| | |
|---|---|
| **Titre:** Title: | Étude comparative de méthodes heuristiques et syntaxiques pour la production automatique de résumés |
| **Auteur:** Author: | Samuel Côté-Bérubé |
| **Date:** | 2005 |
| **Type:** | Mémoire ou thèse / Dissertation or Thesis |
| **Référence:** Citation: | Côté-Bérubé, S. (2005). Étude comparative de méthodes heuristiques et syntaxiques pour la production automatique de résumés [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7605/ |

UNIVERSITÉ DE MONTRÉAL

# ÉTUDE COMPARATIVE DE MÉTHODES HEURISTIQUES ET SYNTAXIQUES POUR LA PRODUCTION AUTOMATIQUE DE RÉSUMÉS

SAMUEL CÔTÉ-BÉRUBÉ
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AOÛT 2005

*Your file Votre référence*
*ISBN: 978-0-494-16771-7*
*Our file Notre référence*
*ISBN: 978-0-494-16771-7*

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

# ÉTUDE COMPARATIVE DE MÉTHODES HEURISTIQUES ET SYNTAXIQUES POUR LA PRODUCTION AUTOMATIQUE DE RÉSUMÉS

présenté par: CÔTÉ-BÉRUBÉ Samuel

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. ROY Robert, Ph.D., président,

M. GAGNON Michel, Ph.D., membre et directeur de recherche,

M. LABIB Richard, Ph.D., membre.

# REMERCIEMENTS

Je remercie d'abord mon directeur de recherche, M. Michel Gagnon, sans qui je n'aurais su avancer dans ce projet. Il m'a donné le goût de la recherche en plus d'être d'un grand support autant financier que moral. Il a été un directeur très présent, toujours disponible et accueillant, même lorsque surchargé de travail. Il m'a communiqué ses idées et ses réflexions qui m'ont été fort utiles tout au long de mes travaux.

Je remercie ensuite le professeur M. Juan-Manuel Torres-Moreno de l'Université d'Avignon pour son aide dans les résumés de divers textes que je lui ai soumis. Je remercie ma conjointe, Mme Audrée Cloutier qui m'a grandement aidé dans la correction du français dans le présent mémoire ; ainsi qu'une amie, Mme Annie Turner, qui dans le cadre d'un cours de bibliothéconomie à l'Université de Montréal m'a aidé de manière professionnelle dans ma recherche de documents.

Je remercie finalement mes parents, famille et amis pour leur support moral. Ils ont su m'encourager à persévérer et à atteindre mes objectifs.

# RÉSUMÉ

Avec la multiplicité de documents disponibles sous forme électronique (documents scientifiques, périodiques, quotidiens, romans), il devient impératif pour quiconque s'intéressant à la lecture de trouver un moyen d'assimiler les informations sans lire tous ces documents intégralement. Nous avons à Polytechnique un logiciel baptisé CORTEX, qui permet d'extraire les phrases significatives de textes de tous genres. Pour bien fonctionner, CORTEX utilisait jusqu'à maintenant dix métriques basées principalement sur la fréquence des mots.

Dans ce mémoire, nous expérimentons diverses études comparatives dans le but d'en arriver à améliorer les performances de CORTEX. Nous proposons d'abord une utilisation différente des métriques déjà implantées, en discriminant certaines d'entre elles. Nous proposons aussi d'ajouter une onzième métrique basée sur la disposition des phrases dans le document. Grâce à ces essais, nous sommes arrivés à améliorer la pertinence des phrases sélectionnées automatiquement.

Les diverses études comparatives comprennent : l'évaluation d'une technique visant à remplacer les pronoms par leur référent, un choix automatique des différentes métriques qui étaient jadis systématiquement utilisées simultanément, et finalement, l'utilisation d'un analyseur syntaxique dans le but de filtrer et/ou bonifier certaines catégories de mots ou de relations entre les mots. Ces études, sans fournir de résultats extraordinaires, ont permis d'identifier quelques avenues intéressantes et ont permis de prendre conscience des limites de l'application des méthodes statistiques à un logiciel de production automatique de résumés.

## ABSTRACT

With the multiplicity of documents available in electronic form (scientific documents, journals, newspapers, novels), it becomes imperative for whoever being interested in reading to find a means of assimilating information without reading completely all these documents. We have at Polytechnique a software named CORTEX, which extracts the most relevant sentences from texts of any kind. For its operation, CORTEX used ten metrics based mainly on the frequency of words.

In this master thesis, we carried out various comparative studies with the objective of identifying the best ways to improve the performances of CORTEX. Initially, we propose a different use of the metrics already implemented, by discriminating some of them. In addition, we propose the use of an additional metric based on the position of the sentences in the document. These tests helped us to improve the relevance of the automatically selected sentences.

Our various comparative studies include : the evaluation of a technique aiming to replace the pronouns by their referent, an automatic choice of differents metrics which were formerly used simultaneously, an finally, the use of a syntactic parser with the aim of filtering and/or improving some words categories or relations between the words. These studies, without providing extraordinary results, help to identify new interesting ideas and made us more convicted of the limits of statistical methods for automatic summarization.

# TABLE DES MATIÈRES

# LISTE DES FIGURES

# LISTE DES TABLEAUX

# LISTE DES ANNEXES

# INTRODUCTION

Depuis environ une quinzaine d'années, des études sont effectuées pour tenter de trouver un moyen d'obtenir, sans intervention humaine, un résumé de texte fidèle au document original. Nous retrouvons, dans la plupart des publications, une section où l'auteur résume le contenu de son discours en quelques lignes afin d'informer le lecteur sur les idées ou le contenu de son œuvre. Qu'arriverait-il si, au lieu de se donner le mal de produire manuellement ce résumé, nous pouvions arriver à utiliser un outil complètement automatisé qui permettrait d'obtenir un résultat de qualité tout à fait équivalent ? Il serait possible de consulter et de comprendre le contenu d'une plus grande quantité d'articles, de livres ou autres publications sans avoir à revoir le contenu en entier, et ce peu importe si ce document est accompagné d'un résumé.

Malheureusement, cette réalité est encore très lointaine. La technologie actuelle ne permet pas d'espérer un tel résultat, ni à court, ni à moyen terme. Présentement, les différents logiciels disponibles admettent, dans le meilleur des cas, l'extraction depuis le texte original de certaines phrases ou segments de phrases, sans reformulation. Les résumés sont obtenus en juxtaposant l'un à l'autre les segments retenus.

De toute évidence, la qualité des résumés ainsi produits est encore très loin de pouvoir concurrencer celle des résumés faits manuellement. De plus, lire des phrases mises ensembles de cette manière peut parfois donner un résultat dépourvu de sens. Quoiqu'il en soit, en attendant que la technologie soit assez mûre pour en arriver aux résultats mentionnés ci-haut, nous pouvons nous servir de l'extraction de phrases que nous possédons actuellement pour d'autres applications. Les condensés de textes peuvent ainsi servir pour des fins telles : classification de documents, annotation automatique, logiciels de questions-réponses, etc. Bref, tout traitement

relativement long à s'exécuter qui pourrait tirer avantage à recevoir en entrée un texte réduit plutôt qu'un document entier.

## 0.1 Définitions et concepts de base

Au cours des dernières années, plusieurs types de logiciels de production automatique de résumés ont fait l'objet d'expériences. Il existe une vaste étendue de genres de documents ou de techniques qu'il est possible d'utiliser dans le but d'arriver à résumer un texte. Nous verrons dans un premier temps les diverses catégories qui existent actuellement, elles seront ensuite séparées et expliquées dans le but de mieux cibler les chapitres qui vont suivre.

### 0.1.1 Résumé par extraction

Lorsqu'il est question de résumés automatiques dans la littérature, nous pouvons faire référence à ce que nous appellons un abrégé (abstract) ou un extrait (extract). Dans le premier cas, nous parlons d'un type de condensé se rapprochant beaucoup du résumé humain. Le logiciel retire les idées principales du texte et reformule les phrases pour obtenir un texte plus court, suivi et cohérent, tout en conservant l'information pertinente. Dans le deuxième cas, le logiciel tente de trouver les phrases possédant le plus de valeur dans le texte et produit un résumé en réécrivant intégralement les phrases retenues. Bien que la première idée semble la meilleure, elle reste aussi beaucoup plus complexe et peu de recherches qui y sont consacrées aboutissent à de bons résultats. La raison parait évidente, les recherches n'en sont encore qu'à leurs balbutiements face à l'extraction d'information, difficile donc d'en produire une reformulation. Le résumé par extraction est par conséquent beaucoup plus étudié puisqu'il constitue lui aussi un défi de taille, mais beaucoup plus réaliste

pour la technologie actuelle. Pour ces raisons, nous traiterons dans ce mémoire de résumés par extraction, et ce, en utilisant uniquement le terme « résumé ».

### 0.1.2 Type de documents traités

Ce ne sont pas tous les logiciels de condensation qui sont destinés au même type de document. Nous rencontrons souvent des logiciels qui ciblent une seule catégorie bien précise : articles de journaux et documents scientifiques sont, par exemple, deux domaines régulièrement ciblés. Nous verrons aux section 1.2.2.1 et 1.2.3.5 les logiciels SweSum et Pertinence Summarizer qui visent, entre autres, ces domaines. Quelle est l'impact du genre de document pour l'élaboration du résumé ? Si nous comparons des textes provenant d'articles de journaux aux dialogues d'un conte pour enfant, la structure de l'écriture sera bien différente : les phrases clés ne se retrouveront pas au même endroit d'un type de document à l'autre. Chacun des genres de documents possède une structure d'informations qui lui est propre et qui peut s'avérer fort utile dans le processus de condensation. Par exemple, un logiciel de résumés se concentrant principalement sur les articles de journaux peut facilement reprendre intégralement les quelques premières phrases de l'article qui contiennent très souvent l'essentiel de l'information à transmettre.

Certains logiciels tiendront compte de ces différences, dont certains seront cités plus bas, et d'autres non. Il s'agit en fait d'un choix important lors de la conception. Si nous prenons en considération la catégorie de documents à traiter, d'un côté nous limitons le logiciel à un genre bien précis, mais d'un autre côté nous possédons une précieuse information additionnelle qui aide à l'obtention d'un condensé de qualité. À l'opposé, si nous ne tenons pas compte de la nature du texte, nous perdons cette précieuse information ; cependant le logiciel devient indépendant du genre de document et par le fait même beaucoup plus flexible. Les logiciels qui

ne dépendent pas du genre de document ne peuvent pas profiter des structures de documents tel la position des phrases importantes dans le texte, ou encore de mots clés fréquent dans certaines disciplines. Ils vont devoir davantage se baser sur d'autres métriques, en se dirigeant beaucoup plus vers des techniques statistiques ou symboliques ; ces techniques seront abordées à la section 1.3. Nous pourrons aussi retrouver dans certains logiciels, tels PERTINENCE SUMMARIZER ou SWESUM dont nous reparlerons aux sections 1.2.3 et 1.2.2, une requête à l'utilisateur avant le début du traitement, lui demandant d'identifier le type de document.

## 0.2 Problématique

La qualité des condensés automatiques traîne encore très loin derrière celle des résumés produits manuellement. Nous nous en rendons immédiatement compte lorsque nous nous attardons à analyser les résultats produits par ces technologies : les textes produits sont perfectibles. Quelles techniques peuvent être utilisées ? Quelles sont les meilleures ? Pouvons-nous hybrider certaines techniques entre elles ? Voilà des questions auxquelles la communauté scientifique n'a su apporter de réponses claires à ce jour. Bon nombre de recherches sont menées dans le monde et chacune explore un cheminement précis avec ses techniques particulières. Malheureusement, encore aujourd'hui, aucune conclusion unanime n'est tirée. Des pistes de solutions peuvent tout de même fournir des avenues de recherches ou des techniques reconnues pour donner de bons résultats.

Sparck-Jones (Sparck-Jones, 1999) souligne l'importance de cibler stratégiquement chacune des étapes du processus afin de développer des méthodes efficaces. Selon elle, la problématique intrinsèque du résumé automatique, entremêlée avec une multitude d'autres facteurs tel l'évaluation, ou la forme des documents à résumer ou à produire, constituent des défis de taille et elle considère que les techniques sta-

tistiques actuelles sont très limitées. Ces limites des méthodes statistiques peuvent-elles être surmontées par des techniques simples ? C'est la question à laquelle nous tenterons de répondre dans ce mémoire.

## 0.3 Objectifs

Notre objectif principal est d'évaluer la portée de différentes méthodes pour la production automatique de résumés. Parmi ces méthodes, nous retrouvons :

- l'implémentation d'une technique de recherche d'information basée sur la position d'une phrase au sein du texte,
- traitement anaphorique visant à remplacer les pronoms par leur référent,
- discrimination de certaines métriques dans des situations particulières, et
- utilisation d'une analyse grammaticale dans le but d'avantager ou de discriminer certaines relations entre les mots.

Notre travail permettra de voir dans quelle mesure ces méthodes améliorent un logiciel de production automatique de résumés basé sur des méthodes statistiques.

## 0.4 Méthode utilisée

À partir d'un logiciel existant qui implémente des méthodes statistiques, certains modules ont été ajoutés afin de réaliser les différents tests. Dans le but de pouvoir comparer l'influence des techniques suggérées, nous avons utilisé onze textes de genres différents (articles scientifiques, romans, etc.). Des évaluateurs humains ont sélectionné de 15 à 20% des phrases les plus pertinentes, pour chacun des textes, afin de former un résumé. Le logiciel retient quant à lui 10% du nombre de phrases initial. Plus le résumé produit automatiquement possédera de phrases communes avec la sélection des évaluateurs humains, plus sa performance sera jugée élevée.

C'est en comparant la différence des résultats pour chacun des traitements qu'il sera possible de discuter de l'efficacité des techniques sur les divers textes.

## 0.5 Plan du mémoire

Ce mémoire présente d'abord l'état de l'art afin de mettre en contexte la recherche effectuée. Viennent ensuite en détail les choix effectués quant au logiciel utilisé, ainsi que les raisons de ces choix. Les chapitres suivants sont consacrés aux méthodes employées : les améliorations apportées à CORTEX grâce à l'ajout d'une métrique et la sélection de la meilleure combinaison de métriques, le traitement des pronoms, l'emploi d'une heuristique pour un choix automatique des métriques, et finalement l'utilisation d'un pré-traitement par un module d'analyse syntaxique. Ce mémoire se termine avec une conclusion sur le travail effectué et l'identification de travaux de recherche futurs.

# CHAPITRE 1

# ÉTAT DE L'ART

Dans ce chapitre, il sera question de quelques recherches antérieures effectuées sur le sujet de la production automatique de condensés de textes. Tout d'abord, une introduction rapide sera présentée à propos de certains travaux qui ont marqué l'évolution du domaine ; ensuite, une seconde section traitera des différents logiciels existants suivie d'une revue de quelques articles précis du domaine.

## 1.1 Historique

Les scientifiques ont commencé à s'intéresser aux résumés automatiques dès la fin des années 50 alors que le premier article sur le sujet paraissait dans le IBM Journal (Luhn, 1958). À l'époque, Luhn apportait l'idée de se baser sur les mots les plus fréquents du texte pour bâtir un condensé automatiquement. Cet article se voyait alors pionnier dans le domaine, mais l'idée de Luhn fut rapidement abandonnée. Le manque d'intérêt de la communauté scientifique était provoqué principalement par la faible disponibilité du matériel (ordinateurs), mais aussi par la quasi absence de documents numériques ; ceci eut pour conséquence que, sur une certaine période, on ne retrouve presque aucun travail de recherche. En effet, pendant environ 30 ans, peu de contributions ont été faites dans ce domaine ; on en compte malgré tout quelques-unes d'importance dont la publication d'Edmundson (Edmundson, 1969) qui comparait quatre méthodes de production de résumés dont trois nouvelles à l'époque. L'idée de se baser sur les mots les plus fréquents du texte était déjà connue grâce aux recherches de Luhn, mais Edmundson apporta les trois nouvelles

idées suivantes encore utilisées à ce jour : se servir des mots du titre pour donner du poids aux phrases qui contiennent ces mêmes termes, repérer des expressions indicatives reconnues généralement pour introduire des idées importantes, et finalement, regarder la position des phrases dans le texte. Malgré ces nouvelles pistes apportées à la fin des années 60, ce n'est qu'au début des années 90 que la production automatique de résumés a réellement été reprise sérieusement. Avec l'avènement d'Internet et la multitude de documents désormais disponibles en ligne, le besoin d'acquérir de plus en plus d'informations en peu de temps se fait grandissant et l'idée initiale de Luhn est devenue intéressante. Les études sur la production automatique de résumés se sont donc multipliées à partir de ce moment, et plusieurs avenues sont maintenant explorées par les équipes de recherches présentes partout dans le monde.

## 1.2 Logiciels existants de production automatique de résumés

Il existe maintenant un grand nombre de logiciels destinés à la production automatique de résumés. Il s'agit d'un domaine de recherche qui intéresse à la fois les entreprises et les institutions académiques ; de ce fait, nous retrouvons deux types de logiciels. D'un côté, les équipes de recherches s'intéressant au problème des résumés automatiques travaillent sur des programmes qu'ils ont développés et qui sont purement destinés à la recherche. De l'autre côté nous trouvons les systèmes développés pour usage et vente commerciale, pour entreprises ou particuliers. La plupart de ces logiciels utilisent une approche statistique basée sur la fréquence d'apparitions des mots dans les différentes phrases. Nous verrons dans les sections à venir les principales caractéristiques de quelques systèmes existants en débutant par les logiciels non commerciaux.

