiques.
- Il existe aussi des algorithmes de JusText en c++ permettant la récupération des contenus des pages HTML.

3- ContentExtraction :

ContentExtraction comprend des algorithmes écrit en c++ permettant la suppression des liens de navigation, en-têtes et pieds de page de pages HTML, et la récupération du contenu de la page en format texte.

4- NCleaner :

C’est un package sous linux permettant la détection des BoilerPlate, et l’extraction du contenu d’une page web.

❖ **Choix de l’outil** :

Selon un article récent (**More Effective Boilerplate Removal-the GoldMiner Algorithm : référence**) sur la comparaison des différents outils, et quelques tests initiaux sur les différents outils, **JusText** semble être la plus efficace dans cette tâche.

B- Module Traitement Linguistique :

Les fichiers texte seront ensuite traités linguistiquement, dans ce deuxième module. Les traitements se feront à partir du texte brut et à partir du texte analysé morpho-syntaxiquement.

➢ **Calculs statistiques** à partir du texte brut et du texte annoté morpho syntaxiquement:

Il s'agit de calculer la fréquence des n-gram sur la totalité du corpus. On appelle n-gram une séquence de mots éventuellement discontinue. Par exemple, dans la phrase *il est quatre heures du soir*, on peut définir les 2-grams suivants, en utilisant une fenêtre de 3 mots : i*l est, il quatre, il heures, est quatre, est heures, est du, etc.* L'intérêt de ce calcul est que la répétition d'un n-gram sur corpus est souvent le signe d'un usage linguistique, que le client cherche à repérer. Ce calcul sera fait sur le texte brut, mais aussi avec un texte analysé morpho syntaxiquement, c'est-à-dire pourvu d'informations linguistiques complémentaires (partie du discours, lemme). Par exemple : *il/il/PROPERS est/être/AUX quatre/quatre/DETNUM heures/heure/NC*. Dans ce cas, le calcul de fréquence prendra en compte la combinaison des informations.

Les outils disponibles :

1- SemanticVectors :

C’est une bibliothèque java open source compatible avec LUCENE, elle permet de crée des modèles sémantiques à partir du texte libre en langage naturel. Ces modèles sont conçus pour représenter des mots et des documents en termes de concepts. Ils peuvent être utilisés pour de nombreuses tâches, telles que la génération automatique de thésaurus, la représentation des connaissances.

2- word2vec :

C’est un outil développés en plusieurs langages (Java, C, Python), il fournit une implémentation efficace des bag-of-Word et skip-gram pour le calcul des représentations vectorielles des mots. Ces représentations peuvent ensuite être utilisées dans de nombreuses applications de traitement du langage naturel.

L'outil **word2vec** prend comme entrée un corpus de texte et produit en sortie les vecteurs de mots. Le fichier de vecteur de mot résultant peut être utilisé dans de nombreuses applications de traitement du langage naturel et d'apprentissage.

3- SDMC :

C’est un outil développé par Thierry CHARNOIS au LIPN qui permet de calculer des séquences répétées sur gros corpus, en combinant les niveaux d'analyse. Cet outil est parfaitement adapté au cas de l'analyse statistique sur corpus annoté morpho syntaxiquement.

❖ Choix de l’outil :

L'outil qui sera utilisé pour le calcul statistique sur texte brut n'a pas encore été choisi. Une phase de test et une comparaison des résultats sera faite.

- **Analyse morphosyntaxique** :

Il s'agit de l’étape de projection d’informations morphosyntaxiques sur les mots. L’analyse morphosyntaxique est l’ensemble des techniques qui concourent à passer d’un texte brut, exempt d’informations linguistiques, à une séquence des mots étiquetés par des informations linguistiques. Par exemple, à partir de la phrase brute *il est quatre heures,*, nous obtiendrons *il/il/PROPERS est/être/AUX quatre/quatre/DETNUM heures/heure/NC.*

Les outils disponibles :

Reconnaître la catégorie morphosyntaxique d’un mot dans un contexte est une tâche non triviale du traitement automatique de la langue écrite.

En effet rendre une machine capable d’identifier la catégorie d’un mot exige de mettre en œuvre des méthodes sophistiquées, en particulier pour les mots ambigus, c’est-à-dire susceptibles d’appartenir à plusieurs catégories différentes.

