REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE

DEPARTEMENT : D'INFORMATIQUE

N° :……..

DOMAINE : Mathématique et Informatique

FILIERE : Informatique

OPTION : TIC

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par : MOHAMMED CHIKOUCHE Samah

Intitulé

Résumé automatique des textes

Soutenu devant le jury composé de :

……..	Université de M'sila	Président
BOUZAROURA Ahlem	Université de M'sila	Encadreur
AMROUNE Nasreddine	Université de M'sila	Co-Encadreur
……..	Université de M'sila	Examinateur

Année universitaire : 2016 /2017

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE
DEPARTEMENT : D'INFORMATIQUE
N° :……..

DOMAINE : Mathématique et Informatique
FILIERE : Informatique
OPTION : TIC

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par : MOHAMMED CHIKOUCHE Samah

Intitulé

Résumé automatique des textes

Soutenu devant le jury composé de :

……..	Université de M'sila	Président
BOUZAROURA Ahlem	Université de M'sila	Encadreur
AMROUNE Nasreddine	Université de M'sila	Co-Encadreur
……..	Université de M'sila	Examinateur

Année universitaire : 2016 /2017

Dédicaces

Je dédie ce modeste travail à celle qui m'a donné

La vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur

Et ma réussite, à ma mère.

A mon époux Nacer et mon enfant Elhadj Ilyas,

A mon père, mes frères (Saleh et Fares) et mes sœurs (Nouha, Fatma, Imane et Amina),

qui n'ont pas arrêté de m'encourager et de me soutenir.

Que Dieu les garde et les protège.

A ma grand-mère, mes tantes, leurs époux et leurs enfants

A mon beau-père, ma belle-mère.

A toutes ma famillle.

REMERCIMENTS

Au terme de ce travail

Nous remercions, en premier lieu, ALLAH le tout puissant pour nous avoir donné la patience, le courage, la force morale et physique pour élaborer ce mémoire.

Nous remercions également Madame BOUZAROURA Ahlem pour avoir accepter de suivre notre projet.

Un grand merci à Monsieur AMROUNE Nasereddine.

Nous les remercions pour leurs conseils utiles, qui ont grandement contribué à notre formation. Nous espérons être digne de la confiance qu'ils ont placé en nous.

Un merci à tous nos enseignants pour toutes les connaissances qu'ils nous ont inculquées tout au long de notre cursus. Nous tenons à vous exprimer toutes nos gratitudes et tous nos respects.

Nos remercîments vont aussi à tous ceux qui ont contribué de près ou de loin à la concrétisation de ce travail, pour leurs conseils, leurs encouragements et leurs soutiens.

Nous tenons à remercier aussi l'honorable jury qui nous a accordé son temps pour lire notre humble travail et juger sa valeur.

Table des matières :

INTRODUCTION GENERALE

Avec l'avènement d'Internet et du web 2.0 des quantités phénoménales d'informations exprimées dans les langues naturelles sont générées. Cette évolution en volume de textes nécessite la production d'outils informatiques performants dont la tâche est de traiter ces langues naturelles, les traduire ou trouver et d'extraire l'information pertinente.

Le besoin s'est rapidement fait sentir de s'appuyer sur les techniques linguistiques pour faciliter la communication homme-machine. Parallèlement la linguistique a pu profiter de la puissance des ordinateurs pour acquérir une nouvelle dimension et ouvrir la voie à de nouveaux domaines de recherche dont le T.A.LN. L'objectif du traitement automatique des langues est la conception de programmes capables de traiter des données exprimées dans une langue naturelle pour lesquelles plusieurs phases d'analyse (morphologique, syntaxique, sémantique et pragmatique) sont nécessaires afin d'en extraire des informations ou les présentés sous une forme condensée.

Le résumé automatique de texte est l'un des applications du TALN les plus connues, il se trouve à la croisée de deux disciplines : traitement automatique de la langue (TAL) et recherche d'information (RI). Le résumé automatique de texte consiste à produire une représentation courte d'un texte tout en conservant l'information pertinente.

A partir de cela on se demande, à quel point ces méthodes peuvent servir pour nous aider de produire des résumés pertinents ? Les quelles parmi toutes ces méthodes sont meilleures pour la langue française et anglaise ? Est-ce qu'une combinaison de ces méthodes peut donner des meilleurs résultats ?

Dans ce mémoire nous allons mettre l'accent sur les points suivants : les différentes techniques utilisées dans le résumé automatique, les techniques de résumé qui peuvent être adaptés à la langue française et anglaise, le corpus d'articles de presse publiés sur le web.

Donc, notre démarche vise à mettre en œuvre un système de résumé automatique des textes en français et en anglais, en adaptant différentes techniques d'extraction qui ont déjà été utilisées pour la langue arabe.

De ce fait, notre travail sera organisé en trois chapitres :

- chapitre 01 : traitement automatique du langage naturelle.
- chapitre 02 : le résumé automatique.
- chapitre 03 : conception et implémentation du système de résumé automatique.

CHAPITRE 01

GENERALITE SUR LE TRAITEMENT AUTOMATIQUE DU LANGAGE NATURELLE

1. Introduction:

On regroupe sous le vocable de traitement automatique du langage naturel (TALN) l'ensemble des recherches et développements visant à modéliser et reproduire, à l'aide de machines, la capacité humaine à produire et à comprendre des énoncés linguistiques dans des buts de communication. Il sera donc question ici de langage humain, d'où l'adjectif naturel, et non pas de langage formel, tel que C ou encore ADA. Ce naturel fait d'ailleurs tout le sel de l'affaire : les langages formels sont précisément conçus et optimisés dans l'optique de manipulations algorithmiques. Il en va tout autrement pour le langage naturel, dont le traitement automatique pose des difficultés majeures. Précision importante, nous nous limiterons quasiment exclusivement au traitement du langage sous forme écrite, le traitement de la parole étant encore, en dépit de convergences de plus en plus marquées avec le traitement de l'écrit, une question de traitement du signal.

2. Historique de traitement automatique de langage naturel (TALN) :

TALN commence dans les années 1950, bien que l'on puisse trouver des travaux antérieurs. En 1950, Alan Turing éditait un article célèbre sous le titre « Computing machinery and intelligence » qui propose ce qu'on appelle à présent le test de Turing comme critère d'intelligence. Ce critère dépend de la capacité d'un programme informatique de personnifier un humain dans une conversation écrite en temps réel, de façon suffisamment convaincante que l'interlocuteur humain ne peut distinguer sûrement sur la base du seul contenu de la conversation s'il interagit avec un programme ou avec un autre vrai humain.

Pendant les années 1960, SHRDLU a été l'un des premiers programmes informatiques de compréhension du langage naturel et le meilleur logiciel qui permettait un dialogue interactif avec l'utilisateur, à base de termes en anglais. Blocks world un système de langage naturel dont la base était des vocabulaires relativement restreints, fonctionnait extrêmement bien, invitant les chercheurs à l'optimisme.

Cependant, le progrès réel était beaucoup plus lent, et après le rapport ALPAC de 1966, qui constatait qu'en dix ans de recherches les buts n'avaient pas été atteints, l'ambition s'est considérablement réduite.

ELIZA un programme qui simule un entretien avec un psychiatre, écrite par Joseph Weizenbaum entre 1964 à 1966. N'employant presque aucune information sur la pensée ou

l'émotion humaine, ELIZA parvenait parfois à offrir un semblant stupéfiant d'interaction humaine. Quand le patient dépassait la base de connaissances.

Pendant les années 1970 beaucoup de programmeurs ont commencé à écrire des « ontologies conceptuelles », dont le but était de structurer l'information en données compréhensibles par l'ordinateur. C'est le cas de MARGIE (Schank, 1975), SAM (Cullingford, 1978), PAM (Wilensky, 1978), TaleSpin (Meehan, 1976), SCRUPULE (Lehnert, 1977), Politics (Carbonell, 1979), Plot Units (Lehnert 1981).

D'autres systèmes similaires à ELIZA ont été créés comme PARADE, Racter, et Jabberwacky. Dès les années 1980, à mesure que la puissance informatique augmentait et devenait moins chère, les modèles statistiques pour la traduction automatique ont reçu de plus en plus d'intérêt. [13]

3. Définition de TALN :

TALN est une discipline à la frontière de la linguistique, de l'informatique et de l'intelligence artificielle, qui concerne l'application de programmes et techniques informatiques à tous les aspects du langage humain. [11]

Le TALN est l'ensemble des méthodes et des programmes qui permettent un traitement par l'ordinateur des données langagières, mais quand ce traitement tient compte des spécificités du langage humain. Il y a des traitements de données langagières (écritures sur fichiers, sauvegardes ou autres) qui ne font pas partie du traitement automatique des langues. [5]

4. Objectifs de TALN:

- Parmi les objectifs de courts termes :

Développer des logiciels ou des programmes informatiques capables de traiter de façon automatique des données linguistiques.

Pour traiter automatiquement ces données, il faut d'abord expliciter les règles de la langue puis les représenter dans des formalismes opératoires et calculables et enfin les implémenter à l'aide de programmes informatiques. [5]

- À longue terme :

Créer un ordinateur qui comprend le langage naturel des humains.