### 1.2.1 Caractéristiques de divers logiciels rencontrés

#### 1.2.1.1 Type de résumé : indicatif, informatif ou critique

Nous retrouvons principalement trois types de résumés de textes : il peut être soit indicatif, informatif ou critique (Morris et al., 1999). Lorsque nous parlons d'un résumé indicatif, nous faisons allusion à une brève indication du sujet du texte. Le résumé rapporte principalement l'idée générale du document original plutôt que les détails de celui-ci. Il permet entre autres de capter l'attention du lecteur, en lui faisant part des idées prédominantes du document. Dans un résumé informatif, nous retrouvons plutôt les informations importantes du texte. Ce type de résumé est utilisé dans le but de remplacer le texte original en rapportant toutes les idées et les conclusions. Finalement, le résumé critique donne un point de vue subjectif par rapport au texte lu. La nature des résumés traités dans le présent document entre sous la catégorie des condensés informatifs.

#### 1.2.1.2 Langues supportées

Certains logiciels, dont des exemples seront fournis aux sections 1.2.2 et 1.2.3, supportent un nombre impressionnant de langues. D'autres sont basés sur la structure et la forme d'une langue et sont plus difficilement applicables pour d'autres langues. Les tests effectués dans ce mémoire ont été réalisés pour la langue française, mais pourraient facilement être appliqués pour l'espagnol (Torres-Moreno et al., 2001).

#### 1.2.1.3 Résumé de documents multiples

Il existe deux catégories de logiciels de production de résumés. Le premier permet de condenser le contenu d'un document unique. Le second quant à lui, permet de rechercher parmi une multitude de documents et de rendre un résumé global en ressortant l'essentiel de tous les documents. La production de résumés à partir de documents multiples ajoutent à la complexité du problème et à son évaluation (Marcu et Gerber, 2001). Ce mémoire ne traitera donc que des résumés de documents uniques afin de mieux cerner l'expérimentation.

### 1.2.2 Logiciels non commerciaux

#### 1.2.2.1 SweSum

SweSum est développé à la « Royal Institute of Technology » (KTH) à Stockholm. Il est sans aucun doute le mieux documenté des logiciels de résumés non commerciaux. En effet, outre un rapport technique (Dalianis, 2000) expliquant les grandes lignes de son fonctionnement, plusieurs documents et articles traitant de diverses expériences sont basés sur ce logiciel (Hassel, 2000; Hassel, 2003; Hassel, 2004), en plus de ceux non cités dans ce mémoire. Disponible en ligne via un simple navigateur web, il comporte des options telles la résolution de pronoms (aussi traité dans le présent mémoire au chapitre 6), le type de document ou encore la langue désirée. Il fonctionne dans plus de sept langues différentes, dont quatre sont testées et stables (anglais, danois, norvégien et suédois) et les trois dernières (français, allemand et espagnol) sont toujours à l'état de prototype, mais peuvent tout de même être utilisées. Ce logiciel est conçu pour résumer des textes du type articles de journaux ou académiques, enregistrés dans un format HTML ou sans formattage.

Voici globalement comment fonctionne SweSum, selon le rapport technique de Dalianis (Dalianis, 2000). Un article de journal écrit sous forme de texte brut est traité avec ces métriques :

- Position des phrases : Les phrases situées en début de texte se voient attribuer une plus grande valeur que les phrases en fin de texte selon un facteur de $1/n$, où $n$ est le numéro de la ligne.
- Valeurs numériques : Les phrases contenant des valeurs numériques seront considérées comme plus importantes que les phrases n'en ayant pas, et par conséquent se verront recevoir un plus haut pointage.

Dans le cas d'un article de journal écrit en format HTML, une métrique additionnelle sera appliquée :

- Balises HTML : Les balises HTML peuvent en effet contenir des informations importantes sur le texte. Ici, les segments contenant des caractères gras recevront un plus haut pointage que ceux n'en contenant pas.

Par la suite, ces métriques sont normalisées et combinées à pondérations égales afin d'obtenir un poids final pour chacune des phrases. Celles avec le plus de poids sont retenues selon le taux de compression spécifié par l'utilisateur.

#### 1.2.2.2 LSA Summarizer

L'Université de Toronto a développé un logiciel de production automatique de résumés (Miller, 2003) sur la base d'un algorithme LSA (Latent Semantic Analysis). Bien que ce simulateur ne semble pas avoir été baptisé, on le nomme généralement LSA Summarizer dans la littérature. L'algorithme LSA consiste en une méthode statistique qui a pour but de trouver des relations de similarité entre différents passages d'un document, et ce sans utiliser de méthodes symboliques ou de dictionnaires (Landauer et al., 1998). Il s'agit d'un algorithme qui traite uniquement avec la matrice de fréquences des mots, qui sera introduite à la section 2.1.1. Cette

matrice contient chacun des mots du texte sur ses colonnes, tandis que ses lignes représentent chacune des phrases. Il est ainsi possible de connaître depuis celle-ci la fréquence de chacun des mots à l'intérieur des phrases du document. L'algorithme estime les cohésions à partir de cette matrice initiale. Les résultats obtenus avec ce logiciel démontrent cependant que l'emploi de la méthode LSA n'a su apporter d'améliorations significatives face aux logiciels qui n'en faisaient pas usage.

#### 1.2.2.3 Sumatra

SUMATRA est un logiciel qui se distingue des autres en plusieurs points. Il est indépendant du type de résumé, en plus de ne pas être basé sur des techniques statistiques comme la majorité des systèmes ; à l'opposé, SUMATRA utilise une approche sémantique (Lie, 1998). Il crée une structure sémantique contenant les relations entre les mots d'une phrase. Par exemple, la phrase « John donne un livre à Mary »associe les trois mots : « John », « livre » et « Mary » avec la relation « donne ». Mais ce qui le distingue encore davantage, c'est qu'il ne s'agit pas d'un système fonctionnant par extraction comme les autres. Il commence par construire une structure sémantique du texte, ensuite un certain poids est attribué aux différentes relations de la structure. Par exemple, pour les relations provenant de phrases en début ou en fin de paragraphe, le poids des relations sémantiques provenant de ces phrases reçoivent un bonus. Finalement, selon le poids donné aux relations, les moins importantes sont mises de côté. C'est par conséquent la structure sémantique du texte qui est résumée et non le document lui-même. Le résultat de SUMATRA n'est donc pas un texte suivi, mais une structure sémantique réduite du texte original. Lie rapporte que le logiciel est en mesure d'extraire près de 50% des informations pertinentes lorsque la compression du document est demandée à 25%.

#### 1.2.2.4 Summarist

SUMMARIST est l'appelation d'un module de production automatique de résumés faisant partie d'une plate-forme de plus grande envergure nommée MUST (Lin, 1999). Développé à l'Institut de recherche ISI (Information Sciences Institute), institut affilié à l'école d'ingénierie USC (University of Southern California), MUST constitue un grand système permettant la traduction multilingue, la recherche d'informations et finalement, la condensation de document. Il utilise entre autres les technologies de la compagnie SYSTRAN (Babelfish) et leur célèbre produit de traduction en ligne. Pour ce qui est du module nous intéressant ici, SUMMARIST, il utilise des techniques statistiques et de recherche d'informations pour produire des extraits ou des abrégés (section 0.1.1). Dans le but de trouver le sujet d'un document, il utilise les techniques suivantes : position des phrases (section 1.3.1.3), expressions indicatives (section 1.3.1.1), et un concept de signature pour associer les segments à un sujet bien précis (Hovy et Lin, 1999). Le fonctionnement de la signature prend en compte une connaissance *à priori* de plusieurs documents. Pour chaque mot associé à un sujet en particulier, une banque de termes qui y sont eux aussi reliés est disponible. Ces signatures exigent donc une certaine forme d'apprentissage préalable pour être en mesure de retrouver une quantité minimale de mots dans une phrase et de l'associer à un sujet. Cette banque de mots évolue automatiquement au fil du traitement de nouveaux textes. Plus un terme possède de co-occurence, plus sa valeur est élevée. On peut ainsi arriver à trouver le sujet d'une phrase en observant le poids de chacun des mots clés. C'est via cette banque de mots que les phrases sont reliées à l'idée qu'elles contiennent. Le résultat que produit SUMMARIST est un extrait du document original, le module permettant les abrégés étant encore rudimentaire.

#### 1.2.2.5 Columbia Newsblaster

COLUMBIA NEWSBLASTER est développé par l'Université Columbia à New York. Il s'agit d'un logiciel qui produit un résumé unique à partir de documents multiples. En effet, contrairement aux logiciels précédents, il récupère l'idée de plusieurs articles parlant d'un même sujet et produit un résumé en utilisant des phrases de plusieurs de ces articles. Les résultats sont plutôt impressionnants et peuvent être consultés quotidiennement avec les nouvelles du jour[1]. Cet engin est constitué du moteur de résumés nommé DEMS SUMMARIZER (Dissimilarity Engine for Multidocument Summarization) (Schiffman et al., 2002). Le principe fondamental de cet outil consiste à se servir des phrases situées en début de paragraphe, ce qui est fort compréhensible puisqu'il est utilisé uniquement dans les résumés d'articles de journaux (section 0.1.2). Cependant, une seconde technique fort intéressante est combinée à celle-ci. Une analyse grammaticale est effectuée dans le but de retrouver les paires sujet-verbe. De ces paires, on mesure une métrique appelée « verb specificity » qui calcule le nombre de fois que les sujets d'une phrase se retrouvent en présence des mêmes verbes. Plus le couple sujet-verbe est rare, plus il est réputé apporter une information unique et pertinente. En combinaison à ces deux astuces principales, on utilise aussi des métriques de taille de phrases (ni trop longue, ni trop courte), de repérage de pronoms pour discriminer les phrases, etc.

#### 1.2.2.6 Cortex

CORTEX est un logiciel récemment développé conjointement entre l'École Polytechique de Montréal et les Universités du Québec à Chicoutimi et à Montréal (Torres-Moreno et al., 2001). Il possède le grand avantage de ne pas être destiné à un seul

[1] www1.cs.columbia.edu/nlp/newsblaster

type de document et peut s'adapter à n'importe quel genre de texte. Les détails de CORTEX ne seront pas donnés ici, puisqu'une description détaillée sera effectuée au chapitre 2. C'est ce logiciel qui fut utilisé pour les recherches du présent mémoire.

### 1.2.3 Logiciels commerciaux

Voici maintenant quelques-uns des logiciels commerciaux de résumés automatiques. Malheureusement beaucoup moins de détails sont disponibles quant à leur fonctionnement, puisqu'il s'agit généralement de secrets commerciaux. Seules les informations accessibles seront mentionnées.

#### 1.2.3.1 Minds

MINDS, originalement connu sous le nom de HYPERGEN, est développé à l'université New Mexico State University (Minds, 1997). Ce logiciel, codé en Java, peut résumer des documents en espagnol ou en anglais, il s'utilise avec des pages Internet, plus précisément des documents HTML. Il utilise trois techniques pour arriver à ses fins. La première, il analyse la structure du document grâce entre autres aux balises html des documents qu'il traite. Deuxième technique, il analyse la fréquence des termes rencontrés. Finalement, la troisième méthode utilisée extrait les mots pertinents des titres, ou de connaissances *à priori* du domaine afin d'allouer une plus grande importance aux phrases dont ces termes se retrouve dans le corps du texte.

#### 1.2.3.2 Copernic Summarizer

COPERNIC SUMMARIZER (Copernic, 2004), a été développé par la compagnie du même nom, Copernic. La clientèle ciblée par cette entreprise est constituée davantage de particuliers que de compagnies, contrairement aux autres logiciels mentionnés précédemment. Le condenseur de textes peut s'adapter à des logiciels bien connus tels : Word, Acrobat, Explorer, Netscape, Outlook et Eudora.

#### 1.2.3.3 Datahammer

DATAHAMMER, développé par Glucose Development Corporation représente un logiciel quelquefois cité dans la littérature. Ce logiciel qui était produit pour les utilisateurs de Mac ne semble plus être disponible ni même maintenu par le fabriquant, puisque aucune mention de ce produit n'est désormais accessible sur le site de Glucose[2].

#### 1.2.3.4 Inxight Sumarizer

INXIGHT SUMARIZER (Inxight, 2002) est développé par la compagnie Insight Software Inc., une compagnie autonome créée au Xerox PARC (Palo Alto Research Center) en 1997. Le logiciel possède une très grande flexibilité et offre la possibilité pour les administrateurs système de les intégrer aux réseaux intranet, extranet ou à leur serveur Internet, afin de leur ajouter la capacité de résumer les documents qui y sont contenus. Par exemple, il est possible de demander un résumé dans un « pop-up » lorsque l'utilisateur tient sa souris au-dessus d'un lien sur une page web. Sa principale utilisation est justement liée aux applications client-serveur. Ce logiciel

[2] `http://www.glu.com`

supporte au-delà de 200 formats de documents y compris les documents HTML. Word, PowerPoint ou ASCII. Le résumé est obtenu en tenant compte des paramètres suivants : nombre de mots du thème principal, les noms propres, la position des phrases dans le document ainsi que la taille de celui-ci. Par la suite, chacune des phrases se voit attribuer une note. L'usager peut modifier ces poids immédiatement avant le lancement de l'exécution, il peut spécifier des epressions indicatives ou encore des mots qu'il sait moins pertinents au résumé. Ceci influence directement la note accordée aux différentes phrases. INXIGHT SUMARIZER fonctionne en 12 langues différentes, soit : anglais, français, allemand, néerlandais, danois, finlandais, italien, espagnol, portugais, suédois et 2 types de norvégien. De plus, la compagnie continue son développement pour supporter prochainement le japonais, le coréen et le chinois (simplifié et traditionnel). Le logiciel se trouve largement répandu dans l'industrie : CNNfn.com, Reuters, l'outil de recherche web Altavista de Compaq, Inktomi, Verity, Internet Financial Network et The Wall Street Journal représentent quelques-unes des entreprises utilisant INXIGHT SUMARIZER de Xerox.

#### 1.2.3.5 Pertinence Summarizer

PERTINENCE SUMMARIZER (Pertinence, 2004) est développé par la compagnie Pertinence Mining, il s'agit d'un logiciel qui peut être utilisé sur un seul poste ou encore sur un serveur, un peu à la manière de INXIGHT SUMARIZER vu précédemment. Il est développé entièrement en XML/Java et permet une intégration relativement facile grâce à ses API. Ce produit est offert pour les systèmes d'exploitation Windows (9x, 2000 et XP) et toutes les versions de Unix et Linux confondues, en autant qu'une machine virtuelle Java y soit installée. Il peut travailler lui aussi en plusieurs langues, en tout 14 différentes : français, anglais, allemand, arabe, chinois, coréen, espagnol, grec, italien, japonais, néerlandais, norvégien, portugais et russe.

D'autres langues devraient s'ajouter prochainement. PERTINENCE SUMMARIZER peut traiter 7 types de documents, correspondant aux extensions bien connues : txt, htm, pdf, rtf, doc, ppt et xls. Il offre la posssibilité, mais non obligatoire, de lui fournir des expressions indicatives ou, à l'inverse, des mots d'exclusion qui indiquent au logiciel que nous désirons ignorer les phrases contenant ces mots. Ceci permet, lors de la production d'un résumé, de cibler davantage le type de résumé voulu. Ce simulateur offre une exclusivité par rapport à la concurrence : l'utilisateur peut choisir parmi un choix de domaines très précis pour identifier le document. Ceci permet de préciser le domaine et d'obtenir un résumé de plus grande qualité : chimie, finance, droit, médecine, télécommunication ou texte de presse sont des catégories qui peuvent optionnellement être sélectionnées. Le résumeur permet aussi plusieurs variantes graphiques pour l'affichage des condensés, ce qui permet une personnalisation du logiciel.

#### 1.2.3.6 Corporum Summarizer

CORPORUM SUMMARIZER (Bremdal, 2000) est développé par la compagnie CognIT a.s. Le résumeur de CognIT a.s fait partie d'une suite de logiciels appelée CORPORUM servant principalement à la gestion et à la classification de documents sur ordinateur. Il peut produire des résumés de textes en trois langues différentes, soit l'anglais, l'allemand et le norvégien. Il demeure aussi en développement pour supporter ultérieurement d'autres langues dont le néerlandais, le suédois et le français. CORPORUM SUMMARIZER peut s'adapter à divers types de documents : articles de journaux, pages HTML provenant du WWW, articles techniques et papiers scientifiques, ou encore des documents plus longs.

#### 1.2.3.7 Word

WORD, ce logiciel de traitement de textes bien connu, produit et popularisé par Microsoft, constitue généralement une comparaison de base utilisée par tous les développeurs de condenseur de textes. En effet, il possède une fonction permettant de résumer un document, et ce, depuis sa version '97. L'option nommée « Auto-Summarize » intégrée au traitement de texte est largement répandue, mais très peu utilisée vu son efficacité douteuse. Sa grande disponibilité fait en sorte qu'il sert de base de référence dans la plupart des recherches.

## 1.3 Techniques de production automatique de résumés

Différentes techniques sont envisageables pour produire automatiquement un résumé. Deux lignes directrices distinctes sont d'actualité : l'approche statistique (numérique) et l'approche symbolique. La première se sert des mots à titre d'unité de mesure, elle note les phrases selon leur position ou selon les types de termes ou d'expressions qu'elle contient. Elle calcule ces différentes mesures afin de discriminer ou d'avantager certaines phrases. La seconde technique vérifie quant à elle la façon dont les phrases sont construites, elle analyse la nature et le rôle des mots. Elle décortique les segments et regarde dans certains cas la sémantique pour obtenir plus d'informations.

Les techniques les plus utilisées seront maintenant présentées avec une brève description de leur fonctionnement.

### 1.3.1 Recherche d'informations

La recherche d'informations ou « Information Retrieval » englobe diverses techniques fortement répandues. La majeure partie du temps, ces méthodes se retrouvent dans les logiciels de condensés de textes de tous les types. Quelques astuces classiques biens connues introduites à l'origine par Edmundson (Edmundson, 1969) seront maintenant présentées.