1. **TreeTagger :**

Est un outil permettant l'étiquetage morphosyntaxique et la lemmatisation. Il a été développé par Helmut Schmid(1994) dans le cadre de projet TC. Il a été utilisé avec succès pour de nombreuses langues (anglais, français, allemand, italien, néerlandais, espagnol, bulgare, russe, grec, portugais, chinois, swahili). Il est adaptable sur toutes les langues en utilisant un lexique et un corpus d'apprentissage manuellement étiquetés.

Pour la langue française, (Stein, 2007) a entrainé cet analyseur sur un corpus d'apprentissage contenant 2 685 146 mots et l'a évalué en utilisant un corpus contenant 500 000 mots.

Il rapporte un taux de précision de 92,7% pour l'étiquetage et 97,8% pour la lemmatisation. **TreeTagger** peut en effet présenter la lemmatisation des mots en plus des étiquettes.

2. **MaltParser :**

Est un système pour l'apprentissage d'analyseurs en dépendances syntaxiques. A partir d'un corpus annoté, le système apprend à projeter des traits syntaxiques et morphosyntaxiques sur des décisions d'analyse (shift, reduce, création d'arcs de dépendances).

C'est un système open source implanté en Java et disponible à l'url http://w3.msi.vxu.se/~nivre/ research/MaltParser.html.

3. **Stanford Pos tagger :**

A pour but d'étiqueter chaque mot en fonction de son rôle syntaxique, par exemple, nom, adjectif, adverbe, etc.

❖ **Choix de l'outil :**

On n'a pas encore choisi un outil pour notre travail (phase de test et comparaison des résultats à effectuer, en liaison avec le client, et en tenant compte des impératifs techniques, notamment l'interfaçage avec les autres modules).

C- Module Stockage et Gestion des Données :

L'objectif de ce module et l'indexation des fichiers texte récupérés auparavant par la technique « BoilerPlate Removal » en utilisant l'outil LUCENE, pour indexer et faire des recherches sur les fichiers textes. Ainsi que le développement d'une interface web pour l'interrogation des données en question.

a) Indexation des données (Fichier Textes et Méta-Données)

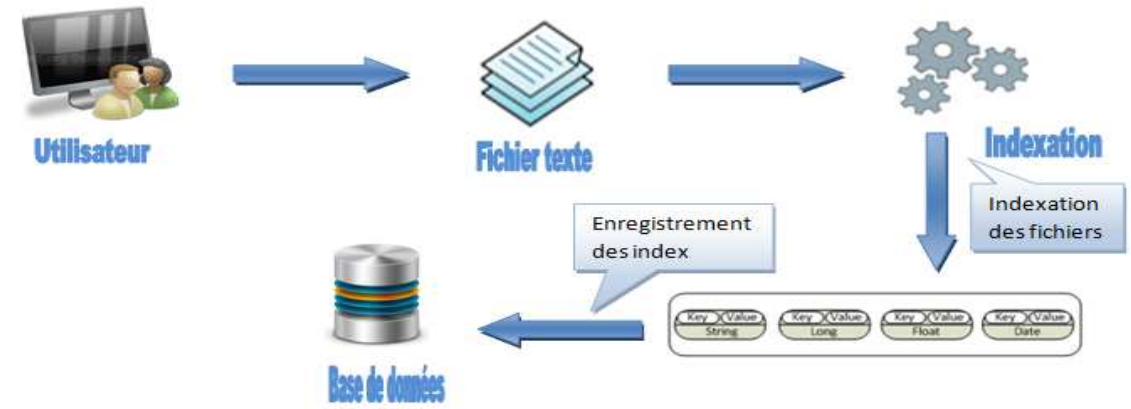

Figure 3: *Indexation de données*

❖ **Les outils disponibles :**

1- Lucene :

Lucene est une bibliothèque open source écrite en Java qui permet d'indexer et de chercher des fichiers textes. Il est utilisé dans certains moteurs de recherche.

2- Xapian :

Xapian est un outil très souple qui permet à des développeurs d'ajouter très facilement à leurs applications des fonctions d'indexation et de recherches.

3- Sphinx :

Est un logiciel libre permettant d'indexer différents types de données.