Le TALN permet de vérifier les théories linguistiques ou, de manière plus générale, de mieux comprendre comment les humains communiquent entre eux. À cette fin, il utilise l'ordinateur pour simuler les capacités humaines de compréhension et de production de la langue naturelle. Les résultats ainsi obtenus, peuvent ensuite être comparés aux performances humaines et les théories sur lesquelles se fondent les simulations, vérifiées. [5]

5. Outils de TALN:

Le traitement automatique des langues naturel nécessite évidemment des outils divers que l'on peut grouper en trois catégories distinctes :

- ✓ Linguistiques: ils décrivent les différentes connaissances relatives à la langue. [6]
- ✓ Formels: ils expriment les connaissances linguistiques dans un formalisme qui convient à un traitement automatique. [6]
- ✓ Informatiques: ils utilisent la description formelle des connaissances dans une application informatique concrète. [6]

6. Domaines de recherche de TALN :

La diversité des outils utilisés dans le traitement automatique a provoqué la diversité du TALN, qui fait intervenir des recherches dans différents domaines :

- ✓ L'informatique :

  Qui permet d'optimiser les algorithmes et les programmes de traitement, mais aussi de développer des techniques formelles de démonstration ou de résolution de problèmes, etc. [4]

- ✓ La linguistiqui :

  Qui fournit des théories explicites du savoir linguistique. [4]

- ✓ La linguistique informatique:

  Qui développe des programmes de TALN et définissent, dans ce but, de véritables langages informatiques spécialisés pour les applications du TALN. [4]

- ✓ Les mathématiques :

  Qui étudient les propriétés formelles des outils de traitement et des théories (sous la forme de l'algèbre, de la logique ainsi que des statistiques). [4]

✓ L'intéligence artificielles :

Qui s'occupe de la représentation des connaissances et de leur utilisation. [4]

7. Applications de TALN:

Parmi les applications de TALN sont :

7.1 L'indexation automatique et la recherche documentaire :

Comme la plupart des informations ont la forme de textes en langue naturelle (références, livres, journaux, articles, etc.), l'intérêt d'une recherche documentaire automatique est évident, elle doit permettre de retrouver automatiquement les informations, les documents pertinents ou les références des documents, qui répondent à une question de l'utilisateur. [17]

7.2 Le résumé automatique :

Sur lequel nous allons consacrer notre étude, est une extension de l'indexation et la recherche automatique, il est devenu aujourd'hui une discipline indépendante. Il sert à produire une version condensée d'un texte afin de ne garder que les informations pertinentes de ce dernier. [17]

7.3 La traduction automatique :

Ou traductique, est l'une des premières applications du TALN, elle se définit comme l'application de l'informatique à la traduction de textes oraux ou écrits d'une langue naturelle de départ (ou langue source) dans une langue d'arrivée (ou langue cible). [17]

7.4 Les correcteurs :

Sont généralement les correcteurs d'orthographe, de syntaxe et de style, qui se limitent à l'analyse du langage et aide l'humain à transformer un texte en un autre qui est la correction du premier. Ce sont des outils très répondus, ils font déjà partie intégrante de la plupart des logiciels de traitement de texte. [17]

7.5 La génération de textes en langue naturelle :

Elle vise à produire des textes en langue naturelle à partir de données non linguistiques, comme des graphiques, des schémas, des dessins ou des données numériques etc. [17]

7.6 Les systèmes de dialogue homme-machine :

Ce sont des interfaces en langue naturelle qui permettent à l'être humain de converser avec les ordinateurs. Il permet par exemple, l'interrogation de banques de données en

langue naturelle, commande de robots ou de machines ou encore dialogue avec des systèmes experts etc. [17]

8. Connaissances sur la langue:

Pour traiter automatiquement les langues naturelles, le programme idéal devrait inclure différentes connaissances relatives à la langue, les différents mots, leur prononciations, leur significations, comment ils se combinent pour former une phrase et comment le sens des différents mots contribue au sens de la phrase. De plus, le programme devrait pouvoir utiliser les connaissances générales sur le monde et les contextes d'utilisation des textes. [1]

Ces différentes connaissances sont généralement classées et étiquetées sous les rubriques suivantes :

- ✓ Les connaissances phonétiques et phonologiques :

Concernent les sons et la manière dont les mots sont réalisés en sons.

- ✓ Les connaissances morphologiques :

Concernent la manière dont les mots sont construits à partir des unités minimales de dignification, appelées morphèmes.

- ✓ Les connaissances syntaxiques :

Décrivent la manière dont les mots se combinent en phrases syntaxiquement correctes et encode ainsi leur régularité structurale.

- ✓ Les connaissances sémantiques :

Concernent les sens des mots et la manière dont les sens se combinent pour former le sens global de la phrase.

- ✓ Les connaissances pragmatiques :

Incluent les connaissances générales sur les concepts du monde et celles relatives à la situation de communication, c'est-à-dire à l'ensemble des conditions psychologiques, sociales et historiques qui déterminent l'émission d'un énoncé à un moment donné du temps et en un lieu donné.

9. Les niveaux de traitement dans un système TALN :

Pour traiter les langages naturels, on a besoin d'informations coordonnées et pertinentes sur la langue à différents niveaux. Le plus souvent, on a recours à cinq niveaux de connaissance sur la langue : phonologique, morpho lexical, syntaxique, sémantique et pragmatique :

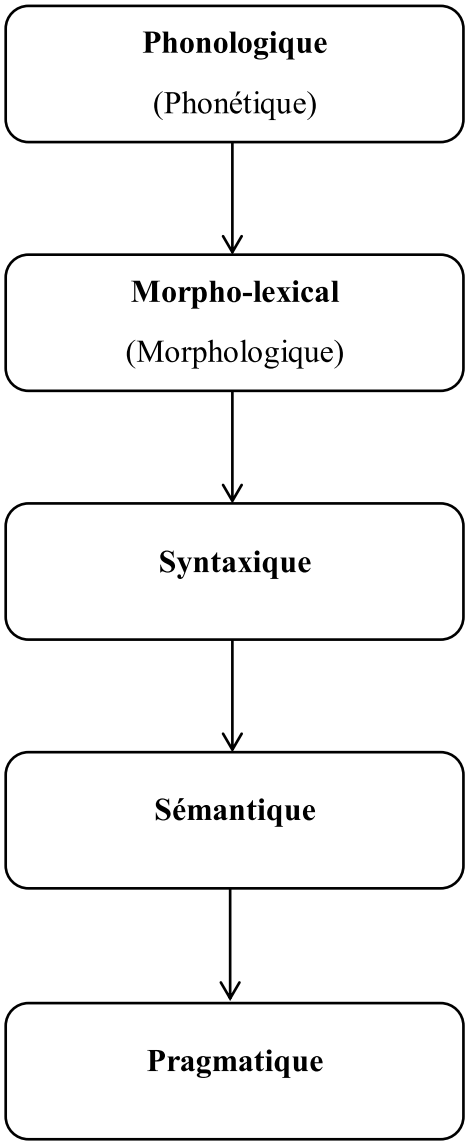


Figure 1.1: Niveaux de traitement du langage naturelle.

9.1 Niveau phonologique (phonétique) :

Ce niveau est utilisé particulièrement dans le traitement de la langue orale, à ce niveau la machine doit reconnaître des signaux acoustiques et les identifier en tant que mots via une interface vocale. [12]

9.2 Niveau morpho-lexical (morphologique) :

Dans ce niveau, le programme TALN étudie la formation des mots et leur variation de formes. Deux termes sont à étudier dans ce niveau, la flexion et la dérivation :

- La flexion :

C'est l'ensemble de modifications que subit un mot dans sa terminaison. Selon le rôle qu'elle joue dans la phrase, la flexion est un processus qui consiste à modifier le radical d'un mot pour lui adjoindre à certains types d'éléments.

- La dérivation :

C'est un procédé de formation de mots nouveaux par addition, suppression ou remplacement d'un élément grammatical d'un mot simple. [12]

9.3. Niveau syntaxique :

On s'intéresse à ce niveau à l'agencement des mots et leurs relations structurelles. Les connaissances syntaxiques concernent la façon dont les mots sont agencés dans une phrase, c'est-à-dire, sa structure grammaticale. Ces connaissances ou règles sont décrites dans des grammaires, leur application permet la formation des phrases correctes et de lever les ambiguïtés. [12]

9.4. Niveau sémantique :

La sémantique est une discipline qui a pour objectif la description des significations propres aux langues et leurs organisations théoriques. En TALN, la sémantique peut être définie comme l'étude de sens des mots, des phrases et des énoncés.

Le rôle de l'analyseur sémantique est donc d'attribuer un sens à la phrase structurée par l'analyseur syntaxique.

Les connaissances sémantiques nécessaires pour donner un sens aux noms sont seules qui explicitent non seulement la relation entres les mots et les objets, actions ou idées qu'ils désignent mais aussi les conditions qui permettent d'évaluer si une phrase à un sens ou non. [12]

9.5. Niveau pragmatique :

Pour pleinement comprendre un ou un texte dans son ensemble, il faut aussi avoir des connaissances pragmatiques, c'est-à-dire, celles qui permettent de situer le mot dans le contexte. Les connaissances pragmatiques précisent une représentation du monde référence

qui constitue la culture commune nécessaire aux interlocuteurs. Le niveau pragmatique est le niveau le plus difficilement accessible aux machines car certains énoncés ne se comprendre que dans un contexte géographique, historique ou culturel donné. [12]

10. Problème de TALN:

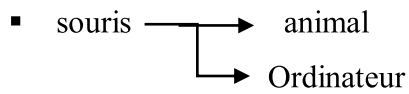
Parmi les problèmes les plus importants du TALN, nous trouvons en premier lieu celui de l'ambiguïté. Nous disons qu'il y a une ambiguïté lorsqu'il y a plus d'une interprétation possible pour une structure linguistique donnée. En effet, l'être humain parlant une langue donnée passe à côté de la plupart des ambiguïtés parce qu'il est doué de la manière de les résoudre, il utilise ses connaissances qu'il a accumulées à propos du monde et du contexte, mais malheureusement les machines ne sont pas encore aussi fortes. Les applications du TALN, sont souvent confrontées aux ambiguïtés à tous les niveaux d'analyse.

Les énoncés en langue naturelle sont donc extrêmement ambigus, la nature des ambiguïtés est variée et les processus de désambiguïsation font intervenir une grande diversité de connaissances qui interagissent de manière complexe.

La résolution des ambiguïtés lors du traitement automatique des langues naturelle comporte deux taches, descriptive ou procédurale. [13]

En citez quelques exemples:

- Ambiguïté des graphèmes (lettres) : comparez la prononciation du i dans lit, poire, maison ….
- Ambiguïté des terminaisons : ainsi un »s » final marque à la fois le pluriel des noms, des adjectifs…
- ambiguïté dans les propriétés grammaticales et sémantiques. Un même mot peut avoir plusieurs sens :



11. Conclusion :

Nous avons présenté dans ce chapitre, une brève introduction au domaine de traitement automatique des langues naturelles, ces différentes disciplines, ainsi que la nécessité de larges connaissances linguistiques et informatiques pour pouvoir développer des applications capables à répondre aux besoins des utilisateurs en terme de traitement automatique des langues, et plus particulièrement, la langue française connue par la complexité de ces structures linguistiques.

Dans le chapitre suivant, nous entamerons la discipline de notre étude, à savoir, le résumé automatique de texte, qui est un domaine de traitement automatique des langues écrites.