#### 1.3.1.1 Utilisation d'expressions indicatives

Lorsqu'un auteur écrit un document, il emploie fréquemment des mots qui lui permettent d'insister sur certaines parties de son texte. Plusieurs expressions indicatives du type : « en résumé », « en conclusion », « notons que », « insistons sur », « cet article parle de », etc., sont très souvent utilisés. Des dictionnaires contenant ces expressions peuvent être employés dans le but d'ajouter un certain poids aux phrases contenant ces mots. Les segments qui incluent ces expressions apparaissent vraisemblablement plus importants que ceux n'en contenant aucune.

#### 1.3.1.2 Le titre du texte

En règle générale dans un document, les quelques mots faisant partie d'un titre décrivent efficacement les lignes du texte qui suit, et ce de manière souvent très précise. L'idée consiste à travailler avec les termes du titre en les considérant comme des expressions importantes. Un peu à la manière des expressions indicatives (section 1.3.1.1), les expressions du titre peuvent apporter une information supplémentaire sur l'importance de la phrase. Un segment contenant un mot qui se retrouve aussi dans le titre est réputé être une partie plus pertinente qu'une autre n'en contenant

pas.

#### 1.3.1.3 Position des phrases

La position des phrases dans le texte représente souvent un très bon indice de la pertinence de la phrase en question. Par contre, cette réalité se remarque beaucoup plus pour certaines catégories de textes que d'autres. Nous retrouvons sur le marché plusieurs logiciels qui visent les résumés d'articles de journaux (voir section 1.2). Pourquoi ?...justement afin d'exploiter cette propriété dans ce type de documents, puisque les éléments pertinents se retrouvent généralement au niveau des premières phrases du document. Nous pouvons aussi penser à d'autres exemples : dans un roman, dans un article scientifique ou dans la plupart des textes, à la toute fin nous retrouvons la conclusion ou le dénouement principal. Dans cette situation, les phrases situées à la fin sont elles aussi, importantes. De plus, la structure d'un simple paragraphe peut elle aussi apporter une information précieuse. La première phrase introduit généralement l'idée qui y est argumentée ou décrite, et la dernière phrase conclut cette idée. Dans le but d'obtenir de meilleurs résumés, il est essentiel de tirer profit de ces positions.

### 1.3.2 Méthode statistique : Fréquence des mots

La méthode la plus utilisée sous de multiples variantes vient incontestablement de Luhn, grâce à son idée d'observer les mots les plus fréquents du texte (Luhn, 1958) : les méthodes statistiques sont basées principalement sur cette idée, nous avons vu plusieurs exemples de logiciels aux sections 1.2.2 et 1.2.3. Un premier filtrage est d'abord effectué dans le texte pour enlever tous les termes du type : articles, verbes de support ; ensuite, avec ceux restants, la matrice de fréquences

note pour chacun des mots le nombre de fois qu'il se retrouve dans les phrases prises une à une. Ainsi, chacune des lignes de la matrice représente une phrase et les colonnes un mot du texte. Par la suite, cette matrice sert de source à une multitude de méthodes numériques. Certaines de ces méthodes seront abordées à la section 2.1.2. Une autre approche à considérer avec la matrice de fréquences consiste à appliquer des méthodes d'apprentissage supervisées en combinaison aux méthodes numériques (Chuang et Yang, 2000). L'idée est d'utiliser plusieurs méthodes statistiques et de construire un vecteur constitué des résultats obtenus pour les différents segments de phrases, par chacunes de ces méthodes. Ensuite, en utilisant diverses techniques d'apprentissage, le logiciel « mémorise » les vecteurs ainsi que l'information concernant la pertinence de la phrase. Il sera ensuite en mesure de déterminer, pour un nouveau document suivant les vecteurs construits, s'ils sont susceptibles d'être pertinents ou non.

#### 1.3.2.1 Discussion sur l'approche statistique

L'utilisation de méthodes statistiques ne possède pas que des avantages. En effet, McCagar (McCagar, 2004) expose plusieurs problèmes reliés à l'utilisation de ces techniques. Parmi ces problèmes, nous en notons quelques-uns de taille :

- L'intervention humaine peut difficilement être évitée. En effet, à plusieurs instants du processus un appui humain peut être requis : pour le pré-traitement du document dans le but de le rendre « compréhensible » pour le logiciel, et pour le post-traitement, car l'utilisateur doit s'assurer que les opérations effectuées sont conformes, sinon les étapes à venir pourraient ne pas fonctionner. Si nous désirons un résumé cohérent, nous devons généralement reformuler certains syntagmes[3] afin que les phrases puissent s'enchaîner correctement.

[3]Groupe de mots formant une unité à l'intérieur de la phrase.

- Les références ambiguës s'évitent difficilement. Bien que certaines techniques comme la résolution des anaphores (section 1.5.2) puissent améliorer grandement cet aspect, il reste que s'il s'agit d'un processus automatisé, ce n'est pas un procédé parfait et s'il s'agit d'une intervention manuelle, l'importance de l'aide humaine est rehaussée.
- Certains mots peuvent prendre un sens contradictoire à celui désiré. Par exemple, si une phrase débute par l'expression « à l'opposé », et que cette phrase est retenue, elle n'aura plus de sens dans le résumé si la phrase précédente ne s'y retrouve pas, ce qui est fort probable.

La popularité des méthodes statistiques-neuronales est due à la rapidité avec laquelle un document peut être traité. À l'opposé, ce n'est pas le cas pour les méthodes symboliques qui demandent une analyse profonde du texte, ce qui représente une opération de longue haleine, malgré la puissance des ordinateurs actuels. De plus, les résultats obtenus avec les méthodes numériques sont généralement d'une qualité satisfaisante et la différence actuelle entre ceux-ci et ceux obtenus avec des méthodes plus « intelligentes » n'est pas suffisamment significative. C'est pourquoi les techniques mathématiques sont toujours d'actualité.

### 1.3.3 Méthode symbolique : Utilisation de sémantique ou linguistique

Tels que vu précédemment, certains systèmes utilisent une approche sémantique plutôt que la traditionnelle approche statistique (section 1.2.2.3). Pour ce faire, les logiciels se servent de marqueurs grammaticaux et discursifs en plus de certains autres indices linguistiques (basés sur la structure même de la phrase, du paragraphe) pour annoter le document. Par exemple, un article (Minel et al., 2001) présente la manière dont sont implémentées ces techniques : d'une part, les expressions indicatives (section 1.3.1.1) permettent d'établir les sections sur lesquelles l'auteur à lui-même explicitement insisté. Ensuite, des recherches sont effectuées

sur des « relations organisatrices de connaissances », telles des relations définitoires et causales, afin de cibler certaines connaissances potentielles à ces endroits. Ces techniques sont entourées d'une méthode d'exploration contextuelle afin de trouver d'autres indices linguistiques : morphèmes[4] grammaticaux[5], indicateurs discursifs de relations entre concepts, relations de causalité, etc. Cette technique est efficace pour l'analyse sémantique, mais comme toutes les méthodes du même type, elle s'avère relativement lente et le calibre des condensés obtenus suite à ces analyses ne sont pas d'une aussi grande qualité.

### 1.3.4 Méthode symbolique : Utilisation des ontologies

L'utilisation des ontologies[6] peut apporter une information supplémentaire au résumé, elle amène une structure et des liens entre les différents mots du texte. Des applications concrètes de l'utilisation des ontologies pour les résumés de textes ont déjà été expérimentées. Une étude comparative (Wu et Liu, 2003) entre les ontologies et certaines méthodes statistiques classiques ne permet cependant pas de conclure en l'efficacité de cette méthode sur la sélection des phrases.

---

[4]En liguistique, on définit généralement un morphème comme la plus petite unité de son porteuse de sens qu'il soit possible d'isoler dans un énoncé. *Source : wikipedia.org*

[5]Les morphèmes grammaticaux sont des morphèmes qui appartiennent à une classe fermée, tels que « tu », « à », « et », etc. *Source : wikipedia.org*

[6]En informatique, une ontologie est un ensemble structuré de concepts. Les concepts sont organisés dans un graphe dont le relations peuvent être : des relations sémantiques ; des relations de composition et d'héritage (au sens objet). La structuration des concepts dans une ontologie permet de définir des termes les uns par rapport aux autres, chaque terme étant la représentation textuelle d'un concept. *Source : wikipedia.org*

## 1.4 Méthodes d'évaluation de la qualité des résumés automatiques

Lorsque nous obtenons le résumé d'un logiciel, il devient primordial de pouvoir en évaluer sa qualité, sa pertinence. À la simple lecture, il paraît parfois évident de voir si les idées principales du texte original se retrouvent dans le texte résultant ou encore si certaines ont été oubliées, mais que faire pour détecter de légères améliorations ou détériorations. Comment pouvons-nous comparer la performance de divers logiciels entre eux ? Il s'avère impératif de trouver une méthode plus objective que la simple lecture suivie de l'appréciation du résultat. Malheureusement, encore aujourd'hui aucune méthode ne s'avère parfaite et plusieurs articles, mémoires et thèses considèrent uniquement cet aspect (Saggion et Lapalme, 2000) (Mani, 2001b) (Hassel, 2004). Nous pouvons regrouper les évaluations sous deux catégories, la section suivante les présentera.

### 1.4.1 Intrinsèque ou Extrinsèque

Deux types d'évaluation existent et peuvent être utilisés : intrinsèques ou extrinsèques (Mani et Maybury, 1999). Dans le premier cas, il s'agit d'une personne qui évalue directement le résumé selon certains critères établis. Dans le deuxième cas, la qualité d'un résumé est basée sur la manière dont il interfère sur une autre tâche. Par exemple, si le résumé est utilisé sur un logiciel de questions-réponses, la qualité du résumé sera jugée selon les réponses fournies par le système. En ne modifiant que le résumé, si l'acuité des solutions obtenues augmente, le résumé sera réputé être de meilleure qualité (Torres-Moreno et al., 2004). Pour la méthode intrinsèque, il existe une méthode simple et souvent utilisée. Nous demandons à certaines personnes de retenir les phrases qu'elles jugent les plus pertinentes et nous évaluons l'intersection entre celles-ci et celles retenues par le logiciel (Luhn, 1958). Cette technique

simple s'avère malheureusement loin d'être idéale puisqu'il existe une grande différence entre les choix effectués par les logiciels et ceux faits par les humains (Rath et al., 1999). Nous retrouvons aussi un logiciel, SEE (Summary Evaluation Environment) (Lin, 2001), permettant d'évaluer un résumé en comparaison à un second modèle idéal. Ce logiciel permet d'évaluer à la fois les abrégés et les extraits (voir section 0.1.1). Chacune des phrases (ou toutes autres divisions utilisées) est présentée à l'utilisateur, ce dernier doit entrer le niveau de couverture (cohérence) du sujet par la phrase présentée en rapport au modèle fourni en plus d'associer les idées entre les documents. Une fois l'ensemble des phrases évaluées, une appréciation globale du résumé est aussi demandée à l'usager. Le logiciel fourni finalement les résultats établis à partir de l'intersection entre les éléments du modèle et ceux du document évalué.

### 1.4.2 Contenu ou Cohérence

Nous pouvons orienter l'évaluation afin de représenter deux aspects d'un résumé : son contenu ou sa cohérence. Les cas extrêmes suivants démontreront cette différence. À titre d'exemple, le résumé d'un texte titré avec l'expression suivante : « Les Chiens ». L'obtention d'un résumé parfaitement cohérent, mais dont le contenu serait complètement absent pourrait se produire. En effet, si le condensé obtenu porte sur les animaux en général avec des phrases bien structurées, le résumé s'avérerait cohérent, mais le contenu défaillant. À l'opposé, si notre extrait traite des chiens et de leurs caractéristiques, et qu'il rapporte toutes les informations pertinentes du texte, mais que celles-ci apparaissent dans un désordre total, un résumé avec un contenu idéal et une cohérence nulle en serait le résultat.

## 1.5 Post-traitement des résumés : traitement des anaphores

Les anaphores sont un « Procédé qui consiste à reprendre un élément du discours antérieur par un élément grammatical qui y renvoie. », selon l'Office de la langue française du Québec (`http://www.granddictionnaire.com`). La résolution des anaphores ne constitue pas une technique de production de résumés en elle-même, mais elle peut s'avérer utile sur certains aspects pour l'amélioration des systèmes existants. Pour le moment, certains résultats obtenus venant de travaux antérieurs seront revus. Ils portent sur un cas particulier du traitement des anaphores : le traitement des pronoms. Le contenu d'un article à ce sujet est repris à la section 1.5.2. Le présent mémoire traite également de ce même sujet au chapitre 6.

### 1.5.1 Expansion du résumé pour pallier le problème des anaphores

Outre la substitution des anaphores dont nous venons de discuter, il existe un autre procédé afin de permettre de conserver l'idée derrière les phrases contenant les anaphores. Ce procédé consiste à insérer des phrases additionnelles au résumé, celles contenant les mots qui permettent de préserver le contexte. Cependant, si nous sommes stricts sur la quantité de phrases à inclure dans le résumé, elles devront être introduites au détriment d'autres phrases. Nous reparlerons aussi de cette méthode au chapitre 6.

### 1.5.2 Résolution des anaphores : Pronominal Resolution in Automatic Summarisation (Hassel, 2000)

Cette thèse de Hassel démontre la possibilité de traiter les pronoms d'un texte afin d'en améliorer la cohérence et de préserver ses informations importantes. Elle est

basée sur le logiciel SweSum (introduit à la section 1.2.2.1) et l'idée est implémentée sous forme de préprocesseur, soit l'extension PRM (Pronoun Resolution Module) de SweSum. Les textes traités sont écrits en suédois et proviennent de journaux, sous format HTML.

Le document explique principalement la difficulté qui réside dans le traitement des anaphores, la méthode utilisée pour les résoudre et les algorithmes implémentés afin d'y arriver. L'extension réalisée au moment de la rédaction de la thèse ne résolvait que les pronoms suédois « han, honom, hans » et « hon, henne, hennes », que Hassel associe respectivement aux pronoms anglais « he, him, his » et « she, her, her ».

Afin d'évaluer les résultats, dix textes ont été utilisés en deux versions : une première où les anaphores ont été traitées automatiquement par le préprocesseur PRM, et la seconde représente la version originale avec les anaphores. Neuf étudiants ont de plus participé à l'expérience. La méthode utilisée consiste à demander à tous les étudiants de résumer les textes à l'aide de SweSum. Par la suite, ils indiquent subjectivement à partir de quel taux de compression la cohérence du texte est brisée, et à partir de quel autre pourcentage certaines informations essentielles sont perdues. Les résultats démontrent que la résolution des pronoms n'a presque rien changé si on regarde les informations importantes du texte. Pour ce qui est de la cohérence, elle n'est améliorée qu'en présence de textes possédant une grande quantité de pronoms.

Le chapitre 6 du présent mémoire reprend certains tests similaires à ceux effectués pour SweSum. Nous pouvons par conséquent anticiper des résultats similaires.

# CHAPITRE 2

# CORTEX

Toutes les techniques que nous avons expérimentées en vue de l'amélioration de la qualité des résumés ont été appliquées à CORTEX (Torres-Moreno et al., 2001), que nous avons déjà introduit à la section 1.2.2.6. Ici, nous décrirons ses caractéristiques un peu plus en détail afin de permettre la compréhension du contexte dans lequel les recherches ont été effectuées.

## 2.1 Fonctionnement

CORTEX se base principalement sur une représentation vectorielle des textes (Salton et McGill, 1986) pour extraire les phrases les plus pertinentes. Cette technique, dont les idées principales seront développées au cours de ce chapitre, a été adaptée à la production automatique de résumés à l'intérieur de CORTEX (Torres-Moreno et al., 2001). L'algorithme utilisé se sert de méthodes statistiques, qui permettent une analyse très rapide pour de longs textes comparativement aux méthodes basées sur une analyse syntaxique ou sémantique. Les condensés ainsi obtenus sont de type informatif et pourraient, tel qu'expliqué à la section 1.2.1.1, permettre au lecteur de connaître en quelques lignes le contenu du document original.

Pour arriver à produire le condensé, CORTEX construit d'abord une matrice de fréquences de mots dans laquelle chacune des lignes représente une phrase et chacune des colonnes représente un mot du texte. Ainsi, nous pouvons connaître la fréquence d'apparition de chacun des mots dans chacune des phrases. Seuls les

mots revenant au minimum deux fois dans le texte entier sont retenus dans cette matrice (Torres-Moreno et al., 2001). Une fois la matrice construite, un ensemble de métriques est appliqué à chaque phrase du texte. Ces métriques sont ensuite combinées selon une méthode que nous aborderons plus loin, afin de fournir une valeur entre 0 et 1. Cette valeur représente une évaluation de la pertinence de la phrase. Onze métriques ont été définies, ce qui nous donne 2047 combinaisons possibles (nous ne considèrons pas la combinaison « aucune des métriques »). Nous verrons en détail à la section 2.1.2 les dix premières de ces métriques qui étaient déjà présentes dans CORTEX. La section 5.1 traitera de la onzième, que nous avons ajoutée au système.

### 2.1.1 Pré-traitement

Le pré-traitement est une sorte de pipeline dans lequel le texte est soumis. À chacune des étapes, un traitement différent est appliqué au texte. Ces opérations sont constituées de plusieurs types de filtrage, de la segmentation et de la lemmatisation[1]. La figure 2.1, inspirée du plus récent rapport technique de CORTEX (St-Onge, 2003), représente la séquence d'opérations appliquées au document en guise de pré-traitement.

[1]Lemmatiser signifie que nous ne conservons que la forme non fléchie du mot : infinitif pour les verbes, singulier pour les mots pluriels, etc.