❖ **Critères de choix du LUCENE :**

- ✓ Lucene est une des librairies de récupération de données les plus complètes et riches en fonctionnalités.
- ✓ Haute performance indexation ;
- ✓ Puissants, précis et efficace Algorithmes de recherche ;
- ✓ De nombreux types de requêtes puissantes:
- ✓ Recherche de proximité ;
- ✓ Recherche par les Méta-Données (title, datePub, description, …).
- ✓ Choix de client.

b) Interface web pour l'interrogation des données

Il s'agit de développer ou utiliser des interfaces web déjà disponibles pour l'interrogation des index et l'affichage des résultats des requêtes.

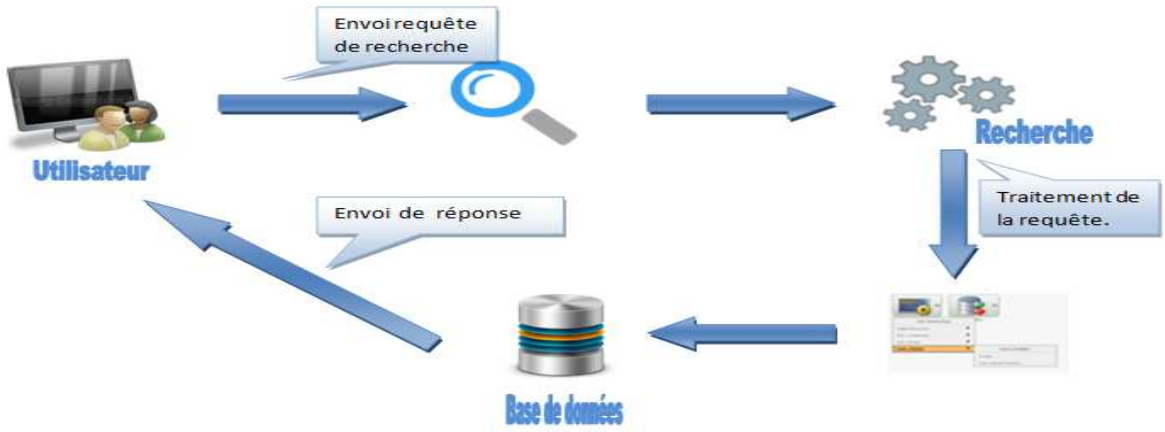

Figure 4: *Interface Web*

❖ **Les outils disponibles :**

1) **Luke** : (Lucene Index Toolbox)

Est un outil de diagnostic et de développement pratique qui accède aux index de LUCENE déjà existants et permet de montrer et modifier leur contenu de plusieurs façons.

2) **LIMO :** (Lucene Index Monitor)

Donne des informations de bases sur les index utilisés par le moteur de recherche LUCENE.

1) **Solr :**

Est une plateforme logicielle de recherche s'appuyant sur le moteur de recherche LUCENE.

2) ElasticSearch :

Est un moteur de recherche libre (open source) basé sur LUCENE.

III. Recette et Livrable

Pour chaque module, une série de tests (« recette ») seront mis en place, en liaison avec le client, permettant d'assurer une qualité de résultats minimum, et une efficacité des programmes. Ces tests seront mis en place dans la première phase du projet, en liaison avec le client.

D'autre part, le programme répondra aux exigences du client en ce qui concerne l'installation sur une machine serveur Linux. Le programme sera livré sous forme d'un programme exécutable et d'une API permettant de lancer le traitement dans son entier ou seulement certains modules. Ce livrable principal sera accompagné des deux éléments suivants :

- **Documentation**

Le produit sera livré avec un manuel d'utilisation, un manuel d'installation et de déploiement, ainsi qu'une fiche technique détaillant toutes les fonctionnalités du produit, et les différents appels du programme possibles.

- **Rapport de fin de projet**

A la fin du projet, un rapport complet sera livré détaillant les programmes développés, les résultats obtenus pour chaque module et les limites éventuelles.

IV. Environnement de développement :

- ✓ **Plateforme :** Linux

C'est un choix du client.

- ✓ **Modélisation :** UML

Vu l'importance cruciale de la modélisation dans le cycle de vie de n'importe quelle application, il fallait utiliser un langage de modélisation qui s'adapte à nos besoins et à nos exigences. On va présente de ce partie le diagramme de use cas

- ✓ **Langage de programmation :** Java et PHP

Le langage **Java** est un langage de programmation orienté objet, open source, multiplateforme.

- ✓ **Outils de développement :**

NetBeans est un environnement de développement intégré, facile à utiliser, placé en open source et il est disponible sur toutes les plates formes Linux, Windows.