CHAPITRE 02

LE RESUME AUTOMATIQUE

1. Introduction:

De nos jours, des millions de documents électroniques (voire des milliards) sont disponibles via l'Internet (réseaux sociaux, forum…etc.) ; il devient ainsi de plus en plus difficile d'accéder aux informations intéressantes sans l'aide d'outils spécifiques, dont la technologie du résumé automatique qui permet de choisir un ou plusieurs documents utilement. Le but du résumé est d'aider le lecteur à décider si le document source contient l'information recherchée ou pas. Il se peut aussi que le lecteur n'ait pas besoin de lire la totalité du document source simplement parce que l'information recherchée existe dans le résumé.

Le résumé automatique a connu un fort renouveau ces dernières années. Si les recherches menées dans ce domaine s'inscrivent dans une tradition longue de plus de 50 ans , elles ont fortement évolué récemment : l'apparition de gros corpus parfois hétérogènes et la généralisation des techniques d'analyse de surface ont à la fois renouvelé les besoins et les approches. Plus récemment encore, avec l'avènement de médias plus interactifs (le fameux Web 2.0), la nécessité de repérer les citations, les jugements et les opinions s'est avérée de plus en plus cruciale.

En effet, de telles données, aujourd'hui facilement accessibles, peuvent aider à comprendre les besoins et les attentes de toute une population, et également à analyser les opinions concernant des produits, des personnalités ou même des propositions politiques.

Dans le présent chapitre on essaye de présenter, clarifier les différentes approches, les méthodes et définir quelques concepts liés au résumé automatique qui est notre cas d'étude.

2. Definitions:

Depuis le livre de résumé automatique on trouve les définitions suivantes :

- Le résumé :

  Est une forme de compression textuelle avec perte d'information. [17]

- Le résumé automatique de texte :

  Est une des méthodes de fouille de texte qui permet de compresser un document avec perte d'information, tout en conservant son information. Il s'agit d'une problématique importante du Traitement Automatique de Langues (TAL). Résumer consiste à condenser l'information la plus importante issue d'un ou plusieurs documents, afin d'en produire une version abrégée de son contenu. [17]

3. Les différentes approches de résumé automatique :

Il existe trois approches principales pour générer des résumés de texte :

3.1 Extraction :

Consistant à choisir des extraits appropriés (des phrases, des paragraphes, etc.) du texte original et à les enchaîner dans une forme plus courte. Le texte résumé est extrait du texte sur une base statistique ou en employant des méthodes heuristiques ou une combinaison des deux.

Souvent, on extrait du texte source les phrases complètes jugées les plus importantes. Cette approche a l'avantage d'être facile à réaliser mais elle risque d'introduire une certaine incohérence dans les résumés.

L'objectif de cette approche est de pouvoir fournir rapidement, sans analyse en profondeur du texte, un résumé à l'utilisateur. On repère et extrait les segments textuels (phrases ou paragraphes) les plus pertinents du texte afin de construire un sous-ensemble d'extraits textuels que l'on considère comme un résumé.

L'avantage de la méthode par extraction est de ne pas passer par une analyse en profondeur du texte, et de pouvoir fournir un résumé de façon plus simple sans devoir générer du texte.

Le résumé par extraction est qu'il évite la génération de texte. Ceci permet d'une part, de se concentrer sur la sélection du contenu pertinent et d'autre part, d'obtenir un résumé lisible et linguistiquement correct. [8]

3.2 Abstraction :

le texte résumé est une interprétation du texte original avec un processus de production par récriture du texte source en une version plus courte par le remplacement de certains concepts. Sa mise en œuvre exige l'utilisation de grammaires et de lexiques pour l'analyse syntaxique et la génération, en plus d'une modélisation de la compréhension humaine des textes. Cette approche est la plus difficile. [17]

3.3 Compression des phrases :

Consiste à générer un résumé par compression de phrases : les phrases ainsi extraites sont ensuite compressées afin d'éliminer l'information superflue. [17]

Ces approches peuvent être combinées en vue d'obtenir de meilleurs résumés.

4. Objectifs du résumé automatique :

L'objectif de résumé automatique de textes présente le texte source dans une version plus courte avec la sémantique.

Les résumés automatiques se diffèrent par rapport à leurs objectifs : Indicatif ou Informatif

4.1 Le résumé indicatif :

Le résumé indicatif a pour fonction de fournir au lecteur suffisamment d'informations pour qu'il puisse juger s'il doit consulter ou non le texte source. Pour un document correspondant à ce que recherche un lecteur, le résumé indicatif doit pouvoir acheminer correctement ce lecteur vers celui-ci à travers la lecture de son contenu qui doit pouvoir le faire décider ou non de la consultation du document. Ce type de résumé contient seulement des éléments partiels par rapport au résumé informatif, mais surtout des éléments pertinents en vue de répondre à sa fonction. De ce fait, s'il consiste à diriger le lecteur vers le document initial, il ne se substitue pas à la lecture de ce dernier. [17]

4.2 Le résumé informatif :

Le résumé informatif fournit un ensemble informations permettant de donner un large panorama du contenu d'un texte. Pour cela, l'ensemble des principaux sujets doit être rapporté. De plus, le résumé informatif tend à conserver l'organisation générale du texte source. Ainsi les sujets principaux qui sont rappelés dans le résumé sont répartis de manière fidèle par rapport à l'organisation initiale afin de donner un juste aperçu du texte source. [17]

5. L'utilité et les types de résumés :

Est de produire une version condensée d'un ou plusieurs textes en utilisant des techniques informatiques.

L'avantage le plus important d'utiliser un résumé, il réduit le temps de lecture. [2]

Il y a plusieurs types de résumés selon leur but :

- ✓ Mono-document(le résumé d'un document isolé),
- ✓ Multi-document(le résumé d'un groupe de documents pas forcément hétérogène, portant souvent sur une thématique bien précise.), guidé ou non par une requête utilisateur, entre autres.

Dernièrement des résumés autres que textuels ont vu leur jour. Ainsi des résumés audio et vidéo font partie des recherches actuelles. Des résumés dans des domaines très spécialisés

comme la médecine ou la chimie organique posent des vrais défis aux systèmes de traitement automatique de la langue naturel. [8]

6. Méthodes de résumés automatiques :

Nous présentons brièvement différentes méthodes employées pour l'extraction des phrases, elles sont basées sur le calcul d'un score associé à chaque phrase afin d'estimer son importance dans le texte. Le résumé final ne gardera que les phrases avec les meilleurs scores.

6.1 Méthodes à base de mots :

Cette méthode est basée sur le fait que l'auteur se sert (pour exprimer ses idées principales) de quelques mots-clés qui ont tendance à être récurrents dans le texte. Le résumé automatique est alors produit en recherchant dans le texte source les unités de texte minimales réunissant ses mots-clés. Ce principe est souvent appliqué en différentes variantes présentées dans les sous-sections qui suivent. [8][9]

6 .1.1 Mots-clés prédéfinis :

Pour calculer le score de chaque phrase S selon les mots-clés qu'elle contient, on peut calculer le score suivant :

$$Score_{mot-clé}(S) = \sum_{weS} a(w) * F(w) \quad (1)$$

$$a(w) = \begin{cases} A \; si \; w \in liste \; de \; mots - clés \; (A > 1) \\ 1 \; sinon \end{cases}$$

F(w) est la fréquence du terme w dans la phrase S

La liste de mots-clés peut être introduite par l'utilisateur (domaine d'intérêt) ou composée des mots-clés établis par l'auteur. L'importance du poids du terme w est donné par A×F(w), avec A> 1.

6.1.2 Titres :

Étant donné que le titre est l'expression la plus significative et qui résume le mieux un document en quelques mots, on peut dire que la phrase qui ressemble le plus au titre est la plus marquante du document. Par conséquent, on peut attribuer à chaque phrase un poids en fonction de sa ressemblance avec le titre.

Dans ce cas on considère les mots du titre du texte comme des mots-clés et on produit le résumé en sélectionnant les phrases qui couvrent certains mots apparaissant dans un titre.

$$Score_{titre}(S)\ \sum_{wes} b(w) * F(w) \tag{2}$$

$$b(w) = \begin{cases} A\ si\ w \in liste\ de\ mot\ titre\ (A > 1) \\ 1\ sinon \end{cases}$$

6.1.3 Distribution des termes :

L'idée de cette méthode est de considérer comme importantes les phrases qui contiennent des mots importants du texte. Un mot est considéré important s'il est employé assez fréquemment dans le texte. Pour le calcul des fréquences on considère généralement les mots qui appartiennent à des classes non fermées de la langue telles que les noms et les verbes.

$$Score\ tf.idf(S) = \frac{1}{|S|}\sum_{wes} tf.idf(w) \tag{3}$$

$$tf.idf(w) = \frac{tf(w) - 1}{tf(w)} log \frac{DN}{df(w)}$$

|S|= nombre de mots dans la phrase S.

Tf(w) est la fréquence du terme w dans le document.

df(w) est le nombre de document du corpus ou le terme w apparait.

DN est le nombre de document dans le corpus.

$$Score(S) = \frac{1}{|S|}\sum_{wes} Score(w) \tag{4}$$

$$Score(w) = F(w) * \frac{log(|S|)}{S(w)}$$

F (w) est la fréquence du terme w dans la phrase.

S (w) nombre de phrase dans lesquelles w apparait.

6.2 Méthode à base de position :

Cette méthode suppose que la position d'une phrase dans un texte indique son importance dans le contexte. Les premières et les dernières phrases d'un paragraphe, par exemple, peuvent transmettre l'idée principale et devraient donc faire partie du résumé.
Comme variante de cette méthode on peut citer la méthode Lead : c'est une méthode qui détermine les phrases importantes en extrayant celles qui sont en tête. Cette méthode est efficace pour résumer les articles de journaux, puisque les phrases importantes ont tendance à apparaître dans les premières phrases de l'article.

On définit le score d'une phrase S à la position i comme suit :

$$Score_{lead}(Si) = \beta i$$

$$\beta i = \begin{cases} B > 0 \; si \; i < N \\ 0 \quad\quad si \; i \geq N \end{cases} \qquad (5)$$

βi est une fonction rectangulaire qui modélise la distribution de phrases importantes selon leur position.

Dans le cas où les dernières phrases auraient une certaine importance, il suffit d'introduire un nouvel intervalle pour la valeur de i.