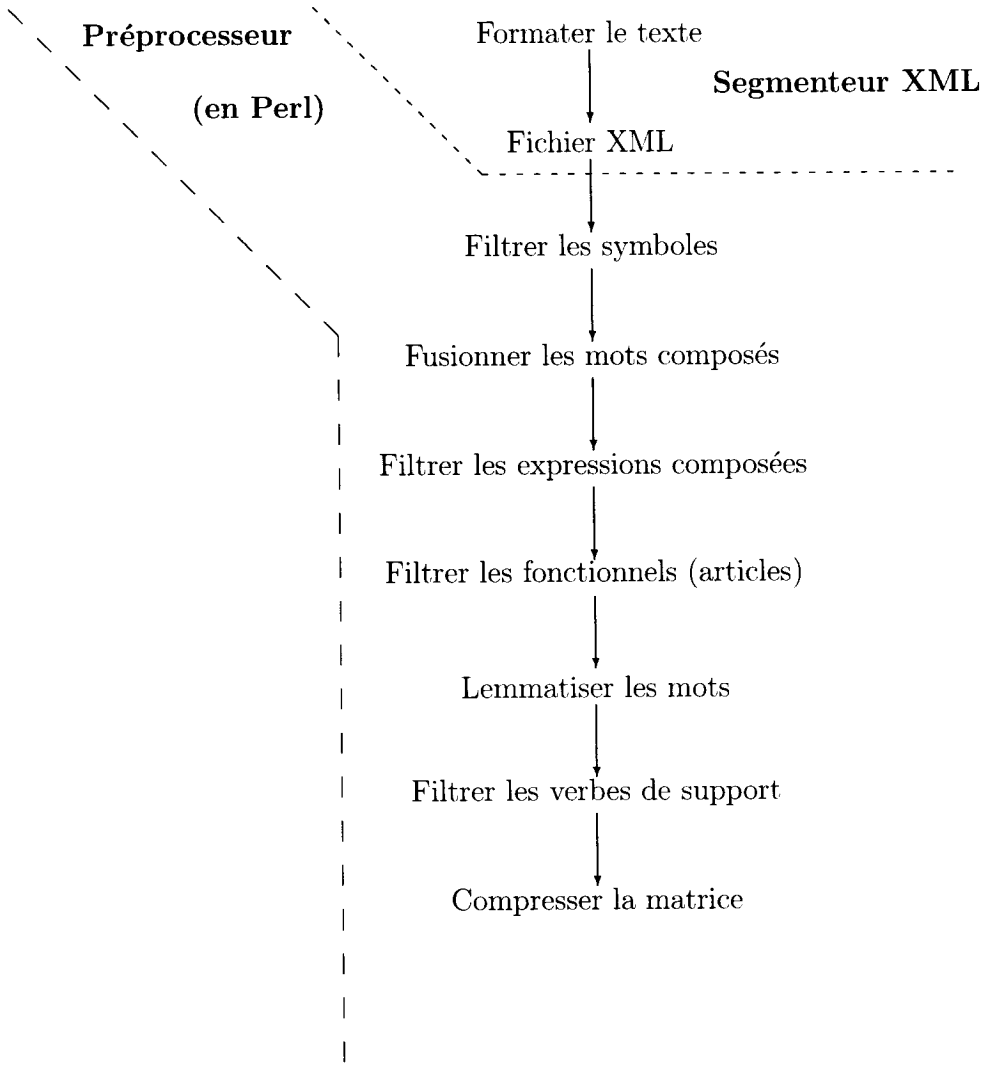


Figure 2.1 Pipeline du pré-traitement de Cortex

La segmentation du texte en XML est réalisée dans une première étape via un module externe qu'est le segmenteur. Ce dernier s'occupe de découper le texte en phrases et de les baliser pour permettre aux modules suivants de repérer adéquatement les segments. Il serait aussi possible pour de plus longs textes, d'utiliser un découpage plus grossier (par exemple en paragraphes ou pages) afin de réaliser les analyses.

Les expériences ayant été réalisées sur des textes bruts, les titres et sous-titres ont dû être segmentés manuellement. Par la suite, le fichier XML est utilisé comme entrée de base à un module de pré-traitement écrit en Perl. Comme le démontre la figure 2.1, suite à la segmentation, les symboles sont retirés du document (ponctuation, caractères spéciaux, chiffres, ...) et nous fusionnons les mots composés. Cette fusion est effectuée à l'aide d'un dictionnaire contenant des mots composés connus ; lorsqu'une expression est rencontrée, nous relions les mots par le caractère « _ » (« abat jour », « chauffe eau », etc.), il s'agit là de mots qui doivent s'écrire avec un trait d'union, cependant les symboles étant filtrés au départ, nous nous devons de les réinsérer. Ensuite, nous filtrons les expressions moins pertinentes telles « à_jamais » ou « dans_ce_cas »et les articles tels « le », « la », etc, et nous lemmatisons. Finalement nous éliminons les verbes de support (avoir, être, pouvoir, etc.) qui ne feraient qu'ajouter du bruit. Nous verrons au chapitre 8 que les étapes de filtrage et de fusion expliquées ci-haut peuvent être éliminées grâce à l'utilisation d'une analyse grammaticale.

Les termes restants sont représentés sous forme d'une matrice appelée la matrice fréquentielle. Telle que décrite en début de chapitre, cette matrice sert de base à différentes métriques. Plus précisément, dix des onze métriques utilisées se basent sur des équations mathématiques. Tout d'abord, deux variables nécessaires pour poser les équations sont présentées ci-dessous :

$N_P$ : Nombre de phrases au total.

$N_L$ : Nombre de mots distincts ayant au moins une double présence dans le texte.

Maintenant, la matrice fréquentielle : chaque élément représente le nombre de fois que le mot $i$ est présent dans la phrase $\mu$.

$$\gamma = \begin{bmatrix} \gamma_1^1 & \gamma_2^1 & \gamma_3^1 & \cdots & \gamma_i^1 & \cdots & \gamma_{N_L}^1 \\ \gamma_1^2 & \gamma_2^2 & \gamma_3^2 & \cdots & \gamma_i^2 & \cdots & \gamma_{N_L}^2 \\ \gamma_1^3 & \gamma_2^3 & \gamma_3^3 & \cdots & \gamma_i^3 & \cdots & \gamma_{N_L}^3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma_1^\mu & \gamma_2^\mu & \gamma_3^\mu & \cdots & \gamma_i^\mu & \cdots & \gamma_{N_L}^\mu \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma_1^{N_P} & \gamma_2^{N_P} & \gamma_3^{N_P} & \cdots & \gamma_i^{N_P} & \cdots & \gamma_{N_L}^{N_P} \end{bmatrix}, \quad \gamma_i^\mu \in \{0, 1, 2, \ldots\} \tag{2.1}$$

Le processus de pré-traitement se termine une fois cette matrice obtenue. Afin de finaliser le tout, la matrice est compressée dans le but de réduire le temps consacré au passage de l'information du module en Perl, vers le cœur de CORTEX, en C++. Le procédé utilisé pour la compression est une technique simple. En résumé, les informations relatives à la matrice sont notées dans un fichier texte à l'aide de caractères et de nombres. Elle s'inspire d'une technique classique de compression, RLE (Run Length Encoding), technique qui s'avère très efficace puisque la matrice contient une quantité importante de 0.

### 2.1.2 Les métriques

À la suite du pré-traitement, les valeurs des métriques sont calculées pour chaque phrase du texte. Le point de départ de chaque métrique est la matrice de fréquences (éq. 2.1). Puisque le pré-traitement compressait cette dernière, elle devra par conséquent être décompressée pour débuter le processus. Par la suite, nous supposons l'existence d'une matrice identique à la matrice de fréquences $\gamma$, à la différence que chaque mot possède la valeur 1 ou 0, selon qu'il se retrouve ou non dans la phrase,

respectivement. Formellement, la matrice se définit ainsi :

$$\xi = \begin{bmatrix} \xi_1^1 & \xi_2^1 & \xi_3^1 & \dots & \xi_i^1 & \dots & \xi_{N_L}^1 \\ \xi_1^2 & \xi_2^2 & \xi_3^2 & \dots & \xi_i^2 & \dots & \xi_{N_L}^2 \\ \xi_1^3 & \xi_2^3 & \xi_3^3 & \dots & \xi_i^3 & \dots & \xi_{N_L}^3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \xi_1^\mu & \xi_2^\mu & \xi_3^\mu & \dots & \xi_i^\mu & \dots & \xi_{N_L}^\mu \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \xi_1^{N_P} & \xi_2^{N_P} & \xi_3^{N_P} & \dots & \xi_i^{N_P} & \dots & \xi_{N_L}^{N_P} \end{bmatrix} \tag{2.2}$$

Avec

$$\xi_i^\mu = \left\{ \begin{array}{ll} 1 & \text{si } \gamma_i^\mu \neq 0 \\ 0 & \text{si } \gamma_i^\mu = 0 \end{array} \right\} \tag{2.3}$$

Avec ces définitions en main, il est maintenant possible de poser les métriques suivantes, qui permettront d'assigner un poids pour évaluer l'importance des différentes phrases :

- **F :** Cette métrique nommée « Fréquence des mots » alloue une plus grande valeur aux phrases contenant plusieurs mots faisant partie de la matrice. Cela a pour effet d'avantager les phrases longues. Plus une phrase contient de mots appartenant à la matrice, plus sa note est élevée, et cette éventualité est quasi directement reliée à la longueur de la phrase. À la section 3.2.1, il sera démontré que cette métrique employée seule servira comme base de comparaison pour les autres combinaisons. Elle est basée sur l'équation ci-dessous :

$$F^\mu = \sum_{i=0}^{N_L - 1} \gamma_i^\mu. \tag{2.4}$$

Si nous regardons la formule d'un autre sens, la note de la phrase est calculée en

comptant le nombre total de mots qu'elle contient en y soustrayant le nombre de mots fonctionnels, le nombre de verbes de support et le nombre de mots ne se retrouvant nulle part ailleurs dans le texte ; les mots composés ne compte que pour un.

- **I :** Cette métrique nommée « Interaction de segments » a pour but de trouver des relations entre les phrases.

$$I^{\mu} = \sum_{\substack{i=0 \\ \xi_i^{\mu} \neq 0}}^{N_L-1} \sum_{\substack{j=0 \\ j \neq \mu}}^{N_P-1} \xi_i^j. \tag{2.5}$$

Deux phrases sont dites en relation si elles possèdent des mots communs. La force du lien peut se définir comme étant le nombre de mots communs aux deux phrases. Le poids final d'une phrase est établi en effectuant la somme de la force des relations qu'elle possède avec toutes les autres phrases.

- **D :** Cette métrique nommée « Somme fréquentielle des probabilités » est basée sur la probabilité de retrouver un mot dans le texte. La métrique suit l'équation suivante :

$$D^{\mu} = \sum_{i=0}^{N_L-1} p_i \gamma_i^{\mu}, \tag{2.6}$$

avec

$$p_i = \frac{1}{T} \sum_{\mu=0}^{N_P-1} \gamma_i^{\mu}, \tag{2.7}$$

et

$$T = \sum_{\mu=0}^{N_P-1} \sum_{i=0}^{N_L-1} \gamma_i^{\mu} = \sum_{\mu=0}^{N_P-1} F^{\mu}. \tag{2.8}$$

En bref, si un mot a une probabilité d'apparition élevée et qu'il se retrouve dans la phrase $\mu$, celle-ci sera avantagée puisqu'elle possède un mot important.

- **E :** Cette métrique nommée « Entropie » est également basée sur la probabilité

de retrouver un mot dans une phrase, elle est suit l'équation ci-dessous :

$$E^{\mu} = -\sum_{\substack{i=0 \\ \xi_i^{\mu} \neq 0}}^{N_L - 1} p_i \log_2 p_i, \tag{2.9}$$

avec $p_i$ calculé à l'équation 2.7. Nous pouvons conceptualiser cette métrique comme étant la quantité d'informations contenues dans une phrase, tel que décrit dans (Mani, 2001a).

- **Y :** Cette métrique nommée « Distances d'Hamming » est basée sur la matrice du même nom. La matrice de Hamming $H$ se constitue de chacun des mots du document à la fois sur ses lignes et ses colonnes, chaque case représentant ainsi une paire de mots. Voici la métrique :

$$Y^{\mu} = \sum_{\substack{m=1 \\ \xi_m^{\mu} \neq 0}}^{N_L - 1} \sum_{\substack{n=0 \\ \xi_n^{\mu} \neq 0}}^{m-1} H_n^m \tag{2.10}$$

et la matrice

$$H_n^m = \sum_{j=0}^{N_P - 1} \left\{ \begin{array}{ll} 1 & \text{si } \xi_m^j \neq \xi_n^j \\ 0 & \text{autrement} \end{array} \right\} \quad \text{pour} \quad \begin{array}{l} m \in [1, N_L - 1] \\ n \in [0, m-1] \end{array}. \tag{2.11}$$

Nous pouvons remarquer que la matrice d'Hamming se trouve à être triangulaire inférieure, ce qui se comprend aisément, puisque pour contenir chacune des paires de mots, cette seule partie suffit. La valeur de la case se voit attribuer un nombre élevé lorsque les deux mots du couple se retrouvent rarement utilisés dans la même phrase. La métrique considère donc importants les termes utilisés seuls, considérant que les paires de mots trop fréquents peuvent très souvent s'avérer être des synonymes.

- **P :** Cette métrique nommée « Poids d'Hamming des segments » s'avère être

presque identique à la métrique **F** (éq 2.4), à l'exception près qu'elle somme les mots de la phrase qu'une seule fois si ceux-ci se répètent.

$$P^{\mu} = \sum_{i=0}^{N_L - 1} \xi_i^{\mu}. \tag{2.12}$$

- **T :** Cette métrique nommée « Somme des poids d'Hamming des mots par segments » regarde pour chacun des mots d'une phrase donnée, si les autres phrases du texte utilisent ce mot. Elle est basée sur l'équation ci-dessous :

$$T^{\mu} = \sum_{\substack{i=0 \\ \xi_i^{\mu} \neq 0}}^{N_L - 1} \psi_i, \tag{2.13}$$

avec

$$\psi_i = \sum_{\mu=0}^{N_P - 1} \xi_i^{\mu}. \tag{2.14}$$

Plus précisément, la note donnée à la phrase sera la somme, pour chacun des mots, de toutes les occurrences de ce même terme dans les autres segments du texte.

- **L :** Cette métrique nommée « Poids d'Hamming lourd » tente de retenir les phrases ayant les mots les plus diversifiés du texte. Elle est basée sur l'équation ci-dessous :

$$L^{\mu} = \phi^{\mu} T^{\mu}, \tag{2.15}$$

avec

$$\phi^{\mu} = \sum_{i=0}^{N_L - 1} \xi_i^{\mu} \tag{2.16}$$

et $T^{\mu}$ est le même que calculé précédemment à l'équation 2.13. Concrètement, cette métrique s'avère être exactement la même que **T**, mais nous y multiplions aussi le nombre de mots différents dans la phrase.

- **O** : Cette métrique nommée « Somme des poids d'Hamming de mots par fréquence » agit encore une fois un peu à la manière de la métrique **T**. Voici l'équation la modélisant :

$$O^{\mu} = \sum_{i=0}^{N_L - 1} \psi_i \gamma_i^{\mu}. \tag{2.17}$$

  La différence avec l'équation 2.13 se trouve dans le fait que nous y multiplions la fréquence du mot considéré.

- **A** : Cette métrique nommée « Angle entre un titre et la phrase » avantage les phrases contenant des mots du titres. Elle est basée sur l'équation ci-dessous :

$$A^{\mu} = \frac{\sum_{i=0}^{N_L - 1} \gamma_i^{\mu} \gamma_i^{T_{\mu}}}{\|\gamma^{\mu}\| \, \|\gamma^{T_{\mu}}\|}. \tag{2.18}$$

  Ici, nous ne faisons que comparer chacun des mots de la phrase avec le titre et le(s) sous-titre(s) au(x)quel(s) la phrase appartient. Afin de combiner les comparaisons, la métrique effectue un produit scalaire normalisé comme si la phrase et le titre étaient des vecteurs à $N_L$ dimensions. Plus une phrase s'approche du titre, plus l'angle est faible et plus le produit scalaire normalisé tendra vers 1.

- **X** : Cette métrique n'était pas incluse à l'origine dans Cortex, elle sera introduite au chapitre 5. Contrairement aux dix autres métriques, elle n'utilise pas la matrice $\gamma$ mais se base plutôt sur la position des phrases dans le texte. Elle favorise les phrases aux extrémités des sections des documents.

### 2.1.3 Algorithme de décision

Dans le but de combiner les métriques entre elles et de pondérer les phrases, Torres-Moreno propose un algorithme particulier (Torres-Moreno et al., 2001) qui, contrairement à une simple moyenne, permet de scinder efficacement en deux parties la distribution des résultats. Voici en quoi consiste cet algorithme pour une phrase $\mu$,

tel que décrit dans (St-Onge, 2003) et (Torres-Moreno et al., 2004) :

$$\Sigma_{\alpha}^{\mu} = \sum_{\substack{\nu=0 \\ \|\lambda_{\mu}^{\nu}\|>0.5}}^{\Gamma-1} (\|\lambda_{\mu}^{\nu}\| - 0.5) \tag{2.19}$$

$$\Sigma_{\beta}^{\mu} = \sum_{\substack{\nu=0 \\ \|\lambda_{\mu}^{\nu}\|<0.5}}^{\Gamma-1} (0.5 - \|\lambda_{\mu}^{\nu}\|) \tag{2.20}$$

Avec

$\nu \equiv$ indice de la métrique

$\Sigma \equiv$ somme des différences absolues entre $\|\lambda\|$ et 0.5

$\alpha \equiv$ métriques normalisées dites positives

$\beta \equiv$ métriques normalisées dites négatives

$\Gamma \equiv$ nombre de métriques considérées

Une fois ces deux sommations réalisées pour la phrase, la décision sera prise selon l'algorithme suivant :

SI $\Sigma_{\alpha}^{\mu} > \Sigma_{\beta}^{\mu}$ ALORS

$$\Lambda^{\mu} = 0.5 + \frac{\Sigma_{\alpha}^{\mu}}{\Gamma}$$

SINON

$$\Lambda^{\mu} = 0.5 - \frac{\Sigma_{\beta}^{\mu}}{\Gamma}$$

Où nous obtiendrons :

$\Lambda \equiv$ valeur finale de poids

Nous réalisons ainsi une sorte de moyenne des métriques supérieures à 0.5 distincte de la moyenne des métriques qui y sont inférieures. Il ne s'agit pas là d'une moyenne à proprement parler, puisque la division est effectuée avec le nombre total de métriques et non uniquement avec le nombre de métriques considérées pour la

sommation. C'est le plus grand de ces deux résultats qui sera utilisé pour calculer la valeur finale de poids. La figure 2.2 fait une synthèse complète du fonctionnement de CORTEX.