V. Répartition des Tâches :

Nous détaillons ci-après les différentes tâches de l'équipe de développement. Nous rendrons compte au client régulièrement du travail effectué, minimalement une fois tous les quinze jours. Les tâches sont divisées sur les membres du groupe de travail, selon les compétences de chacun dans les différents domaines.

1- Réparation des tâches :

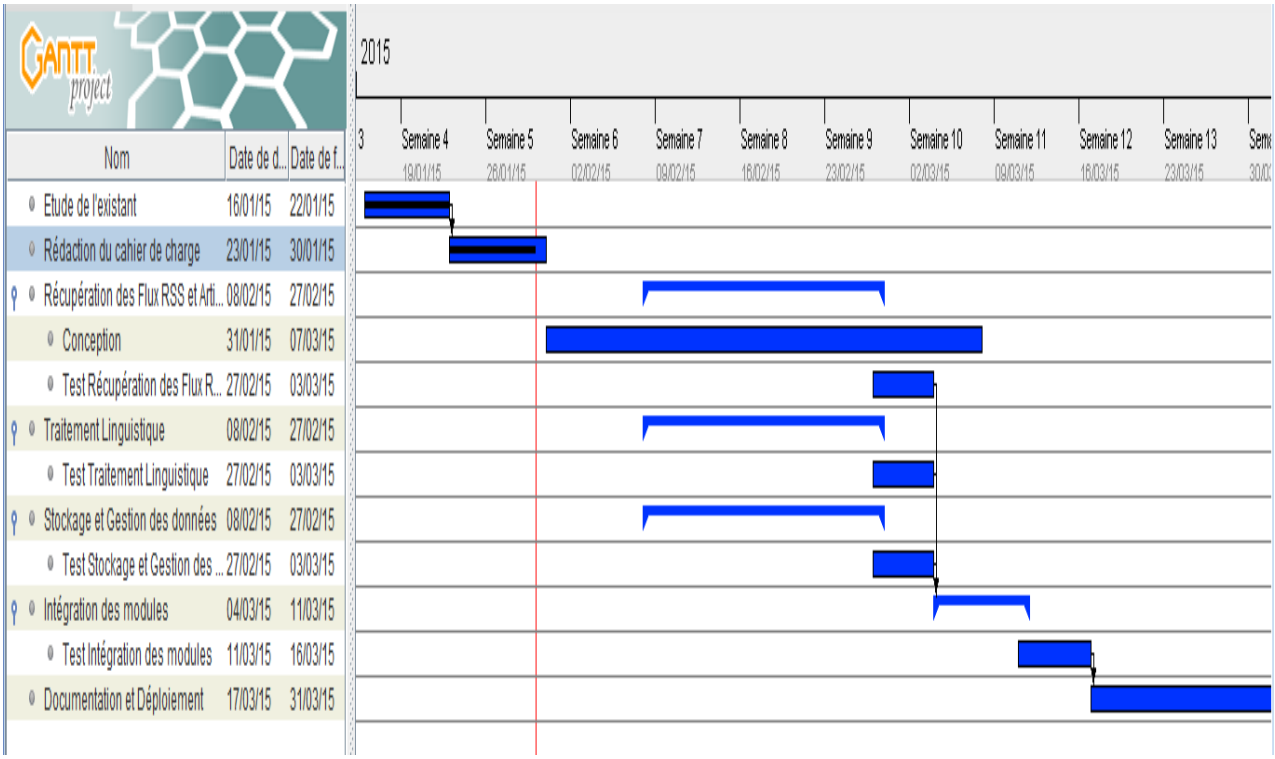

Figure 5: *Diagramme de Gant 'Répartition de tâches'*

2- Gestions des ressources :