L'inconvénient de cette méthode est qu'elle dépend de la nature du texte à résumer ainsi que du style de l'auteur. [8][9]

6.3 Méthode dépendant de la longueur de phrase :

Cette méthode attribue un poids à une phrase en fonction du nombre de mots dans la phrase. Deux techniques peuvent être employées pour le calcul du score :

- longueur de chaque phrase (Li) par rapport à la longueur maximale de la phrase.

$$Score_{long}(S_i) = L_i / L_{max}$$

- affecte un score nul à une phrase plus courte qu'une certaine longueur (L minimale) :

$$Score_{long}(Si) = \begin{cases} 0 \; si \; Li \leq L_{min} \\ \frac{L_i - L_{min}}{L_{min}} \quad si \; L_i > L_{min} \end{cases} \qquad (6)$$

6.4 Méthode à base d'expressions indicatives :

Cette méthode choisit des unités de texte avec des indications spécifiques ou des expressions spécifiques. Par exemple, pour les textes scientifiques, on a comme expressions le but de ce travail..., ce papier présente..., les résultats et des conclusions sont de bons candidats pour indiquer les phrases à inclure dans un résumé. Des textes de types différents peuvent avoir des expressions indicatives différentes.

On peut déduire un score pour une phrase d'un texte quelconque à analyser en fonction de la ressemblance qu'elle présente, pour le trait donné. [8][9]

On pourrait définir le score d'une phrase S correspondant à un certain motif comme :

$$Score_{cue}(S) = \begin{cases} 1 \; si \; S \; correpond \; \text{à} \; un \; motif \\ 0 \qquad\qquad\qquad\qquad\quad sinon \end{cases} \tag{7}$$

6.5 Méthode basée sur les relations (cohésion lexicale) :

L'exploitation des fréquences de mots est un bon moyen pour faire ressortir les termes clés dans un texte mais elle ne prend pas en compte les relations entre les mots dans les différentes parties du texte. L'extraction de phrases basée sur la fréquence de mots cause souvent un manque de cohésion. Pour pallier ce problème, on a développé une approche basée sur la cohésion grammaticale (c'est-à-dire, la référence, la substitution, la conjonction) et la cohésion lexicale (c'est-à-dire, des mots liés sémantiquement).

Cette méthode suggère que plus une phrase est liée à une autre dans un texte, plus elle est appropriée dans ce contexte c'est-à-dire qu'elle exprime le même sujet. Ainsi, des phrases liées doivent être choisies ensemble pour composer un résumé. L'omission de certaines phrases fortement corrélées pourrait produire des textes incohérents.

L'identification de telles corrélations est basée normalement sur un thésaurus ou lexique informatisé qui permet de déterminer les relations entre les mots. On construit des chaînes lexicales à partir des mots candidats du texte, ces chaînes regroupent des mots liés par des relations obtenues à partir du thésaurus. Les phrases qui sont le plus connectées aux chaînes lexicales sont extraites. [8][9]

6.6 La méthode d'exploration contextuelle :

La méthode d'exploration contextuelle vise à identifier les connaissances linguistiques dans le texte en les restituant dans leurs contextes et en les organisant en tâches spécialisées. L'approche est fondée sur la construction manuelle d'une base de données de marqueurs linguistiques et une expression de règles d'exploration contextuelle. Ces règles appliquées aux phrases du texte source vont filtrer les informations sémantiques indépendantes du domaine avec les étiquettes sémantiques hiérarchisées comme : énoncés structurants, définition, causalité, etc. La stratégie de sélection des unités saillantes est fonction des besoins des utilisateurs.

L'exploration contextuelle appuie son analyse sur une hiérarchisation des connaissances. À cet effet, on distingue quatre niveaux de connaissances :

- ✓ les connaissances linguistiques (grammaticales et lexicales), indépendantes des connaissances sur le monde externe.
- ✓ les connaissances propres à un domaine particulier : elles concernent le savoir-faire lié au domaine de compétence et les règles qui organisent ce domaine.
- ✓ les connaissances socioculturelles qui dépendent de l'environnement social, des usages, des coutumes, etc.
- ✓ les connaissances encyclopédiques qui sont générales et communes à tous les êtres humains.

Les stratégies décisionnelles de l'exploration contextuelle s'expriment sous la forme de règles heuristiques qui identifient en premier lieu un indicateur pertinent, caractéristique du problème à résoudre. Ensuite le contexte linguistique est fouillé pour rechercher des indices linguistiques afin de prendre une décision adéquate.

Les avantages de cette technique sont : l'indépendance entre les connaissances linguistiques nécessaires au système et les connaissances accumulées sur un domaine particulier. Elle permet une extensibilité incrémentale, en complétant les listes déjà établies (recherche d'indices plus fins) et en affinant les règles d'exploration (Descellés). Un Système d'exploration Contextuelle est donc plus ou moins performant selon la richesse des indices pris en compte et la finesse de l'exploration. [8][9]

6.7 Méthode hybride :

Les méthodes présentées dans les sections précédentes utilisent des traits (fréquence, position, expression indicative, etc.) qui ne peuvent isolément garantir des résultats optimaux. On combine souvent ces traits par exemple avec l'équation suivante :

$$Score_{hybride}(S) = a_1 * Score_{tf.idf}(S) + a_2 * Score_{lead}(S) + a_3 * Score_{cue}(S) + a_4 * Score_{titre}(S) \quad (8)$$

Les poids ai peuvent être fixés arbitrairement ou déterminés de manière expérimentale (par apprentissage par exemple).

Certaines expériences sur un corpus hétérogène de 200 documents ont montré que si on combine les méthodes cue, titre et position (poids zéro pour la méthode mots-clés), on obtient de meilleurs résultats que si on les combine avec la méthode mots-clés.

Dans le cas de textes journalistiques, ont combiné les méthodes de distribution de termes, du titre, de la position et de la cue, en considérant la spécificité du texte. Ils ont fait ressortir que les phrases qui commencent par des nominaux ou contiennent des mots du titre semblent être plus pertinentes que des phrases n'ayant pas ce caractère. De plus, les mots ou les phrases n'apparaissant que dans quelques paragraphes sont plus importants que ceux mentionnés dans tous les paragraphes. Pour garder la cohérence du texte, le résumé est composé d'une sélection de paragraphes pertinents. [8][9]

7. Conclusion:

Dans ce chapitre nous avons présenté quelques techniques pour le résumé automatique de textes : les méthodes d'extraction offrent certains avantages : simplicité de mise en œuvre,

Certaines méthodes semblent offrir de meilleurs résultats que d'autres, cela est dû en grande partie à la nature du texte et au style de l'auteur.

Dans le chapitre suivant nous allons présenter la méthode choisie avec les algorithmes et les démonstrations nécessaires.

CHAPITRE 03

CONCEPTION ET IMPLEMENTATION DU SUSTEME

1. Introduction:

Ce chapitre expose les détails techniques liés à notre système (résumé automatique), l'environnement de développement, langage de programmation choisis et quelques captures d'écran présenter le résumé d'un document ou multi-document de notre système.

2. Description de l'approche proposée :

Comme nous avons vu dans le 2éme chapitre il existe trois approches pour le résumé automatique, dans notre approche nous allons utiliser l'approche par extraction.

L'objectif du résumé extractif est de sélectionner les phrases les plus pertinentes du texte. La méthode proposée utilise la linguistique pour le prétraitement des phrases et la lemmatisation .Ainsi, un algorithme de clustering appliqué sur une matrice de similarité pour regrouper l'ensemble des phrases similaires dans des clusters. Puis, on applique des paramètres de sélection pour déterminer la ou les phrases les plus pertinentes dans chaque cluster. En conséquence, le résumé est la réorganisation des phrases extraites. [17]

Donc, le système de résumé comprend trois étapes majeures :

- Prétraitement.
- Extraction de termes de fonctionnalité (mots clés, mots de titre, termes fréquents).
- Sur l'ensemble des phrases du texte appliquer l'algorithme de classification du clustering *K-means* couplé avec une mesure de similarité.
- Pour chaque cluster, on choisit la ou les phrases les plus pertinentes.
- Réorganiser les phrases pour avoir le résumé.

Le schéma suivant représente l'organigramme de notre approche.

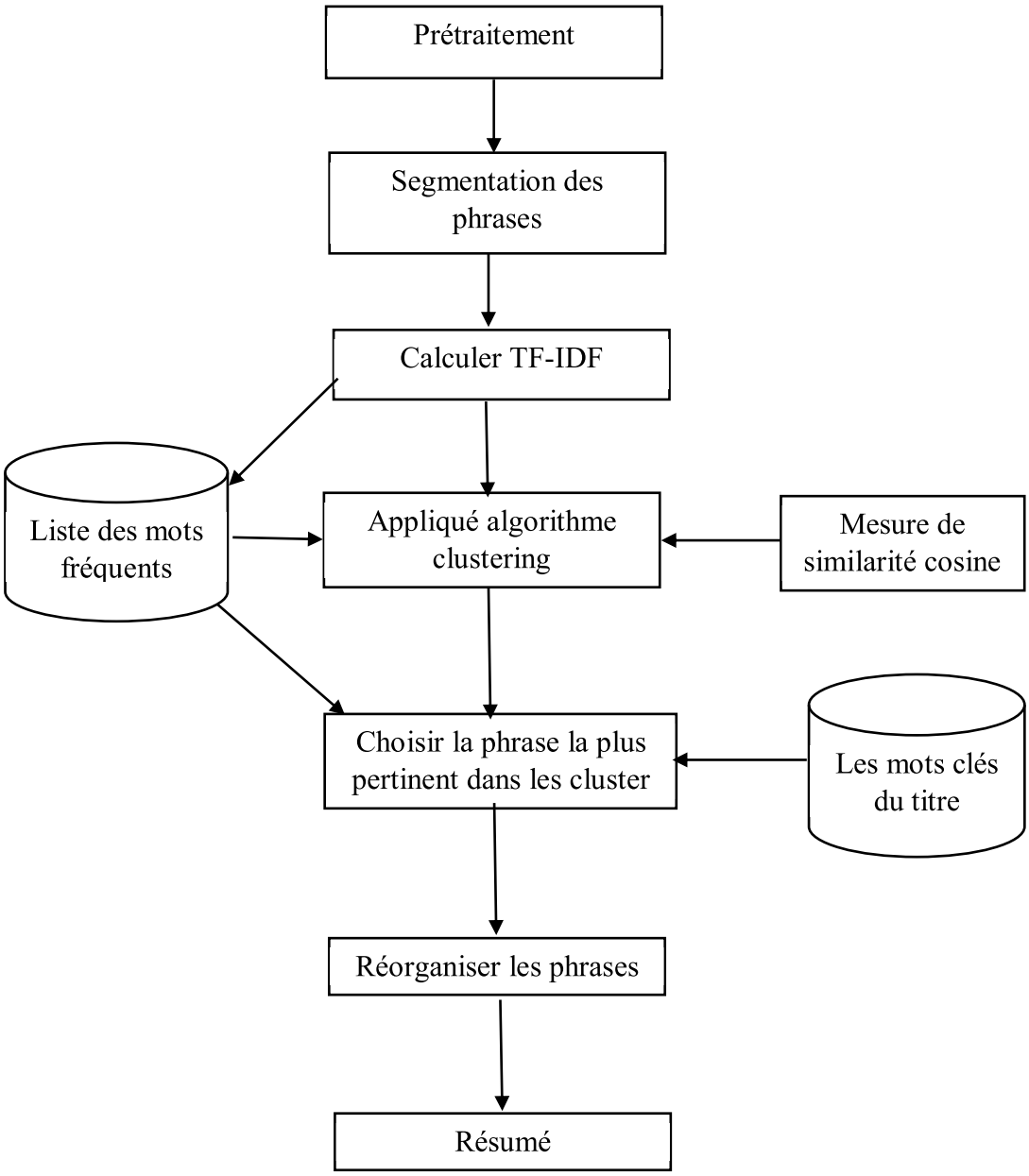


Figure 3.1: Organigramme pour le résumé.