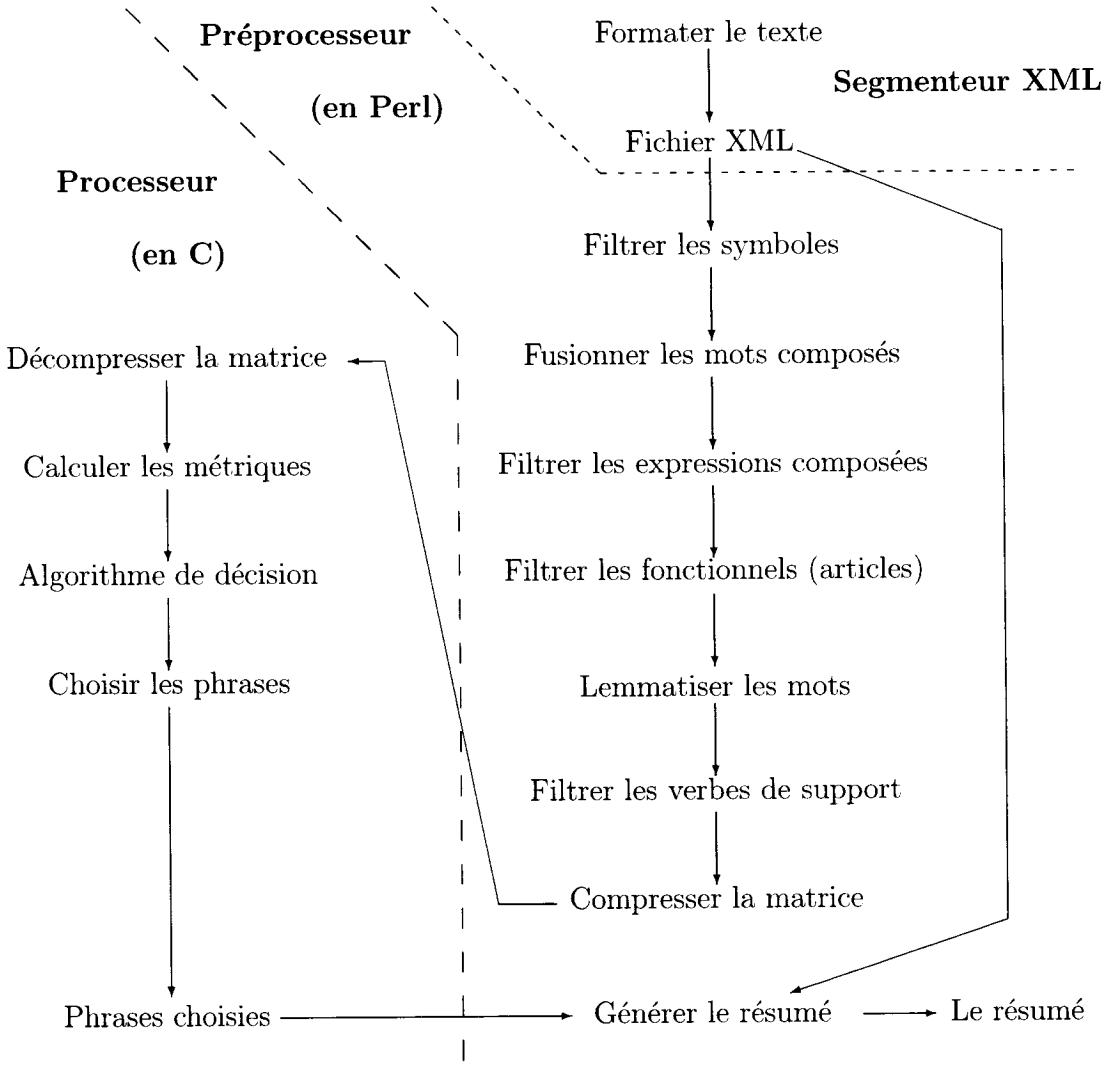


Figure 2.2 Pipeline du traitement complet de Cortex

Regardons maintenant à l'aide de la figure 2.3, un exemple réel provenant du texte *Cybermédias* (phrase #43, annexe I.1).

| | | |
|---|---|---|
| Phrase originale | : | `Les annonceurs aiment bien être rassurés, ne pas se lancer à l'aveuglette dans un projet publicitaire, même si les chiffres qu'on leur fournit semblent crédibles.` |
| Filtrage symboles | : | `les annonceurs aiment bien être rassurés ne pas se lancer à l aveuglette dans un projet publicitaire même si les chiffres qu on leur fournit semblent crédibles` |
| Fusion mots composés | : | `les annonceurs aiment bien_être rassurés ne pas se lancer à l aveuglette dans un projet publicitaire même si les chiffres qu on leur fournit semblent crédibles`[a] |
| Filtrage articles | : | `annonceurs aiment bien_être rassurés lancer aveuglette projet publicitaire chiffres fournit semblent crédibles` |
| Lemmatisation | : | `annonceurs aimer bien_être rassurer lancer aveuglette projet publicitaire chiffrer fournir sembler crédible` |
| Filtrage v.support | : | `annonceurs aimer bien_être rassurer lancer aveuglette projet publicitaire chiffrer fournir sembler crédible`[b] |

Figure 2.3 Exemple de pré-traitement d'une phrase par CORTEX.

[a]Nous remarquons ici un des désavantages d'utiliser la reconnaissance d'expressions via un dictionnaire, CORTEX reconnaît ici bien être comme le nom, ce qui ne devrait pas être le cas. Nous verrons au chapitre 8 comment palier à ce problème.

[b]Ici, nous filtrons les verbes de support (avoir, être, etc.). Cependant, cette phrase n'en contient déjà aucun.

Après avoir franchi ces étapes de pré-traitement, les mots restants sont ajoutés à la matrice de fréquences à condition que ceux-ci soient représentés à plus d'une reprise dans le texte entier. Des valeurs numériques sont ainsi associées aux phrases suivant les métriques décrites précédemment à la section 2.1.2.

## 2.2 Évaluation de Cortex

Des études de CORTEX (Torres-Moreno et al., 2001; Torres-Moreno et al., 2002) montrent comment il se compare face à d'autres logiciels connus. Ces études l'ont comparé avec MINDS, COPERNIC SUMMARIZER et WORD (déjà introduits aux sections 1.2.3.1, 1.2.3.2 et 1.2.3.7 respectivement). Les conclusions de ces études démontrent que CORTEX se classe très bien parmi ses concurrents. Des résultats comparables à ceux de MINDS, comparables voir même supérieurs à COPERNIC SUMMARIZER y sont rapportés. Finalement, les résultats produits par WORD sont hors de compétition alors qu'il s'est trouvé incapable de retenir la moindre phrase d'importance dans les tests effectués.

Nous verrons à la section 3.2 les méthodes utilisées pour évaluer les travaux du présent mémoire. Les résultats qui y seront présentés compareront CORTEX avec lui-même. Si nous arrivons à l'améliorer encore davantage, notre but sera ainsi atteint. En effet, nous venons de voir que des études antérieures démontrent que CORTEX se classe bien face à la concurrence. Depuis les trois dernières années, un seul des trois logiciels utilisés dans ces comparaisons a évolué : COPERNIC SUMMARIZER. Pour sa part, MINDS n'a subit aucune modification et le projet semble désormais avoir été abandonné. MICROSOFT n'ont pas amélioré eux non plus leur module de résumés.

Le tableau 2.1 montre ainsi une comparaison actuelle de CORTEX et COPERNIC SUMMARIZER.

Tableau 2.1 Comparaison entre CORTEX et COPERNIC SUMMARIZER version 2.1

| | COPERNIC (%) | | | | CORTEX ETX (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 40 | 50 | 15 | **35** | 55 | 50 | 18 | 41 |
| Épicier | 40 | 47 | 27 | **38** | 60 | 50 | 40 | 50 |
| Kanthume | 41 | 32 | 32 | **35** | 46 | 58 | 25 | 43 |
| Football | 33 | 0 | 22 | **18** | 73 | 36 | 55 | 55 |
| J'Accuse | 41 | 24 | 18 | **28** | 67 | 33 | 29 | 43 |
| Univers | 31 | 23 | 31 | **28** | 43 | 50 | 50 | 48 |
| Opus Dei | 34 | 34 | 37 | **35** | 36 | 40 | 44 | 40 |
| Cybermédias | 40 | 50 | 40 | **40** | 14 | 57 | 14 | 28 |
| Sciences | 35 | 35 | 40 | **37** | 39 | 52 | 52 | 47 |
| Sirène | 28 | 34 | 18 | **27** | 63 | 40 | 51 | 51 |
| Travail | 17 | 26 | 35 | **26** | 36 | 16 | 28 | 27 |
| Moyenne | | | | **32** | | | | 43 |

La technique utilisée pour établir ces résultats sera détaillée au chapitre 3. Nous pouvons tout de même survoler le tableau rapidement et s'apercevoir que CORTEX retourne de meilleurs résultats sur presque la totalité des expérimentations. La colonne de gauche contient le nom de onze textes traités par les deux logiciels, alors que les lettres A, B et C correspondent à trois juges indépendants.

# CHAPITRE 3

# MÉTHODOLOGIE

Afin d'atteindre les objectifs visés, plusieurs choix ont dû être effectués. Nous verrons au cours des prochaines pages quels ont été ces choix ainsi que ce qui les a motivés. Les textes ayant servi aux évaluations, la façon dont ont été réalisées ces mêmes évaluations, et finalement les techniques envisagées dans le cadre de cette maîtrise sont tous des sujets qui seront abordés dans les sections à venir.

## 3.1 Corpus de textes

Le corpus que nous avons utilisé est constitué de onze textes de sujets variés, allant du conte jusqu'au rapport scientifique, en passant par la monographie. Certains textes sont plus narratifs, d'autres comportent des dialogues ou encore sont davantage à caractère informatif. Plusieurs variétés de textes ont été retenus afin de pouvoir constater l'influence des divers traitements sur les différents styles d'écritures.

Voici les textes utilisés, classés par ordre alphabétique :

1. La Croissance Des Cybermédias et la Publicité sur Internet[1]
   Ce texte constitue le plus petit du corpus, de style informatif et il possède de courtes phrases.

[1] `http://www.quebecscience.qc.ca/Cyber/1.0/1_29_72.asp`

2. Qu'est-ce qu'une fausse science[2]

   Ce texte informatif est constitué autant de courtes phrases que de très longues.

3. Les Futurs contradictoires du Travail[3]

   Cette monographie comporte plusieurs phrases majoritairement de longueur moyenne et longue. Il contient très peu d'anaphores malgré sa longueur. Il est aussi constitué de plusieurs titres de sections et de sous-sections.

4. J'Accuse[4]

   Ce texte écrit à la première personne s'adresse directement à un lecteur précis, il s'agit d'un document argumentatif. Il est constitué de phrases généralement courtes.

5. Kanthume : Un Projet d'Analyse Analogique Suivant Un Modèle Cognitif d'Induction[5]

   Il s'agit d'un mémoire de maîtrise française. Ce document scientifique comporte plusieurs titres de sections et de sous-sections.

6. Notes sur le foot-ball[6]

   Ce texte de style informatif est formé de phrases généralement de longueur moyenne.

7. L'Opus Dei[7]

   Ce texte, à mi-chemin entre un texte informatif et un conte fantaisiste narratif, contient plusieurs titres de sections et de sous-sections. Les phrases y s'avèrent généralement assez courtes et nous

[2] http://www.ac-nice.fr/philo/outils/d-faussesc.htm
[3] http://perso.wanadoo.fr/bernard.perret/texte4.htm
[4] http://abu.cnam.fr/cgi-bin/donner_unformated?jaccuse3
[5] http://mediatheque.ircam.fr/articles/textes/Lartillot02f
[6] http://abu.cnam.fr/cgi-bin/donner_unformated?football1
[7] http://tentacules.net/toc/misc/article.php?id=263

y retrouvons plusieurs noms propres.

8. La Petite Sirène[8]

   Ce conte, plutôt long pour un texte de ce style, comporte beaucoup de pronoms personnels se référant à la même personne, soit « petite sirène », deux mots faisant partie du titre principal.

9. Rapport sur les Affaires de l'Amérique du Nord Britannique[9]

   Ce texte mieux connu sous le nom *Rapport Durham*, de style narratif, rapporte des faits et des événements. Il se compose de phrases de longueur moyenne ou courte.

10. Satan l'Épicier[10]

    Ce conte pour enfant comporte plusieurs dialogues. Il est constitué de plusieurs phrases très courtes.

11. La Vie Dans l'Univers[11]

    Ce texte de style informatif se compose principalement de phrases de longueur moyenne.

Le tableau qui suit montre chacun de ces textes détaillés selon leurs caractéristiques.

---

[8] `http://www.hattemer.fr/Noel_contes/Conte_Andersen_Petite_Sirene.htm`
[9] `http://northernblue.ca/canchan/cantext/colonial/1838durf.php`
[10] `http://histoire-en-ligne.com/article.php3?id_article=544`
[11] `http://lirenligne.free.fr/livre.php?livre_id=29`

Tableau 3.1 Caractéristiques des différents textes du corpus

| | Nombre total de mots | Nombre de mots différents[a] | Nombre de phrases |
|---|---|---|---|
| Durham | 6515 | 1038 | 210 |
| Épicier | 3438 | 706 | 191 |
| Kanthume | 5506 | 865 | 230 |
| Football | 2761 | 669 | 102 |
| J'Accuse | 4912 | 840 | 207 |
| Univers | 4373 | 782 | 139 |
| Opus Dei | 8529 | 1524 | 443 |
| Cybermédias | 1276 | 306 | 62 |
| Sciences | 5627 | 937 | 225 |
| Sirène | 8358 | 973 | 346 |
| Travail | 8264 | 1296 | 244 |

[a]Ce nombre exclut les mots filtrés par le pré-traitement tel que vu à la section 2.1.1

## 3.2 Méthode d'évaluation

Nous avons cité à la section 1.4 quelques techniques existantes pour évaluer la qualité d'un résumé. Dans le contexte des présents travaux, CORTEX est utilisé seul, il est donc entendu qu'une méthode intrinsèque devra être employée. Dans un premier temps, il serait possible (mais sans doute peu efficace) de lire le résumé obtenu et de juger arbitrairement de sa qualité. Cette technique ne s'avère pas très précise et laisse place à beaucoup d'ambiguïté. En effet, elle repose sur une base totalement arbitraire, les textes s'en retrouvent possiblement classifiés de manière biaisée. La

seconde technique envisageable consiste à faire ressortir manuellement, avant de lancer le résumé automatique, les phrases qui nous semblent les plus pertinentes. Ensuite, une fois les phrases sélectionnées et après avoir lancé CORTEX, nous comparons les sélections avec les résultats obtenus via CORTEX. Plus le nombre de phrases en commun est élevé, plus la valeur du résumé est rehaussée.

Les évaluateurs devaient sélectionner parmi toutes les phrases d'un document, celles qu'ils jugeaient les plus pertinentes. Chacun d'eux devait retenir environ 15 à 20% des phrases du document original dans sa sélection. Lors de l'utilisation de CORTEX, ce dernier sélectionnait seulement 10% des phrases du document, puisqu'au delà de ce seuil, il nous semblait qu'il ne s'agissait plus vraiment d'un résumé. Évidemment, puisque le nombre de phrases retenues par les évaluateurs humains est plus élevé, les résultats obtenus le seront aussi. Cependant, la section 3.2.1 tient compte de ceci ; en effet plus les évaluateurs humains sélectionnent de phrases, plus la base de comparaison est élevée. Le résultat produit par CORTEX est le pourcentage des phrases retenues que nous retrouvons également dans la sélection manuelle.

Évidemment, cette technique n'est pas parfaite et elle laisse encore place à une certaine subjectivité. Par contre, il est important de constater que le résumé « parfait » n'existe pas et que si nous demandions la même tâche à deux personnes différentes, il est peu probable que nous obtenions des choix de phrases identiques. Quoi qu'il en soit, les évaluations avant/après se font en relation avec une seule et même évaluation manuelle ; les deux résumés sont ainsi comparés avec un « idéal » commun. De plus, cet exercice est fait plusieurs fois pour un seul texte ; en effet, plus d'une personne ont été solicitées afin de résumer chacun des textes manuellement. Chacun des onze textes possède donc trois résumés à la main provenant de personnes différentes.

### 3.2.1 Mesures de base

Afin de se donner un seuil minimum à franchir, certaines mesures simples peuvent servir de bases de comparaison. Les résultats sont donc comparés à trois mesures élémentaires :

1. Phrases les plus longues (Approximée avec la métrique F seule)

   Nous avons vu à la section 2.1.2 que la métrique F n'avantageait pas les phrases les plus longues directement, mais bien celles possédant le plus de mots dans la matrice de fréquences : les mots, autres que fonctionnels, revenant plus d'une fois dans tout le document. Cependant, si nous approximons le fait que chacune des phrases possède une proportion équivalente de mots utiles (ceux qui seront recopiés dans la matrice) en fonction de sa longueur réelle, les phrases les plus longues seront aussi celles où nous retrouverons le plus de mots utiles. Il s'agit donc d'une méthode quasi équivalente à la sélection des phrases à partir desquelles nous retrouvons le plus de mots utiles, soit la métrique F prise seule.