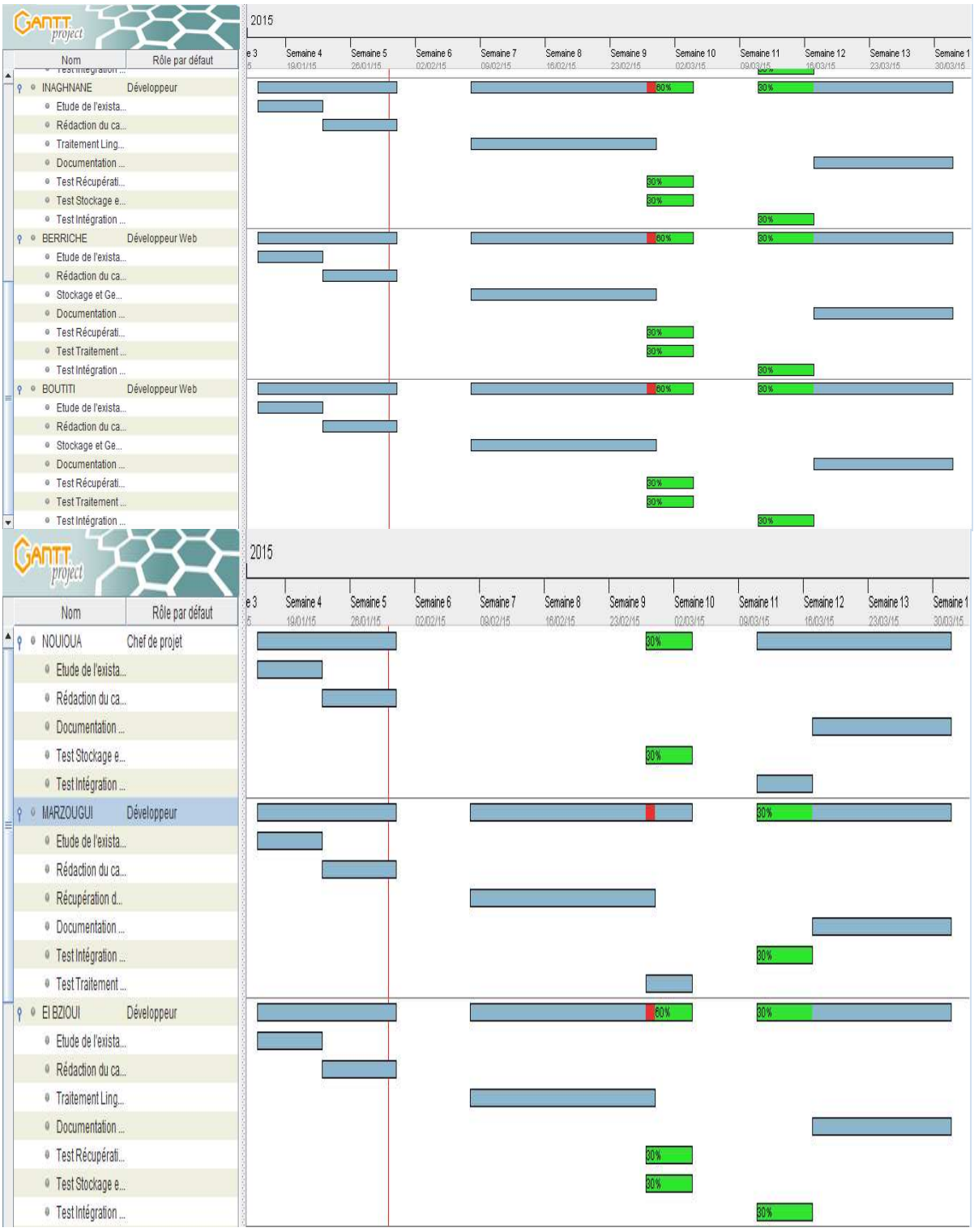

Figure 6: *Diagramme de Gant 'Gestion des ressources humaines'*

Conclusion

La réalisation de ce projet a été une bonne occasion pour nous d'une part d'acquérir de nouvelles connaissances, et d'autre part, d'assimiler les différents outils acquis durant ce semestre en matière de développement. Nous avons eu l'occasion d'améliorer nos connaissances en conception, d'appliquer l'UML et langage JAVA pour concevoir une grande partie de notre travail. En outre, de bien comprendre et mettre en œuvre le déroulement d'un cycle de vie d'un logiciel de gestion.

L'objectif visé à travers ce travail est de concevoir et d'implémenter un système qui permet de récupérer un maximum d'informations à partir d'un site web. Les différentes tâches fixées ont été réalisées à partir de plusieurs hypothèses. Nous avons modélisé les opérations importantes en respectant les contraintes fixées.

Nous avons trouvé cette expérience très enrichissante, au niveau personnel que professionnel. Nous espérons avoir répondu au mieux aux attentes du projet à travers nos travaux et nos conclusions.