2.1 Prétraitement :

Cette étape est indispensable pour tous algorithme de résumé automatique, elle peut être en sous tâches telles que : la segmentation du document en phrases, la tokenization, la suppression des mots vides, Racinisation et la normalisation. [10]

La figure suivante représente la phase du prétraitement

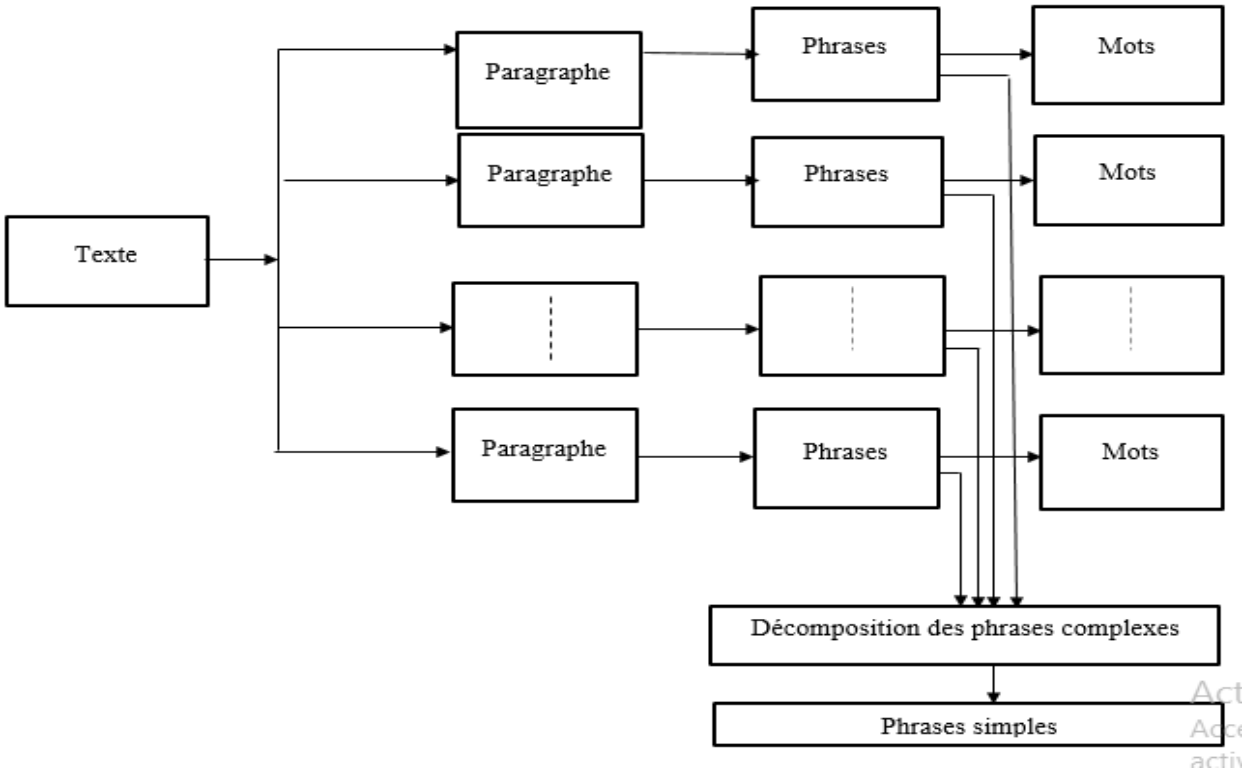


Figure 3.2: Les sous tâches de la phase prétraitement.

➢ Segmentation le document en phrases :

C'est le processus consistant à décomposer le document textuel donné dans ses phrases constitutives avec son nombre de mots. [10]

La phrase est segmentée en identifiant la limite de la phrase qui se termine par :

- ✓ L'arrêt complet (.).
- ✓ Le point d'interrogation (?).
- ✓ La marque d'exclamation (!).

- Tokenization :

Consistant à diviser les phrases en mots en identifiant les espaces, les virgules et les symboles spéciaux entre les mots. La liste des phrases et des mots est donc maintenue pour un traitement ultérieur. [10]

- Supprimé les mots vides :

Les mots d'arrêt sont des mots communs qui portent une signification moins importante que les mots-clés.

Ces mots devraient être éliminés, sinon la phrase le contenant peut influencer le résumé généré exemple (de, la, etc….). [10]

- Racinisation :

Un mot peut être trouvé sous différentes formes dans le même document. Ces mots doivent être convertis sur leur forme originale pour simplifier.

L'algorithme de dérivation est utilisé pour transformer les mots en leurs formes canoniques. Dans ce travail, on utilise le *stemmer* de Porter qui divise un mot dans sa forme racine en utilisant une liste de suffixes prédéfinie. [10]

- Normalisation :

Consiste à convertir tous les caractères dans le même type de carnet de lettres - majuscules ou minuscules. [10]

2.2 Caractéristiques d'extraction :

Une fois qu'un document de saisie est divisé en une collection de phrases ces dernières sont classées en fonction de quatre caractéristiques importantes : la fréquence, la valeur de la position des phrases, les mots clé et la similarité avec le titre. Ces caractéristiques sont les critères de sélection des phrases pertinentes. [18]

- La fréquence :

La fréquence est le nombre de fois qu'un mot se produit dans un document. Si la fréquence d'un mot dans un document est élevée, on peut dire que ce mot a un effet significatif sur le contenu du document. La valeur de fréquence totale d'une phrase est calculée en résumant la fréquence de chaque mot dans le document. Pour réaliser cette étape on a employé l'algorithme TF-IDF qui est le plus connu et le plus utilisé pour extraire les termes fréquents.

TF-IDF est l'une méthode de pondération souvent utilisée en recherche d'information et en particulier dans la fouille de textes. Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus. Des variantes de la formule originale sont souvent utilisées dans des moteurs de recherche pour apprécier la pertinence d'un document en fonction des critères de recherche de l'utilisateur. [18]

1. Calcule de TF (fréquence du terme) :

$$TF(w) = \frac{nembre\ de\ terme\ w\ dans\ le\ document}{nembre\ totale\ des\ termes\ dans\ le\ document} \quad (1)$$

2. Calculer idf (fréquence inverse de document) :

$$idf_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \quad (2)$$

ou :

- |D| : nombre total de document dans le corpus.
- $|\{d_j : t_i \in d_j\}|$: nombre de document ou le terme t_i apparait.

3. Calculer tf idf :

$$Tf - idf = tf * idf \quad (3)$$

➢ La valeur de la position des phrases :

Position de la phrase dans le texte, décide de son importance. Les phrases au début définissent le thème du document tandis que les phrases finales concluent ou résument le document. La valeur de position d'une phrase est calculée en attribuant la valeur la plus élevée à la première phrase et la valeur la plus basse jusqu'à la dernière phrase du document. Puisque la position des phrases sont importantes. On a opté à associer à chaque phrase des cordonnées *(prg,pos)* ou *prg* est le numéro du paragraphe et *pos* et la position de la phrase dans le paragraphe. [18]

- Les articulateurs :

Les articulateurs sont des expressions conjonctives (par conséquent, par conséquent, enfin, entre temps, ou d'autre part) qui relie les limites de la communication et signale les relations sémantiques dans un texte.il faut les déterminer pour décomposer les phrases complexes. [18]

- La similitude avec le titre

La similitude avec le titre comprend les mots dans les titres et les en-têtes(en cas d'un sous sommaire comme le cas des articles long, exemple les articles du site Wikipédia). Ces mots sont considérés comme ayant des poids supplémentaires dans la notation de la phrase pour la synthèse. Au plus les mots du titre sont des termes fréquents par défaut. [18]

- Position des phrases :

Dans les systèmes du résumé automatique la position des phrases dans le texte original joue un rôle primordial pour l'acquisition du sens et pour comprendre le résumé produit. Ainsi, la première et la dernière phrase sont importantes il convient toujours de les garder dans le résumé produit. [18]

- Le score des phrases :

On a ajouté ce critère pour favorisé les phrases ayant le plus nombre d'apparition des termes fréquents. Le score final d'une phrase est une combinaison linéaire de fréquence, la valeur de position de la phrase, les poids de mot clé et Similarité avec le titre du document. [18]

2.3 Phase du résumé :

Jusqu'à présent on a un ensemble de phrases et un ensemble de termes fréquents qui vont former une matrice NxM dont N est le nombre de phrases extraites et M le nombre de terme fréquents dans le document. Donc, chaque phrase est représentée par un vecteur est les cordonnées *(prg,pos)* .

Sur cette matrice on applique une méthode de la classification non supervisée. La classification non supervisé ou le clustering consiste à affecter les objets d'un ensemble de données à des groupes d'où les objets sont les plus similaires.