2. Les premières et dernières phrases de sections (Métrique X seule)

   Il est courant, dans un document, de retrouver les phrases les plus importantes en introduction ou en conclusion d'une section du texte. Une méthode simple de sélection de phrases serait de ne retenir que les premières et les dernières phrases, sans se soucier du contenu. Il s'agit là de l'effet produit par la métrique X.

3. Choix de phrases aléatoires

   C'est sans contredit la technique, si elle en est une, la plus simple. Elle s'avère pertinente lorsque nous désirons nous assurer de l'efficacité d'une méthode. Si celle-ci fournie des résultats équivalents

à ceux obtenus en sélectionnant des phrases aléatoirement, nous pouvons la rejeter.

Le résultat obtenu à l'aide d'une sélection aléatoire dépend du nombre de phrases tirées. En effet, si un plus grand nombre de phrases est choisi, les probabilités d'un résultat élevé sont augmentées. Rappelons que le résultat obtenu est le pourcentage des phrases retenues par CORTEX que nous retrouvons également dans la sélection manuelle (ou aléatoire dans ce cas). C'est la raison pour laquelle nous remarquons dans le tableau 3.2 une valeur différente pour chacun des textes. Le résultat indiqué est déterminé en tirant aléatoirement un nombre égal de phrases à celui provenant des évaluations manuelles. Ainsi, pour un nombre total $N$ de phrases dans un texte, si $p$ phrases sont retenues aléatoirement, l'espérance du pourcentage d'intersection entre les $p$ phrases choisies et le nombre $c$ de phrases retenues par CORTEX sera de $\frac{p}{N}\%$, et ce, peu importe la valeur de $c$, tant qu'elle est supérieure à 0. En effet, si nous considérons $p$ phrases idéales dans le bassin des $N$ phrases et si nous sélectionnons une quantité $c$ de phrases parmi celles-ci, la proportion de phrases idéales sera la même que dans le bassin total des phrases.

Si nous ne réussissons pas à obtenir de meilleurs résultats qu'avec l'une de ces trois techniques, il sera possible de conclure que la méthode utilisée est inefficace, puisqu'un calcul très simple fournirait un meilleur résultat. Le tableau 3.2 montre les résultats obtenus à l'aide de ces trois techniques.

Tableau 3.2 Tableau comparatif de trois techniques de base.

| | Phrases longues | Premières et dernières phrases | Hasard |
|---|---|---|---|
| Durham | 44 % | 27 % | 20 % |
| Épicier | 50 % | 32 % | 23 % |
| Kanthume | 39 % | 32 % | 20 % |
| Football | 42 % | 27 % | 25 % |
| J'Accuse | 33 % | 11 % | 13 % |
| Univers | 43 % | 28 % | 18 % |
| Opus Dei | 44 % | 35 % | 22 % |
| Cybermédias | 43 % | 33 % | 26 % |
| Sciences | 39 % | 29 % | 22 % |
| Sirène | 40 % | 30 % | 23 % |
| Travail | 21 % | 21 % | 18 % |
| Moyenne | 40 % | 28 % | 21 % |

Nous remarquons que le choix des phrases les plus longues s'avère être la mesure la plus efficace parmi les trois techniques simples présentées. Nous pourrons ainsi la sélectionner comme base de comparaison, en faisant abstraction du choix aléatoire des phrases et de la sélection des premières et dernières phrases de section, puisque nous savons ces techniques moins efficaces.

## 3.3 Méthode d'évaluation des techniques expérimentées

Dans le but de vérifier l'efficacité des techniques étudiées, des résultats sont recueillis avant et après chaque changement apporté au logiciel. Pour en évaluer l'impact, nous regardons si les notes attribuées par CORTEX sont, la majeure par-

tie du temps, à la hausse ou à la baisse, suite à la modification. Par exemple, pour un texte donné, si une majorité des notes provenant des différents évaluateurs sont à la hausse, nous considérerons une amélioration pour ce document. Ensuite, si la grande majorité des textes étudiés se voit améliorée, il sera possible de conclure en l'efficacité de la méthode étudiée. Évidemment, la déduction inverse s'avère tout aussi valide. Avec cette méthode, nous évaluerons l'ajout d'une nouvelle métrique, la sélection des meilleures combinaisons de métriques à utiliser, l'impact de la substitution des pronoms d'un texte, la possibilité de choisir dynamiquement les métriques à employer et finalement l'utilisation d'une analyse syntaxique du document.

# CHAPITRE 4

# RÉINGÉNIERIE DE CORTEX

Au départ, le code original de CORTEX était codé en Perl et en C. Le module de traitement en C avait déjà été, au cours des années, la cible de plusieurs tests et expérimentations, ce qui rendait la qualité du code très discutable. En effet, il était courant de rencontrer des fonctions sans paramètre ni valeur de retour, qui travaillaient uniquement avec des variables et des tableaux globaux. Le code était conséquemment très difficile à lire et c'est la raison pour laquelle nous nous devions de trouver une solution afin de faciliter l'intégration de nouveaux modules et de nouveau code, ainsi que la maintenance de celui existant. La solution envisagée fut celle de la réécriture complète du module en C de CORTEX en un moteur beaucoup plus modulaire, en C++. Pour réaliser les études décrites dans ce mémoire, il fallut modifier substantiellement le code source utilisé. Ce chapitre résume les modifications globales effectuées sur CORTEX. Faisant abstraction des détails d'implémentation, les modifications majeures seront mentionnées. Notons cependant que la revue complète du code a de plus permis la détection et la correction de quelques bogues dans le code, rendant celui-ci encore plus stable. La figure 4.1 illustre le fonctionnement de CORTEX avant et après les modifications faites pour les tests de ce mémoire. Les sections à venir reprennent chacun des éléments ajoutés, apparaissant en rouge sur la figure, et survolent ces modifications.

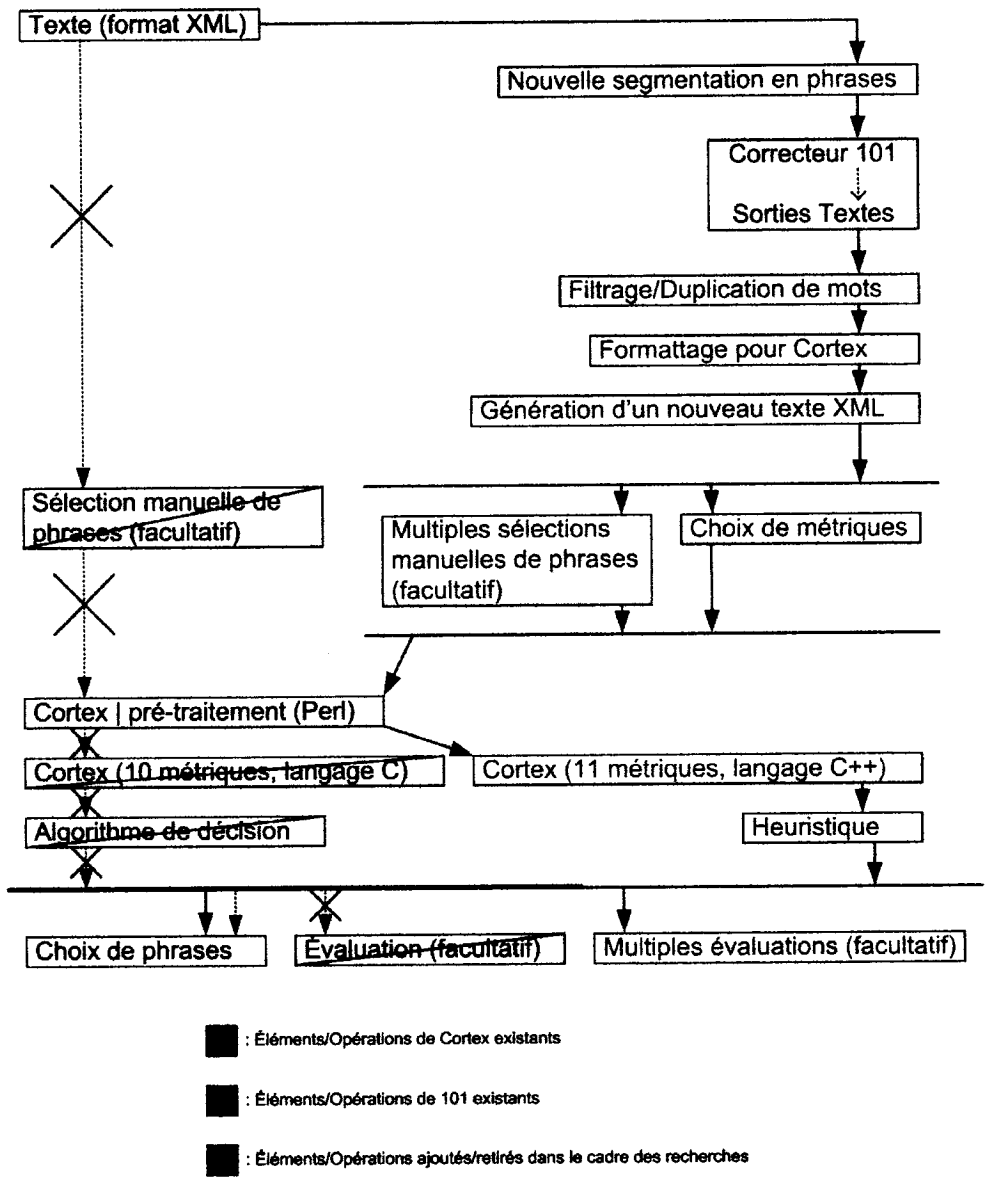


Figure 4.1 Figure illustrant les éléments déjà existants de Cortex (en noir) et du Correcteur 101 (en bleu), ainsi que les éléments/opérations ajoutés aux logiciels afin de réaliser les tests du présent mémoire (en rouge).

## 4.1 Migration du code

La section 2.1.2 explique le fonctionnement des dix métriques de CORTEX. Ces métriques contenues dans CORTEX étaient codées sous forme de plusieurs fonctions plus ou moins spécifiques. En effet, la plupart de ces fonctions modifiaient une importante quantité de variables globales. L'idée de base de la migration vers un code en C++ était, entre autres, de faire de l'intégration ou de la modification de nouvelles métriques une opération simple. De plus, il était primordial d'éliminer les éléments globaux et d'instancier des entités concrètes et distinctes sous forme d'objet pour des éléments tels les métriques ou la matrice de fréquences (abordés aux sections 2.1.2 et 2.1). Le diagramme UML de classes de CORTEX est illustré à la figure 4.2.

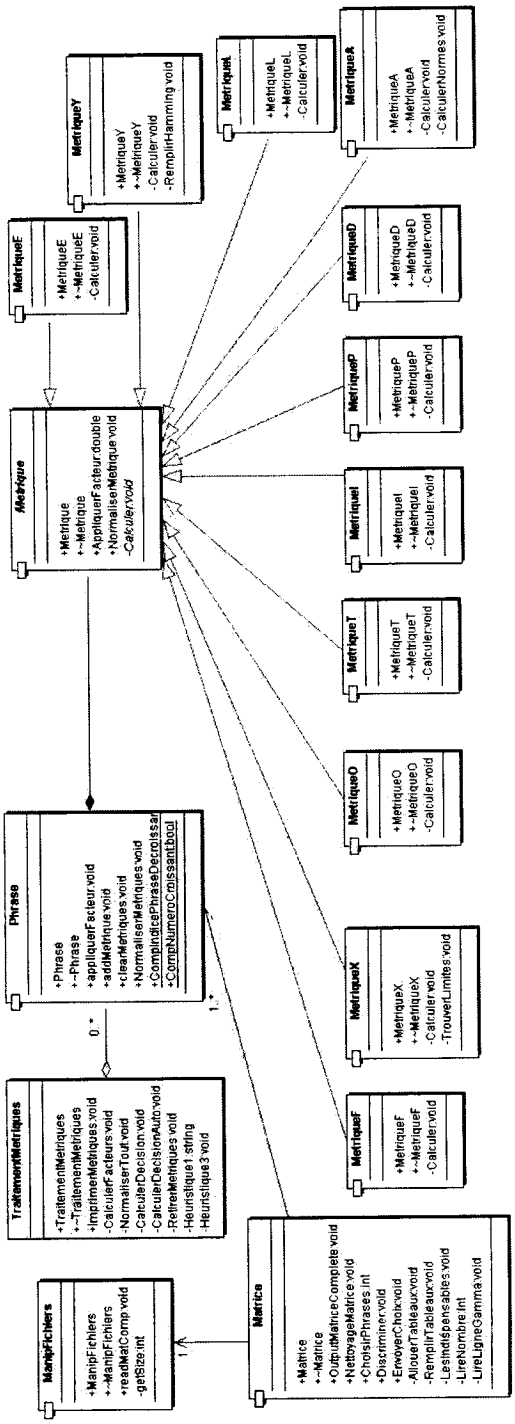


Figure 4.2 Diagramme UML de classes pour CORTEX.

Un des principaux avantages du code tel qu'il est devenu, consiste en sa capacité à redéfinir aisément de nouvelles métriques. Grâce au polymorphisme, le processus ne nécessite maintenant que la dérivation de la classe de base abstraite « Metrique ». Toutes les méthodes nécessaires à l'utilisation de la métrique se retrouvent dans cette classe, il ne reste qu'à en dériver une classe enfant et de redéfinir la fonction virtuelle pure « Calculer » pour implémenter l'algorithme rattaché à la métrique. Cette méthode est appelée automatiquement lors de la création de l'objet. Ensuite, seules deux méthodes publiques sont accessibles : AppliquerFacteur et NormaliserMetrique, qui permettent respectivement d'appliquer un facteur multiplicatif à la valeur de la métrique (non utilisé pour les tests de ce mémoire), et une méthode permettant de normaliser la valeur de la métrique en fournissant les valeurs limites possibles.

Les métriques sont une composition, au sens UML, de la classe « Phrase ». Elles n'existent qu'à l'intérieur de celle-ci. Les phrases contiennent donc toutes les métriques à appliquer, ainsi que les méthodes publiques suivantes : appliquerFacteur, jouant le même rôle que dans la classe « Metrique », mais cette fois-ci agissant sur la valeur de la phrase ; addMetrique, permettant l'ajout d'une nouvelle métrique à la phrase ; clearMetrique, pour éliminer toutes les métriques ; NormaliserMetriques, pour lancer la normalisation de toutes les métriques contenues dans la phrase ; et finalement deux fonctions statiques permettant de comparer les indices et les numéros entre deux phrases.

La matrice, contenue dans la classe du même nom, contient la matrice de fréquences décrite à la section 2.1. C'est cette matrice qui possède toutes les phrases du document. Les méthodes publiques disponibles avec la classe « Matrice » sont : OutputMatriceComplete, permettant d'afficher le contenu de la matrice ; NettoyageMatrice, méthode qui élimine les phrases vides (ne contenant aucun mot pertinent) ; ChoisirPhrases, parcourant les phrases disponibles et sélectionnant les plus perti-

nentes ; Discriminer, fonction utilisée pour discriminer certaines phrases inférieures à un seuil (non utilisée pour les tests de ce mémoire) ; et finalement, EnvoyerChoix, permettant d'écrire le numéro des phrases retenues dans un fichier.

Une classe additionnelle, « TraitementMetriques », permet de gérer l'ensemble des métriques, de les ajouter aux phrases et de lancer l'algorithme de décision. Elle contient la liste des phrases que l'on retrouve dans la matrice et accède aux métriques via celles-ci.

En résumé, la matrice contient toutes les phrases du texte et chacune des phrases contient un nombre quelconque de métriques, dépendamment de celles utilisées lors de l'exécution. Une métrique est contenue dans une seule phrase à la fois, puisque la valeur de la métrique est définie indépendamment pour chacune des phrases (revoir la section 2.1.2 pour plus de détails). Un aspect intéressant de cette architecture, qu'il n'était pas possible d'utiliser précédemment, vient du fait que chaque métrique est un objet indépendant à l'intérieur d'une phrase. Il est donc possible d'utiliser des combinaisons de métriques différentes pour chacune des phrases, alors qu'avec l'ancienne conception elles étaient toutes nécessairement affectées par les mêmes métriques. Nous allons d'ailleurs exploiter cette caractéristique au chapitre 7, lorsque nous tenterons de trouver automatiquement les métriques à utiliser.

Une fois la réécriture du module en C de CORTEX vers un code en C++ terminée, il devient évident que la compréhension des différentes fonctions, dorénavant sous forme de classes et de méthodes, est rehaussée. Avec un temps d'exécution du même ordre[1] et, obligatoirement, des résultats identiques, le nouveau moteur de CORTEX est beaucoup plus malléable. Ainsi, plusieurs modifications, qui seront décrites dans les sections à venir, ont par la suite pu être effectuées afin de permettre

[1] Aucun test n'a concrètement été réalisé au sens de la vitesse d'exécution ; cependant, la différence, s'il y en a une, n'est pas humainement perceptible. Puisque l'utilisation de CORTEX est directement effectuée par l'humain, la différence de temps est négligeable.

une utilisation de CORTEX beaucoup plus intuitive.

### 4.1.1 Modification des entrées : choix des métriques

La modification des entrées est plus précisément reliée à la manière dont les métriques sont utilisées. En effet, tel qu'il sera expliqué à la section 5.2.1, dans la version originale de CORTEX, toutes les métriques étaient simultanément utilisées. Par conséquent, nous devions effectuer les modifications nécessaires au code existant afin de permettre un choix ciblé de métriques à l'exécution. Ainsi, CORTEX a été modifié dans le but de permettre aisément de spécifier, lors du lancement du logiciel, les métriques désirées pour cette exécution. La série de métriques peut maintenant être choisie et donnée à la ligne de commandes, au lancement de l'exécutable.