L'algorithme de clustering divise un groupe hétérogène de données, en sous-groupes de manière à ce que les données considérées comme les plus similaires soient associées au sein d'un groupe homogène et qu'au contraire les données considérées comme différentes se

retrouvent dans d'autres groupes distincts ; l'objectif étant de permettre une extraction de connaissance organisée à partir de ces données

Objectif d'apprentissage non supervisé est de structuration des données en classes homogènes On cherche à regrouper les points en clusters ou classes tels que les données d'un cluster soient les plus similaires possibles. [19]

Dans notre approche l'algorithme de clustering est utilisé pour :

- Regrouper les phrases porteuses de la même information.
- Le problème des phrases redondantes sera réglé car chaque cluster contient les phrases les plus similaires possibles
- Déterminer les phrases les plus pertinentes dans chaque cluster en appliquant les critères de sélections mentionnées au-dessus.

On a choisi l'algorithme K-means pour le Clustering du à sa simplicité et efficacité :

❖ K-means :

Est l'un des algorithmes d'apprentissage sans surveillance les plus simples qui résolvent le problème de clustering bien connu. La procédure suit un moyen simple et simple de classer un ensemble de données donné à travers un certain nombre de clusters (supposons k clusters) fixés apriori. L'idée principale est de définir les centres k, un pour chaque grappe. Ces centres devraient être placés d'une manière astucieuse en raison d'une situation différente cause des résultats différents. Donc, le meilleur choix est de les placer autant que possible loin les uns des autres. L'étape suivante consiste à prendre chaque point appartenant à un ensemble de données donné et à l'associer au centre le plus proche. Quand aucun point n'est en attente, la première étape est terminée et un âge de groupe précoce est terminé. À ce stade, nous devons recalculer k nouveaux centroides comme barycenter des grappes résultant de l'étape précédente. Après avoir ces k nouveaux centroïdes, une nouvelle liaison doit être effectuée entre les mêmes points de consigne de données et le nouveau centre le plus proche. Une boucle a été générée. À la suite de cette boucle, nous remarquons que les k centres changent leur emplacement étape par étape jusqu'à ce que de plus en plus de modifications soient effectuées ou, en d'autres termes, les centres ne se déplacent plus. Enfin, cet algorithme vise à minimiser une fonction objective connue sous le nom de fonction d'erreur au carré donnée par :

$$J = \sum_{j-1}^{k} \sum_{i-1}^{n} | \left| x_i^{(j)} - c_j \right| |^2 \quad (5)$$

où,

« ||Xi – Vj|| » est la distance euclidienne entre xi et vj.

« Ci » est le nombre de points de données dans i^{th} cluster.

« Ci » est le nombre de centres de cluster.

Puisque on veut regrouper les phrases les plus similaires on a besoin d'une mesure de similarité. Dans une classification non supervisé utilisant l'une des techniques classiques, la similarité entre deux phrases peut être mesurée par plusieurs métriques telles que la distance euclidienne, la distance du cosinus.

On a choisi la similarité cosinus qui permet de calculer la similarité entre deux vecteurs à N dimensions en déterminant le cosinus de l'angle entre eux. Cette métrique est fréquemment utilisée en fouille de textes. [19]

Soit deux vecteurs A et B, l'angle cos θ s'obtient par le produit scalaire et la norme des vecteurs :

La similarité obtenue : $sim\ \text{cosinus}(\theta) \in [0,1]$

$$\cos(\theta) = \frac{A.B}{||A||.||B||} \quad (4)$$

2.4 La réorganisation des phrases :

A ce stade on a plusieurs clusters et pour chaque cluster on a choisi une ou plusieurs phrases pertinentes. On a un ensemble non ordonnées de phrases. Le résumé sera donc la réorganisation des phrase en fonction des cordonnées (pgr,pos) et les critères de sélection précisément le critère de position de la phrase dans le texte originale. [16]

3. Variante pour le système :

Pour une variante de notre système de résumé automatique, on propose de réaliser un résumé multi-document : C'est une procédure automatique conçue pour extraire l'information de plusieurs documents textuels écrits sur le même sujet. Le résumé résultant permet aux utilisateurs individuels, tels que les consommateurs d'informations professionnelles, de se familiariser rapidement avec les informations contenues dans une vaste collection de

documents. L'approche adoptée restera la même, la différence réside que l'ensemble des phrases est créé depuis l'ensemble des documents.

Cependant, La tâche de résumé multi-document est beaucoup plus complexe que de résumer un document et c'est une très grande tâche. Cette difficulté résulte de la diversité thématique dans un vaste ensemble de documents.

Une bonne technologie de synthèse vise à combiner les principaux thèmes avec l'exhaustivité, la lisibilité et la concision. La mise en œuvre actuelle comprend le développement d'une technique extractive qui combine le premier regroupement de documents et ensuite le regroupement de phrases dans ces documents. Les résultats obtenus sont remarquables en termes d'efficacité et de réduction de la redondance dans une large mesure.

4. Implémentations :

Pour réaliser notre approche on a utilisé les outils suivants :

4.1. Netbeans :

NetBeans est un environnement de développement intégré (IDE), placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML.

Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Développement Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plateforme. L'IDE Netbeans s'enrichit à l'aide de plugins. [14]

4.2. Java :

Java est le dernier né des langages de programmation orientée objet. Il n'est pas un journal relatif à l'informatique qui ne présente régulièrement des articles concernant le langage Java. Java envahit l'industrie et les projets sont nombreux. Il y a peu de projets réels

mais beaucoup de prototypes de validation de la technologie. Le deuxième symposium Java qui s'est tenu à Berlin fin 1997 a vu la présentation de nombreux produits. Les constructeurs et les sociétés de logiciel s'investissent dans la technologie Java. Aucun ne colloque concernant la programmation orientée objet ne peut faire l'impasse sur Java. Enfin, Java conquiert les universités et les écoles où sa publicité, alliée à une certaine simplicité de programmation pour des résultats spectaculaires, induit la demande des élèves. Nous allons présenter Java, sa technologie et examiner objectivement les raisons de ce succès.

Le langage Java et ses concepts ont été développés par Sun Microsystems Inc. Les concepts ont été pour l'essentiel empruntés aux langages orientés objet existants. Un site Web Maintient à jour les informations sur Java, son historique et ses nouveaux développements. Ces derniers sont si rapides qu'il est pratiquement impossible d'obtenir une information à jour sans la consultation régulière de ce genre de sites. Même si Java n'est pas exclusivement un langage de développement pour Internet, il a grandement facilité la programmation répartie, celle des interfaces graphiques et celle des « applets », petits programmes exécutés par les « Navigateurs Internet ». De fait, la vie de Java se passe sur Internet. C'est là qu'il faut chercher les dernières nouveautés, demander des aides en cas de doute, rechercher les derniers outils de développement ou les bibliothèques récentes.

Le concept Java a été développé par l'équipe de James Gosling de Sun Microsystems Inc. dans le but d'obtenir des programmes téléchargeables et indépendants du support d'exécution, machine et système d'exploitation. Le marché ciblé est celui des machines portables de faible capacité mais pouvant être connectées au réseau.

Il est aussi celui des navigateurs Web qui ont besoin d'interactivité pour n'être plus de simples afficheurs d'informations multimédias. [16]

4.3. OpenNLP :

La bibliothèque Apache OpenNLP est une boîte à outils fondée sur l'apprentissage par machine pour le traitement du texte en langage naturel. Il comprend un détecteur de phrases, un tokenizer, un identifiant de nom, un tagger de pièces de discours (POS), un broyeur et un analyseur. Il a de très bonnes API qui peuvent être facilement intégrées à un programme Java. Cependant, la documentation contient des informations non-actualisées. [15]

On peut citez quelques exemples :

- Détecteur de phrases:

Le détecteur de phrases est destiné à détecter les limites des phrases. Compte tenu du paragraphe suivant :

```
Hi. How are you? This is Mike.
```

Le détecteur de phrases renvoie un ensemble de chaînes. Dans ce cas il y a deux phrases :

```
Hi. How are you?
This is Mike.
```

❖ Exemple de code :

```
public static void SentenceDetect() throws InvalidFormatException,
                IOException {
        String paragraph = "Hi. How are you? This is Mike.";

        // always start with a model, a model is learned from training data
        InputStream is = new FileInputStream("en-sent.bin");
        SentenceModel model = new SentenceModel(is);
        SentenceDetectorME sdetector = new SentenceDetectorME(model);

        String sentences[] = sdetector.sentDetect(paragraph);

        System.out.println(sentences[0]);
        System.out.println(sentences[1]);
        is.close();
}
```

Figure 3.3: Exemple de code de détecteur des phrases.

- Tokenizer:

Les jetons sont généralement des mots qui sont séparés par l'espace, mais il existe des exceptions. Par exemple, "n'est pas" est divisé en "est" et "non, car il s'agit d'un bref format de" n'est pas ". Notre phrase est séparée dans les jetons suivants :

```
Hi
.
How
are
you
?
This
is
Mike
.
```

❖ Exemple de code :

```java
public static void Tokenize() throws InvalidFormatException, IOException {
        InputStream is = new FileInputStream("en-token.bin");

        TokenizerModel model = new TokenizerModel(is);

        Tokenizer tokenizer = new TokenizerME(model);

        String tokens[] = tokenizer.tokenize("Hi. How are you? This is Mike.");

        for (String a : tokens)
                System.out.println(a);

        is.close();
}
```

Figure 3.4: Exemple de code de tokenizer.

3.4 Présentation de l'application :

- On a l'interface du mode de mono-document qui fait le résumé de mono-document :

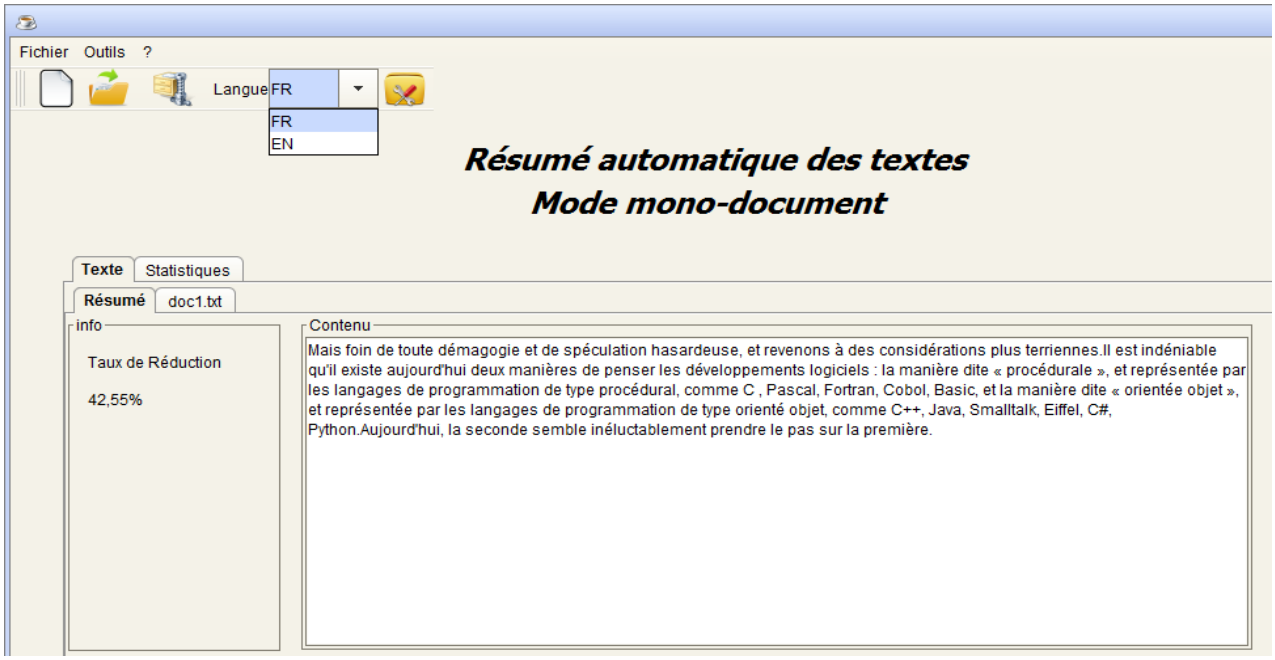


Figure 3.5: Résumé automatique de mono-document.

- L'interface du multi-document qui fait un résumé de multi-document :

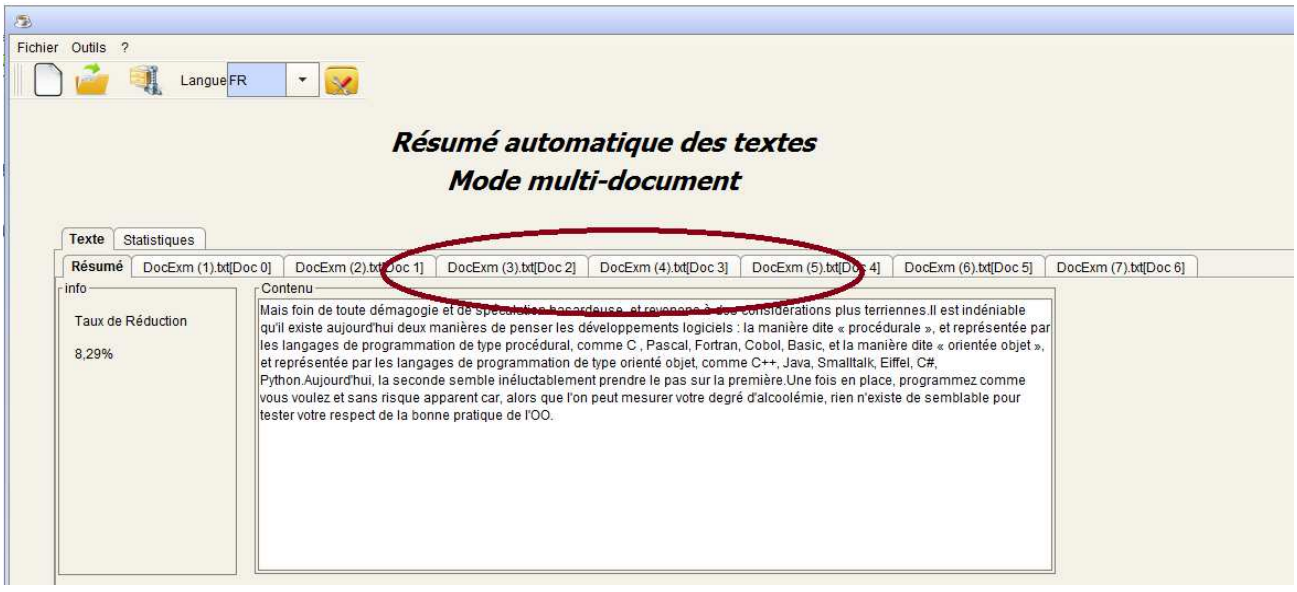
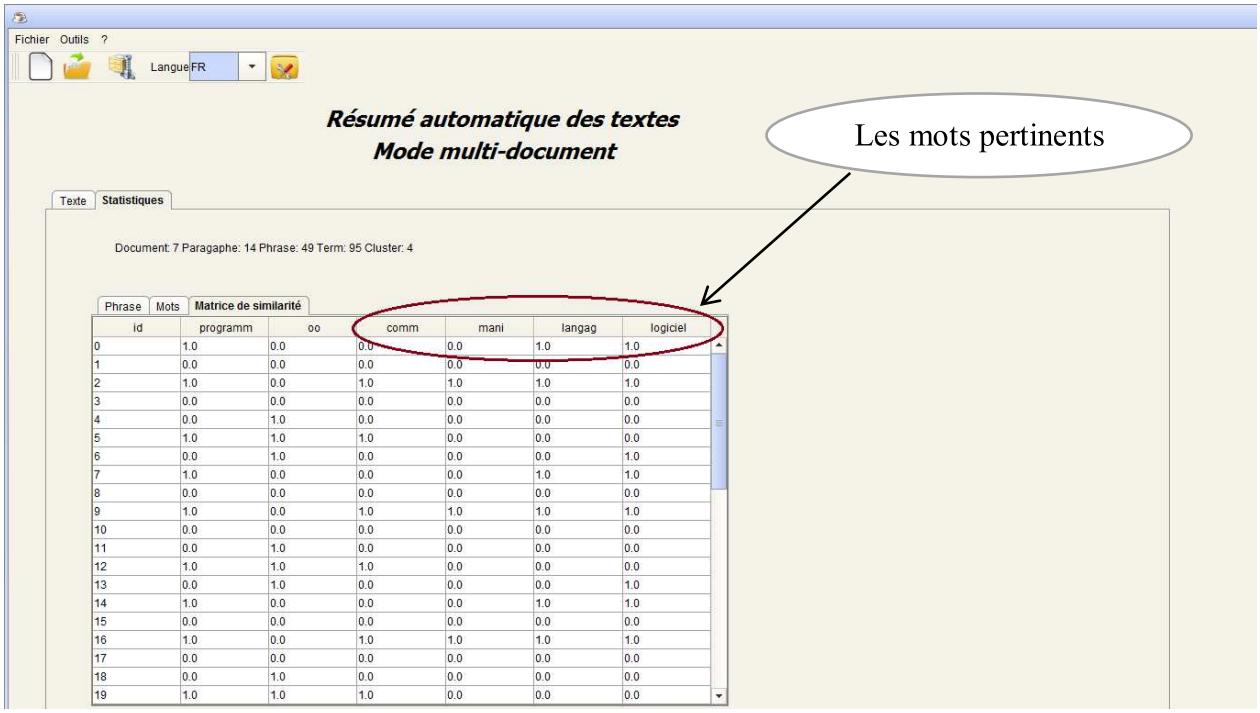


Figure 3.6: Résumé automatique de multi-document

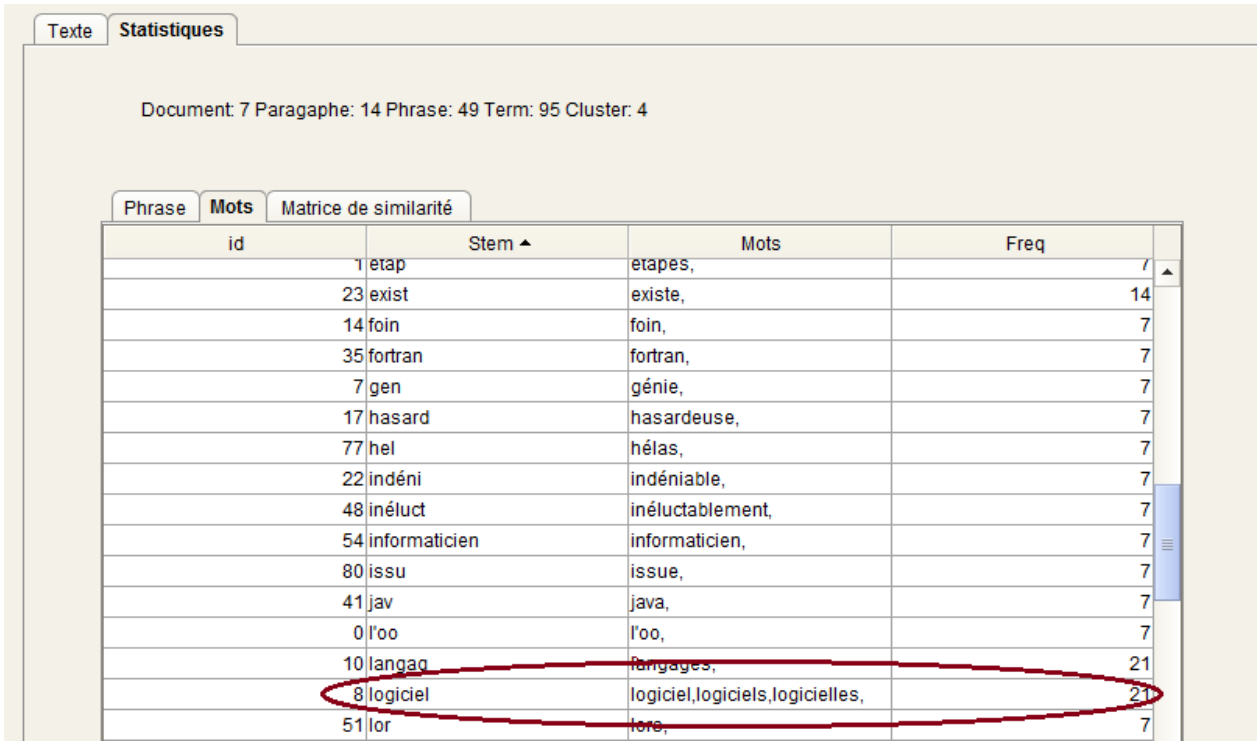
❖ L'interface de matrice de similarité :



id	programm	oo	comm	mani	langag	logiciel
0	1.0	0.0	0.0	0.0	1.0	1.0
1	0.0	0.0	0.0	0.0	0.0	0.0
2	1.0	0.0	1.0	1.0	1.0	1.0
3	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	1.0	0.0	0.0	0.0	0.0
5	1.0	1.0	1.0	0.0	0.0	0.0
6	0.0	1.0	0.0	0.0	0.0	1.0
7	1.0	0.0	0.0	0.0	1.0	1.0
8	0.0	0.0	0.0	0.0	0.0	0.0
9	1.0	0.0	1.0	1.0	1.0	1.0
10	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	1.0	0.0	0.0	0.0	0.0
12	1.0	1.0	1.0	0.0	0.0	0.0
13	0.0	1.0	0.0	0.0	0.0	1.0
14	1.0	0.0	0.0	0.0	1.0	1.0
15	0.0	0.0	0.0	0.0	0.0	0.0
16	1.0	0.0	1.0	1.0	1.0	1.0
17	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	1.0	0.0	0.0	0.0	0.0
19	1.0	1.0	1.0	0.0	0.0	0.0

Figure 3.7: Matrice de similarité.