### 4.1.2 Modification des sorties : évaluations multiples

Cette modification est en relation avec les résultats produits par CORTEX. Nous avons vu au chapitre 3.2 que la méthode d'évaluation de CORTEX est basée sur un système de pondération. Pour arriver à ce résultat, nous avons dû changer considérablement le système d'attribution des notes pour en arriver à obtenir des évaluations que nous jugions adéquates. La méthode repose essentiellement sur la comparaison entre le résultat produit automatiquement et quelques résultats obtenus manuellement. Initialement, la sortie de CORTEX permettait d'obtenir en plus de la sélection de phrases, une comparaison de celles-ci avec un choix manuel de phrases. Cependant, un seul choix pouvait être comparé à la fois, et CORTEX devait être recompilé si nous voulions modifier l'élément de comparaison. Par conséquent, il fallut modifier le code pour permettre aisément le changement de la sélection ma-

nuelle en plus de permettre l'affichage de plusieurs résultats simultanément, pour plusieurs évaluateurs différents. Malgré ces ajouts, le temps d'exécution n'en est pas plus élevé puisque le calcul final du résultat n'est qu'une simple comparaison entre les phrases retenues par CORTEX et celles retenues par les évaluateurs. Les évaluations manuelles doivent être insérées dans un fichier texte sous la forme suivante :

Nom_Évaluateur1
Nombre_de_phrases_retenues
numéro
numéro
...
numéro
##
Nom_Évaluateur2
Nombre_de_phrases_retenues
numéro
...

où « numéro » représente le numéro associé à la phrase retenue par l'évaluateur.

La version actuelle de CORTEX permet ainsi de recueillir plusieurs numéros de phrases pour plusieurs évaluateurs différents, dans un seul fichier texte. Nous évitons de cette manière toute recompilation du code suite à une modification du choix manuel des phrases.

## 4.2 L'analyseur grammatical

### 4.2.1 Nouvelle segmentation en phrases et génération d'un nouveau texte XML

Comme nous le verrons plus loin, il a fallu, dans une de nos expérimentations, interfacer CORTEX avec un analyseur grammatical. Un problème important est dû au fait que cet analyseur segmente les phrases de manière différente à la division produite par CORTEX. Nous avons dû créer un script Perl qui permet de forcer la segmentation des phrases en se basant sur la division déjà établie dans le fichier XML initial. Pour ce faire, le script injecte une à une les phrases (plutôt que le texte en entier) découpées par CORTEX et s'assure de fusionner la sortie de l'analyseur si ce dernier redivise la phrase de nouveau. Une fois l'analyse complétée, la fusion de la phrase s'effectue, si nécessaire, en même temps que la réinsertion de nouvelles balises XML, permettant ainsi de conserver la numérotation. Tous les autres éléments XML qui ne sont pas des phrases (titres principaux, titres de section, etc.) sont systématiquement recopiés dans la version finale, sans passer par l'analyseur.

### 4.2.2 Formatage pour Cortex

Le module d'analyse syntaxique dont nous nous servons est le moteur d'un logiciel commercial, « Le Correcteur 101 $^{TM}$ », et nous n'avons pas la possibilité d'en modifier le code source. Conséquemment, les sorties qu'il produit doivent nécessairement être prises telles quelles. Elles sont obtenues sous forme d'un texte formaté selon un standard maison, ce qui nous oblige à convertir les différentes sorties produites afin de ne conserver que les informations nécessaires, tout en structurant celles-

ci pour qu'elles puissent être comprises par CORTEX. Parmi ces sorties, on note principalement la phrase originale ainsi que plusieurs arbres de dépendances où chacun d'eux contient tous les mots et signes de ponctuation de la phrase, en plus des relations qui relient chacun de ces éléments entre eux. Chacun des arbres ainsi produits constitue une analyse possible de la phrase. En effet, puisque l'analyse effectuée par 101 n'est pas toujours parfaite, les différentes possibilités d'analyses sont fournies, en plus d'un poids sous forme de valeur numérique associé à chacune d'elles, accordant un certain niveau de probabilité à l'analyse.

### 4.2.3 Filtrage/Bonification de mots

Nous devons maintenant utiliser les sorties produites par 101 et effectuer les modifications nécessaires aux programmes afin d'en tirer avantage. Pour ce faire, nous avons utilisé les relations contenues dans l'arbre de dépendances ayant le poids le plus important, donc la probabilité d'exactitude la plus élevée. Puisque nous voulons bonifier et/ou éliminer certaines catégories de mots ou de relations, dépendamment de la technique utilisée (détaillée au chapitre 8), nous devons considérer chacune des relations possibles et déterminer si nous voulons filtrer les mots ou en accentuer l'importance, en multipliant la fréquence du mot dans la phrase. Le module développé pour effectuer ce travail, écrit en langage C++, reconstruit l'arbre de dépendances produit en sortie texte par 101 dans une structure de données. Cette structure permet ainsi de traiter chacun des mots dépendamment les uns des autres. Une base de données, sous forme de fichier texte, contient toutes les catégories et toutes les relations de mots possibles, en plus d'un nombre représentant l'importance de cette catégorie ou relation. Ainsi, chacun des nœuds de l'arbre est exploré pour vérifier la relation qui s'y retrouve. Lorsque la relation du nœud courant doit être supprimée (le nombre associé à la relation est 0), la structure en arbre permet du même coup d'éliminer tout le sous-arbre s'y rattachant. De même,

lorsqu'un mot appartient à une catégorie de mots moins importante, il est simplement retiré. À l'opposé, si nous désirons bonifier certaines relations, par exemple les sujets du verbe, le module produira en sortie un doublon (ou une fréquence plus élevée, dépendamment du nombre que l'on retrouve dans la base de données) pour ce mot, occasionnant ainsi un doublon du mot (ou plus) dans la phrase.

L'une des difficultés majeures rencontrées lors de l'élaboration de ces techniques est la présence de conjonctions de coordination (et, ou) dans les phrases. Regardons l'exemple suivant, figure 4.3 :

```
L'article paraîtra dans la
presse nationale et dans le
journal local.

Arbre paraîtra/verbe
 Circ et/et
  Coord dans/prep
   CmplDePrep presse/nom
    Déter la/deter
    Épithète nationale/adjectif
  Coord dans/prep
   CmplDePrep journal/nom
    Déter le/deter
    Épithète local/adjectif
 Sujet article/nom
  Déter L'/deter
```

Figure 4.3 Exemple de problème introduit par la coordination.

Le problème se situe au niveau du filtrage. Dans cet exemple, le filtre sera appliqué sur le mot « et » alors qu'il devrait plutôt être appliqué sur les éléments coordonnés « dans la presse nationale » et « dans le journal local » ; la conséquence sera la suppression de la relation circonstantielle et d'une partie importante de la phrase. Si nous ne nous préoccupons pas de ce type de situation, le filtrage sera trop important

et nous risquons de laisser tomber d'importantes informations. Pour cette raison, nous avons dû modifier le modèle afin de lui intégrer un anti-filtre permettant de considérer ce type de situation. L'anti-filtre vérifie la nature du mot faisant l'objet de l'analyse, et si nécessaire, reporte le filtrage aux éléments coordonnés.

Regardons maintenant la figure 4.4, qui fait une synthèse du code utilisé pour le filtrage, sous forme d'un diagramme UML de classes.

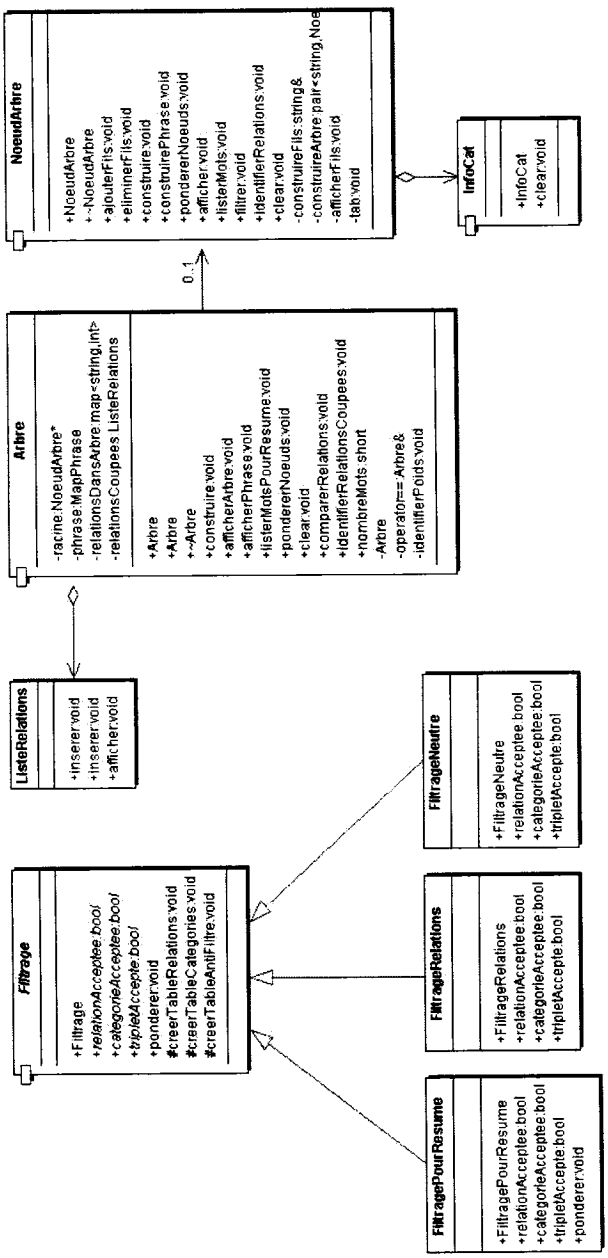


Figure 4.4 Diagramme UML de classes pour le module de filtrage/bonification.

Le design du module de filtrage permet une grande flexibilité. En effet, il fut d'ailleurs utilisé pour d'autres recherches que celles présentées dans ce mémoire (Ga-

gnon et Sylva, 2005). Outre son implémentation facile à comprendre grâce à son modèle objet, le module permet une intégration très simple de nouveaux types de filtres. Si nous désirons baser le filtrage sur de nouveaux critères (relations, catégories, ...), la classe abstraite « Filtrage » peut être héritée et permettre ainsi l'utilisation d'un filtre différent. Les méthodes relationAcceptee et categorieAcceptee sont assez intuitives et permettent de déterminer si la relation ou le mot doivent être filtrer. La méthode tripletAccepte permet d'éviter le problème avec les relations coordonnées, mentionné précédemment, et utilise l'anti-filtre pour vérifier si le triplet relation/catégorie ainsi que la catégorie suivante de l'arbre sont problématiques.

L'arbre en tant que tel est construit à l'aide de deux classes « Arbre » et « NoeudArbre ». Les phrases découpées par l'analyseur grammatical sont représentées sous forme d'une table de hashage, où la position du mot dans la phrase constitue la clé. L'arbre contient aussi une table contenant toutes les relations contenues dans l'arbre ainsi qu'une liste des relations supprimées (informations non utilisées pour les tests de ce mémoire). Finalement, les méthodes publiques disponibles permettent la construction de l'arbre à partir des phrases textes fournies par l'analyseur syntaxique, et la sortie des phrases modifiées selon la structure de l'arbre obtenu.

### 4.2.4 Ajouts à Cortex

Outre les modifications apportées à CORTEX dans le but d'en améliorer la compréhension ou de permettre l'intégration de nouveaux modules, telle l'analyse par 101, certains autres ajouts de code ont été effectués ; nous parlons ici de l'ajout d'une onzième métrique ainsi que de l'ajout d'une heuristique. L'addition de la nouvelle métrique a été grandement facilitée par la modification et la restructuration de

CORTEX, le processus ne nécessitant maintenant que la dérivation d'une classe de base abstraite, tel qu'expliqué précédemment. Finalement, l'ajout d'une heuristique constitue la dernière modification apportée à CORTEX. En remplacement à l'algorithme de décision, cette modification a pour but d'effectuer une sélection plus efficace des phrases en regard à leur poids respectif, déterminé selon les métriques utilisées. Pour ce faire, plusieurs méthodes ont été ajoutées à la classe contenant la matrice de fréquences, afin d'analyser la répartition des résultats et d'affecter une valeur finale à la phrase traitée (voir chapitre 7 pour plus de détails).

Notons en terminant que toutes ces évolutions ont fait passer le code initial de CORTEX qui était seulement de 3100 lignes, à un système beaucoup plus complet, plus simple de compréhension et mieux documenté, comportant plus de 6500 lignes de code.

# CHAPITRE 5

# RÉVISION DU FONCTIONNEMENT DE CORTEX

## 5.1 Ajout d'une métrique

La version originale de CORTEX utilise dix métriques pour sélectionner les phrases pertinentes. Nous avons décidé d'en ajouter une onzième dans le but de vérifier l'influence d'une technique démontrée à maintes reprises comme apportant de bons résultats. L'idée consiste à retenir les premières et dernières phrases des paragraphes tels que vus précédement à la section 1.3.1.3. Cette propriété détermine bien souvent à elle seule une mesure de base (section 3.2.1) afin d'évaluer la qualité d'un condenseur de textes puisqu'elle s'avère très simple à implémenter. Le présent chapitre démontrera comment CORTEX réagit à cette mesure combinée à d'autres via son système de métriques.

### 5.1.1 Compromis

CORTEX n'utilise que des textes bruts sans formattage, segmentés en phrases. Comme entrée du programme, nous ne distinguons que des phrases et des titres de sections. Pour cette raison, il s'avère impossible d'utiliser l'information sur la construction des paragraphes sans modifier la manière dont les données sont lues et insérées. Afin de tirer profit de la technique des premières et dernières phrases d'un paragraphe, sans toutefois nécessiter un changement majeur sur le logiciel, la séparation du document avec les titres de sections est utilisée. Il s'agit d'un découpage plus grossier, mais tout de même intéressant puisque conservateur par

rapport à une division par paragraphes. En effet, nous avantageons potentiellement les mêmes phrases, mais en moins grand nombre. Nous essayerons de démontrer que ce découpage peut lui aussi contribuer à l'amélioration des résultats au même titre qu'un découpage plus fin par paragraphes.

### 5.1.2 Description de la métrique

La nouvelle métrique a été baptisée « Extrémités ». Elle n'est pas basée sur la matrice fréquentielle des mots (éq. 2.1) comme les autres. Elle privilégie simplement les phrases situées en début et en fin de section. La lettre désignant cette technique de positionnement à CORTEX est **X**. Voici la description de la métrique :

$$X^{\mu} = \left\{ \begin{array}{cc} \left( \frac{2\left|\frac{P_F+P_I}{2} - P_\mu\right|}{P_F - P_I} \right)^2 & \text{si } P_F \neq P_I \\ 1 & \text{autrement} \end{array} \right\}, \text{ sachant que } P_F \geq P_I \qquad (5.1)$$

Où $P_I$ : Numéro de la première phrase de la section.
$P_F$ : Numéro de la dernière phrase de la section.
$P_\mu$ : Numéro de la phrase $\mu$ .

Les phrases au début et à la fin de la section reçoivent ainsi une note de 1, et la phrase située en son exact milieu reçoit une note 0. Une distribution quadratique est utilisée plutôt que linéaire, afin d'assurer une plus grande discrimination des phrases. En résumé : plus la phrase est près des extrémités de la section plus elle est avantagée.

### 5.1.3 Tests et résultats

Dans le but de tester l'influence de cette nouvelle métrique, les résultats avant et après la modification ont été comparés. Pour chacun des textes du corpus, trois données ont été mesurées. Nous avons expliqué à la section 3.2 de quelle manière plusieurs résumés manuels ont été obtenus et comment à partir de ces sélections humaines, des notes étaient attribuées. Dans le tableau 5.1 nous retrouvons vis-à-vis la ligne « Moyenne », la moyenne des résultats provenant des diverses sélections manuelles. Pour ce qui est des lignes « Union » et « Intersection », nous observons la note obtenue en conservant l'union ou uniquement l'intersection des phrases sélectionnées. Le tableau montre donc l'évolution des textes selon ces paramètres avec l'utilisation de **X** et de toutes les autres métriques telles qu'utilisées traditionnellement.

Tableau 5.1 Tableau comparatif des résultats, avec et sans l'utilisation de la métrique de positionnement

| | Avant (%) | Après (%) | | Avant (%) | Après (%) | |
|---|---|---|---|---|---|---|
| Cybermédias | 32.14 | 39.29 | Moyenne | 45.93 | 48.15 | Opus Dei |
| | 0.00 | 0.00 | Intersection | 8.89 | 8.89 | |
| | 42.86 | 57.14 | Union | 80.00 | 88.89 | |
| Sciences | 60.87 | 52.18 | Moyenne | 45.71 | 42.86 | Sirène |
| | 21.74 | 17.39 | Intersection | 25.71 | 20.00 | |
| | 91.30 | 78.26 | Union | 68.57 | 71.43 | |
| Travail | 20.00 | 29.33 | Moyenne | 34.85 | 37.88 | Durham |
| | 4.00 | 12.00 | Intersection | 4.59 | 9.09 | |
| | 48.00 | 56.00 | Union | 68.18 | 68.18 | |
| J'accuse | 34.92 | 39.68 | Moyenne | 56.25 | 48.75 | Épicier |
| | 9.52 | 9.52 | Intersection | 30.00 | 30.00 | |
| | 61.90 | 71.43 | Union | 95.00 | 85.00 | |
| Kanthume | 48.61 | 47.22 | Moyenne | 45.24 | 38.10 | Univers |
| | 8.33 | 8.33 | Intersection | 14.29 | 7.14 | |
| | 83.33 | 79.17 | Union | 71.43 | 64.29 | |
| Football | 54.55 | 48.48 | Moyenne | | | |
| | 18.18 | 18.18 | Intersection | | | |
| | 90.91 | 72.73 | Union | | | |

### 5.1.4 Analyse des résultats

Nous remarquons rapidement que les résultats sont très partagés d'un texte à l'autre. Nous pourrions même y voir que l'influence de l'application de la nou-

velle métrique avantage un texte sur deux, ni plus ni moins. De plus, nous pouvons regarder pour quel genre de textes les résultats sont améliorés et pour quels autres subissent une baisse de résultats ; pour ce faire, la section répertoriant les différents textes du corpus (section 3.1) pourrait s'avérer utile. Cependant, aucune grande ligne ne se démarque. Certains longs textes comme *Opus Dei* réagissent bien, d'autres comme *Sirène* non. Il en va de même pour les plus courts tels *Cybermédias* et *Football* qui sont eux aussi contradictoires. Les textes informatifs, les contes et autres genres de textes sont eux aussi départagés. Les documents avec de multiples titres de sections sont aussi souvent avantagés que désavantagés. Nous aurions pu nous attendre à un résultat contraire puisque la métrique X se base justement sur les divisions du texte en sections. À proprement parler, il ne semble pas y avoir de tendance selon le genre de document. Cependant, nous ne pouvons pas conclure immédiatement à l'inefficacité de l'ajout de la métrique. Il sera démontré pourquoi dans le prochain chapitre qui traitera du choix des métriques à appliquer. Cependant, il s'avère pertinent de rappeler que **X** employée seule demeure une mesure comparative de base (section 3.2.1).