❖ Cette interface représente racinisation des mots :



id	Stem	Mots	Freq
1	etap	etapes,	7
23	exist	existe,	14
14	foin	foin,	7
35	fortran	fortran,	7
7	gen	génie,	7
17	hasard	hasardeuse,	7
77	hel	hélas,	7
22	indéni	indéniable,	7
48	inéluct	inéluctablement,	7
54	informaticien	informaticien,	7
80	issu	issue,	7
41	jav	java,	7
0	l'oo	l'oo,	7
10	langag	langages,	21
8	logiciel	logiciel,logiciels,logicielles,	21
51	lor	lore,	7

Figure 3.8: Racinisation des mots.

❖ Cette interface représente les phrases :

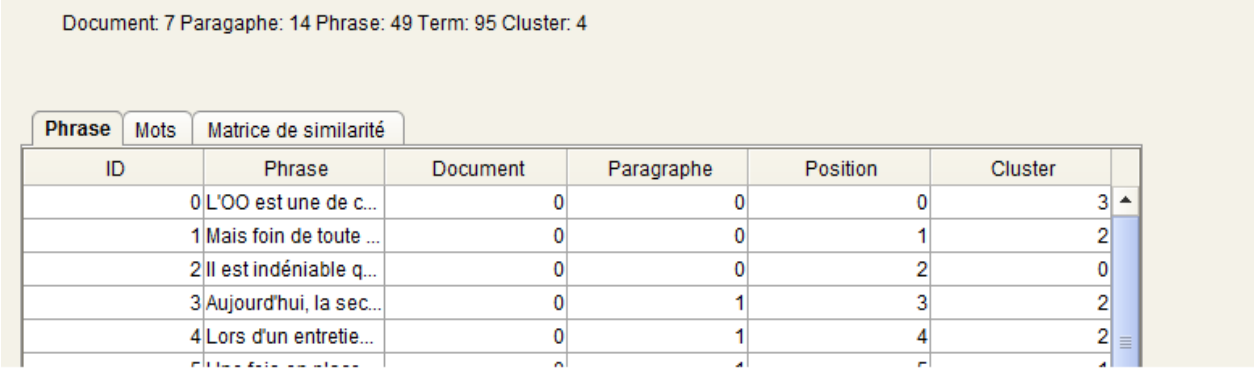


Figure 3.9: Tableau des phrases.

5. Conclusion :

Dans ce chapitre nous avons présenté notre approche en détaille pour le résumé mono-document et multi-document, les techniques de résumé automatique adoptée et les différents outils utilisés pour le développement de notre système du résumé automatique : OpenNLP, NetBeans ….

Notons que le résumé automatique est une tâche très difficile pour être informatisé car plusieurs disciplines jouent un rôle primordial pour aboutir à un résumé condensé, lisible et compréhensible pour un être humain.

CONCLUSION GENERALE

Le résumé automatique est l'une des activités de traitement automatique de langue naturelle les plus ciblé par les recherches ces dernières années. Dans ce mémoire on a présenté, une approche pour le résumé automatique des textes en langue française et anglaise. Si un résumé est disponible, le lecteur n'a pas besoin de lire l'intégralité du document en effet il a une idée sur le contenu de document. La majorité des systèmes de résumé qui existent, garde les phrases pertinentes pour former le résumé final en leur totalité. A travers notre système on a tenté de réduire les phrases malgré le risque de perte d'informations et de cohérence entre les phrases.

On a choisi l'approche par extraction et en appliquant la méthode de classification non supervisée ou le clustering consiste à affecter les objets d'un ensemble de données à des groupes d'où les objets sont les plus similaires, en particulier l'algorithme de K-means pour créer des groupes de phrases similaires qui ont été utilisées, Ensuite à partir des groupes de phrases : la phrase la plus représentative a été choisie pour composer le résumé.

L'approche proposée, contrairement aux méthodes supervisées n'a pas besoin de plusieurs échantillons d'or pour la formation ; Par conséquent, notre approche proposée est plus indépendante du langage et du domaine.

Cependant, on a rencontré des difficultés, le problème majeur est surtout l'absence des corpus types open source pour une éventuelle comparaison et vérification, pour décider si le résumé a un sens par apport aux résumés des corpus ou non.

Il est important de noter qu'on est toujours loin de produire des résumés comparables à ceux produits par les humains. Mais comme perspective on propose de faire la synthèse de plusieurs documents pour donner l'idée générale en un seul document.

Articles :

[1] A, Farzindar., & Roche, M. (2013). Traitement automatique des langues. Revue TAL-Traitement Automatique des Langues, 54(3), 1-73.

[2] BASAGIC, R., KRUPIC, D., & SUZIC, B. (2009). Automatic text summarization. Information Searches and Retrieval, WS.

[3] COLLOBERT, R., WESTON, J., BOTTOU, L., KARLEN, M., KAVUKCUOGLU, K., & KUKSA, P. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(Aug), 2493-2537.

[4] Cori, M., & Léon, J. (2002). La constitution du TAL. TAL, 43(3), 21-55.

[5] GAHBICHE-BRAHAM, S. (2013). Amélioration des systèmes de traduction par analyse linguistique et thématique : application à la traduction depuis l'arabe (Doctoral dissertation, Université Paris Sud-Paris XI).

[6] GILLOUX, M. (1989, May). Traitement automatique des langues naturelles. In Annales des télécommunications (Vol. 44, No. 5-6, pp. 301-316). Springer-Verlag.

[7] LEBARBIER, E., & MARY-HUARD, T. (2008). Classification non supervisée.

[8] NENKOVA, A., & MCKEOWN, K. (2011). Automatic summarization. Foundations and Trends® in Information Retrieval, 5(2–3), 103-233.

[9] NETO, J. L., FRRITAS, A. A., & KAESTNER, C. A. (2002, November). Automatic text summarization using a machine learning approach. In Brazilian Symposium on Artificial Intelligence (pp. 205-215). Springer Berlin Heidelberg.

[10] PAL, A. R., MAITI, P. K., & SAHA, D. (2013). An Approach to Automatic Text Summarization Using Simplified Lesk Algorithm and WordNet. International Journal of Control Theory and Computer Modeling (IJCTCM) Vol, 3.

[11] TANNIER, X. (2008). Traitement Automatique des Langues.

[12] YVON, F. (2006). Des apprentis pour le traitement automatique des langues. Mémoire d'habilitation à diriger des recherches, Université Pierre et Marie Curie, Paris.

[13] YVON, F. (2010). Une petite introduction au traitement automatique des langues naturelles. In Conference on Knowledge discovery and data mining (pp. 27-36).

[14] Benson, C., Muller-Prove, M., & Mzourek, J. (2004, April). Professional usability in open source projects: GNOME, OpenOffice. Org, NetBeans. In CHI'04 extended abstracts on Human Factors in Computing Systems (pp. 1083-1084). ACM.

[15] Deshpande, A. R., & Lobo, L. M. R. J. (2013). Text summarization using Clustering technique. International Journal of Engineering Trends and Technology, 4(8).

[16] Agrawal, A., & Gupta, U. (2014). Extraction based approach for text summarization using k-means clustering. International Journal of Scientific and Research Publications, 4(11), 1.

Ouvrages :

[17] TORRES-MORENO, J. M. (2011). Résume automatique de documents. Lavoisier.

[18] Arnold, K., Gosling, J., Holmes, D., & Holmes, D. (2000). The Java programming language (Vol. 2). Reading: Addison-wesley.

Mémoire:

[19] Keskes, I., Boudabous, M. M., Maaloul, M. H., & Belguith, L. H. (2012, June). Étude comparative entre trois approches de résumé automatique de documents arabes. In Actes de la conférence conjointe JEP-TALN-RECITAL.

الملخص:

تقنيات معالجة اللغة تتطور بشكل لافت للانتباه حيث نجد العديد من الأبحاث في هذا المجال تغطي مختلف الأنشطة منها التلخيص التلقائي. في هذا السياق يندرج عملنا، حيث نقترح نهجا يستند على اللسانيات واللغويات وتكنولوجيا البحث في البيانات لتقديم أفضل شكل مختصر لمستند ما. لتحقيق هذا الهدف، استخدمنا طريقة التجميع Clustering، ومقياس التشابه بالإضافة لمعايير اقتناء اخرى لاختيار الجمل الأكثر اهمية لتشكيل الملخص النهائي.

الكلمات المفاتيح: التلخيص التلقائي، طريقة التجميع Clustering، مقياس التشابه

Abstract:

The naturally language processing technics evolves in a surprising way. Several researches are carried out in this field covering its multiple activities such as the automatic summary. In this context our work integrates, we propose an approach by extraction based on linguistics and data mining techniques to present the best condensed form of a given document. To achieve this objective, the Clustering method, a measure of similarity and selection criteria were used to choose the most relevant sentences forming the product summary.

Keywords: automatic summary, extraction method, clustering, similarity measure.

Résumé :

Les techniques du traitement automatique de langage naturel évoluent d'une façon étonnante. Plusieurs recherches sont menues dans ce domaine couvrant ses multiples activités telles que le résumé automatique. Dans ce contexte notre mémoire s'intègre, on propose une approche d'extraction basée sous la linguistique et la technique du Data Mining pour présenter la meilleure forme condensée d'un document donné. Pour atteindre cet objectif on a utilisé la méthode du Clustering, une mesure de similarité et des critères de sélection pour choisir les phrases les plus pertinentes formant le résumé produit.

Mots-clés : résumé automatique, méthode par extraction, Clustering, mesure de similarité.