## 5.2 Combinaison de métriques à utiliser

Nous avons vu à la section 2.1.2 que CORTEX utilise plusieurs mesures statistiques dans le but de prendre ses décisions sur les phrases à retenir. De plus, le chapitre précédent présentait l'ajout d'une métrique supplémentaire prenant en considération la structure du document. Des études (Torres-Moreno et al., 2002; Torres-Moreno et al., 2004) démontrent le pouvoir de discrimination des différentes métriques statistiques spécifiques à CORTEX. Cependant, aucune ne permet de déterminer s'il serait préférable d'écarter certaines d'entre elles. De plus, ces documents ne présentent aucune étude comparative permettant de voir l'impact réel de ce pouvoir

de discrimination.

### 5.2.1 Utilisation de toutes les métriques

Dans sa version originale, CORTEX est lancé systématiquement avec toutes les métriques qui y sont implémentées, soit les dix métriques présentées au chapitre 2. La section 2.1.3 présente la méthode utilisée pour combiner les métriques entre elles, mais aucune des études réalisées ne s'interroge à savoir s'il serait plus avantageux de se servir d'une autre combinaison plutôt que toutes ces métriques simultanément. Dans certaines circonstances, serait-il préférable d'en laisser quelques-unes de côté ? Arriverions-nous à améliorer les résultats grâce à cette astuce ? Dans ce chapitre, nous tenterons de répondre à cette interrogation en analysant les performances de toutes les combinaisons possibles, soient les métriques originales plus la métrique X que nous avons ajoutée.

### 5.2.2 Métriques redondantes

Certaines métriques avaient déjà été identifiées comme étant redondantes (St-Onge, 2003). Celles-ci sont les couples récurrents : YL, DO, IT et PF. Sur cette base, il a été décidé d'éliminer les doublons en ne conservant qu'une seule métrique par paire identifiée puisqu'en les conservant toutes, nous nous trouvons en fait à doubler le poids de ces métriques. Puisque le présent test ne vise pas à déterminer la meilleure pondération de métriques, mais bien de trouver lesquelles doivent ou ne doivent pas être présentes, quatre métriques ont été supprimées au départ : Y, O, I et P ; ce choix est basé sur les conclusions du rapport technique à leur égard. De ce fait, il reste 7 métriques à analyser.

### 5.2.3 Existe-t-il une combinaison idéale ?

Le cas idéal aurait été de pouvoir déterminer la combinaison qui, une fois fixée, permet d'obtenir un résultat optimal dans tous les documents. Bien évidemment, d'un texte à l'autre, les critères et les caractéristiques changent, il n'existe alors fort probablement pas de solution parfaite. Afin de vérifier cette hypothèse et ayant éliminé les métriques redondantes (section 5.2.2), toutes les combinaisons de métriques ont été testées avec les sept restantes (FDETLAX). De ces métriques, il en découle 127 possibilités : $2^7$, moins 1 puisque nous éliminons la combinaison ne comprenant aucune métrique. Pour chacun des onze textes (décrits à la section 3.1), un résumé a été produit en utilisant les 127 possibilités. En comparant les différentes évaluations obtenues selon la technique définie à la section 3.2, les résultats ont été triés et les combinaisons les plus performantes ont été retenues. La section suivante exposera ces résultats plus en détails, mais nous pouvons affirmer dès maintenant qu'aucune des combinaisons de métriques ne satisfait pleinement tous les documents à la fois.

### 5.2.4 Tests et résultats sur la combinaison à adopter

Soit $N_T$ textes et $N_E$ évaluateurs humains. Soit $\vec{\nu}$ le vecteur obtenu pour un évaluateur sur un texte donné, tel que $\vec{\nu} = (\nu_1, \nu_2, \ldots, \nu_{127})$ , où $\nu_j$ représente le $j^e$ meilleur résultat, si nous supposons le vecteur comme étant trié en ordre décroissant. Soit $n$ le nombre de combinaisons à sélectionner parmi les meilleures. Le programme sélectionne d'abord les $n$ meilleures combinaisons de métriques. S'il existe des valeurs $\nu_{n+1}$, $\nu_{n+2}$, $\ldots$, $\nu_{n+k}$ telles que $\nu_n = \nu_i$, $n+1 < i < n+k$, celles-ci seront aussi sélectionnées. Soit $m = n + k$, la sélection obtenue sera le vecteur $\vec{\nu}_m = (\nu_1, \ldots, \nu_n, \nu_{n+1}, \ldots, \nu_m)$. Sachant qu'une valeur $m$ différente est trouvée pour chacun des évaluateurs et chacun des textes, nous obtenons au total $T = \sum_{i=1}^{N_E} \sum_{j=1}^{N_T} m_j^i$ résultats, où chacun des résultats de $T$ est représenté par une

note et une combinaison de métrique qui y sont associées.

Une fois ces $T$ résultats obtenus, nous vérifions la fréquence d'apparition des combinaisons de métriques. Certaines s'y retrouvent particulièrement souvent, et d'autres très rarement, voire jamais. Les tableaux 5.2, 5.3 et 5.4 démontrent respectivement la fréquence des combinaisons les plus courantes pour $n = 1$, $n = 10$ et $n = 20$. Les figures 5.1, 5.2 et 5.3 illustrent ces tableaux en affichant la répartition du nombre de combinaisons.

Tableau 5.2 Tableau de la fréquence des métriques apparaissant comme premier résultat ($n = 1$, $T = 238$)

| Combinaison | Fréquence d'apparitions parmi les $T$ valeurs | Combinaison | Fréquence d'apparitions parmi les $T$ valeurs | Combinaison | Fréquence d'apparitions parmi les $T$ valeurs |
|---|---|---|---|---|---|
| FDEA | 5 | FDELA | 5 | FDLA | 5 |
| TLA | 5 | ELA | 4 | FDETLA | 4 |
| FDL | 4 | FDTLA | 4 | FELA | 4 |
| FETA | 4 | DET | 3 | DETA | 3 |
| EL | 3 | ET | 3 | ETL | 3 |
| ETX | 3 | FD | 3 | FDELX | 3 |
| FDETA | 3 | FDLX | 3 | FDT | 3 |
| FDTL | 3 | FETLA | 3 | FETLX | 3 |
| FEX | 3 | LAX | 3 | T | 3 |
| ... (80 combinaisons à 2 et moins) | | | | | |

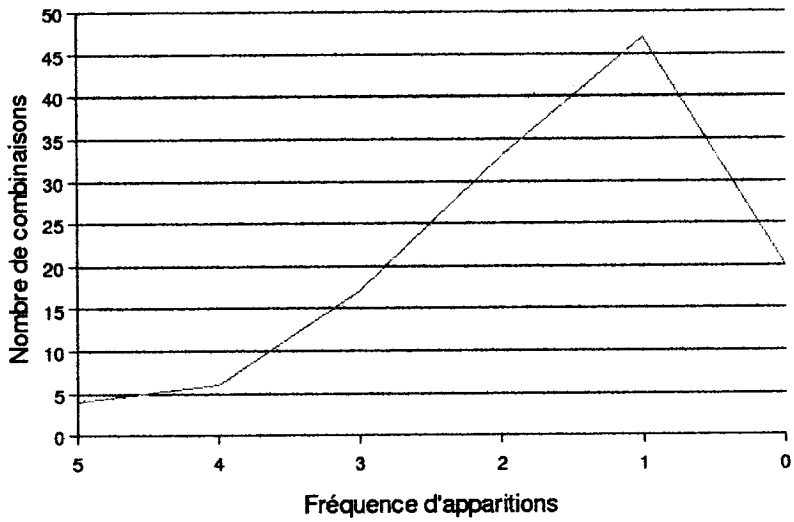


Figure 5.1 Distribution du nombre de combinaisons de métriques selon leur fréquence d'apparition pour $n = 1$, $T = 238$ (synthèse du tableau 5.2).

Tableau 5.3 Tableau de la fréquence des métriques apparaissant dans les 10 premiers résultats ($n = 10,\ T = 757$)

| Combinaison | Fréquence d'apparitions parmi les $T$ valeurs | Combinaison | Fréquence d'apparitions parmi les $T$ valeurs | Combinaison | Fréquence d'apparitions parmi les $T$ valeurs |
|---|---|---|---|---|---|
| ETX | 12 | DEX | 11 | FTX | 11 |
| FELA | 10 | TLA | 10 | DETA | 9 |
| ELA | 9 | ETLA | 9 | FDA | 9 |
| FDL | 9 | FDLA | 9 | FDTLX | 9 |
| FETX | 9 | TLX | 9 | AX | 8 |
| DETLA | 8 | DETLX | 8 | ELX | 8 |
| ETA | 8 | FDE | 8 | FDEA | 8 |
| FDELA | 8 | FDELX | 8 | FDETLA | 8 |
| FDT | 8 | FDTLA | 8 | FETA | 8 |
| FETLX | 8 | FLA | 8 | LAX | 8 |
| ... (97 combinaisons à 7 et moins) | | | | | |

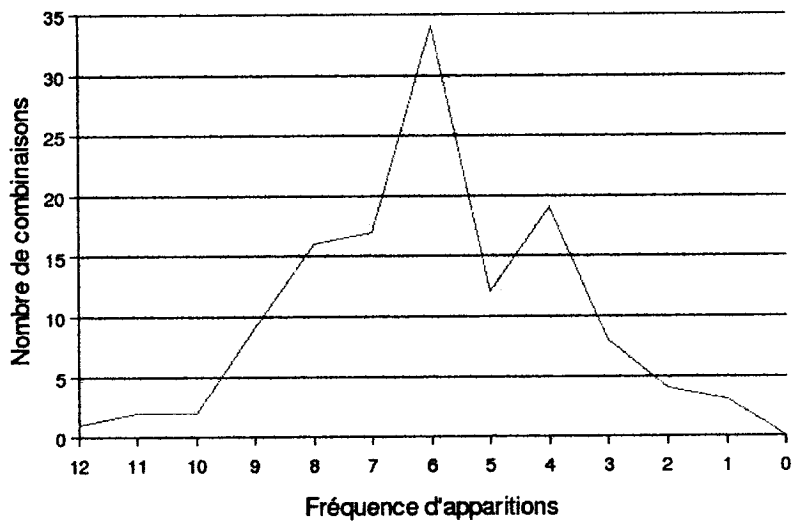


Figure 5.2 Distribution du nombre de combinaisons de métriques selon leur fréquence d'apparition pour $n = 10$, $T = 757$ (synthèse du tableau 5.3).

Tableau 5.4 Tableau de la fréquence des métriques apparaissant dans les 20 premiers résultats ($n = 20,\ T = 1\,309$)

| Combinaison | Fréquence d'apparitions parmi les $T$ valeurs | Combinaison | Fréquence d'apparitions parmi les $T$ valeurs | Combinaison | Fréquence d'apparitions parmi les $T$ valeurs |
|---|---|---|---|---|---|
| DETLAX | 18 | ETX | 17 | FDLAX | 16 |
| DELAX | 15 | DTLAX | 15 | FDTX | 15 |
| FETX | 15 | ELX | 14 | ETLA | 14 |
| ETLAX | 14 | FDEAX | 14 | FDETLA | 14 |
| FDLX | 14 | FTX | 14 | DETAX | 13 |
| DEX | 13 | ETAX | 13 | FDE | 13 |
| FDEX | 13 | FDL | 13 | FDTAX | 13 |
| FETA | 13 | TLX | 13 | AX | 12 |
| DETA | 12 | DETLA | 12 | DTLA | 12 |
| ELA | 12 | ETLX | 12 | FDETAX | 12 |
| FDLA | 12 | FDTLA | 12 | FELA | 12 |
| FETLX | 12 | FEX | 12 | | |
| ... (92 combinaisons à 11 et moins) | | | | | |

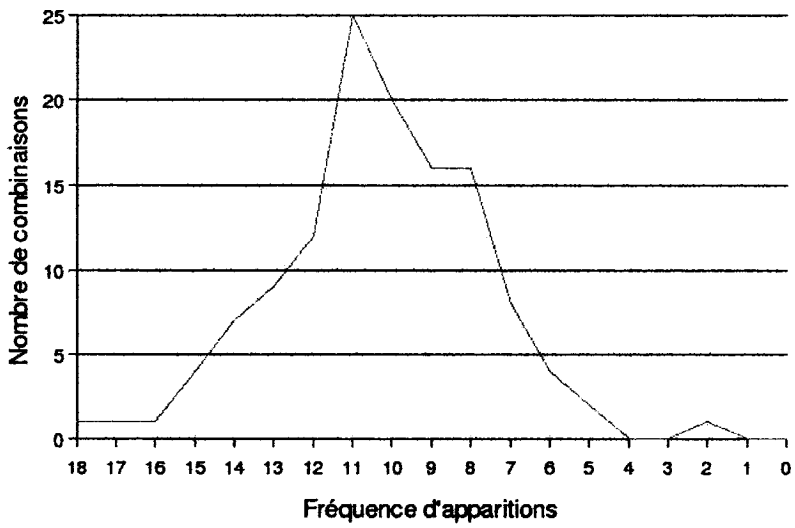


Figure 5.3 Distribution du nombre de combinaisons de métriques selon leur fréquence d'apparition pour $n = 20$, $T = 309$ (synthèse du tableau 5.4).

Chacun des tableaux représente la fréquence à laquelle apparaît la combinaison étudiée. Par exemple, le tableau 5.2 répertorie le nombre de fois que chacune d'entre elles ressort comme étant la plus performante. Si nous réalisons l'intersection entre les combinaisons que nous retrouvons dans les tableaux 5.2, 5.3 et 5.4, nous retrouvons un total de 10 métriques présentes dans les trois tableaux simultanément. Ces combinaisons sont : ETX, FDETLA, FDL, FETA, DETA, ELA, FDLA, FDTLA, FELA et FETLX.

Nous pourrions penser ne tenir compte que de cette information, cependant certaines combinaisons de métriques se classent généralement bien dans presque tous les cas, alors que les plus performantes du tableau 5.2 ne se retrouvent que très rarement en haute position. Pour cette raison, il s'avère pertinent de prioriser les combinaisons hautement classés aux tableaux 5.3 et 5.4, qui répertorient respectivement les 10 et les 20 combinaisons offrant les meilleurs résultats. De cette façon,

nous pouvons démarquer 3 métriques sur les 10 restantes : ETX, FDETLA et FDL. De plus, ces trois combinaisons sont particulièrement intéressantes pour effectuer nos tests puisqu'elles utilisent des métriques sensiblement différentes.

### 5.2.5 Conclusion sur la combinaison à utiliser

Lorsque nous analysons les combinaisons idéales d'un texte à un autre, nous remarquons malheureusement que la plus performante ne se révèlent être presque jamais la même à quelques exceptions près. Malgré tout, ces exceptions peuvent difficilement être considérées puisque, généralement, les performances obtenues avec celles-ci sont décevantes. En observant davantage les données, nous notons cependant que certaines se classent généralement bien dans la plupart des situations, sans nécessairement s'avérer les meilleures dans tous les cas. De celles-ci, la combinaison qui se démarque le plus se trouve à être ETX. Quelquefois au premier rang, mais d'autant plus souvent présente dans les 10 ou 20 meilleures combinaisons sur 127.

De toute évidence, posséder une connaissance *a priori* sur le texte à résumer provenant de tests antérieurs pourrait permettre de déterminer les meilleures métriques à utiliser, cependant le but de ce chapitre était de déterminer une combinaison à employer dans un cas général. À partir de ces observations, il apparaît préférable d'abandonner l'habitude de faire fonctionner CORTEX avec toutes ses métriques et il s'avère beaucoup mieux, dans le cas général, d'utiliser les métriques ETX qui ont démontré, par les expériences, qu'elles se classent bien dans la plupart des cas.

Finalement, nous pouvons vérifier nos affirmations avec le tableau 5.5 qui nous montre les résultats concrets obtenus avec les diverses combinaisons recommandées. La moyenne des onze textes est calculée et reportée dans le tableau. Nous pouvons ainsi constater l'efficacité des métriques suggérées.

Tableau 5.5 Tableau répertoriant la moyenne des résultats obtenus pour tous les textes, à l'aide des 10 métriques suggérées.

| Combinaison | Moyenne obtenue |
|---|---|
| **ETX** | **43** |
| **FDETLA** | **43** |
| FDL | 42 |
| FETA | 41 |
| DETA | 42 |
| ELA | 42 |
| FDLA | 42 |
| FDTLA | 42 |
| FELA | 42 |
| FETLX | 41 |

# CHAPITRE 6

# TRAITEMENT DES PRONOMS

## 6.1 Introduction

Lorsque nous écrivons un document, qu'il soit scientifique ou littéraire, nous utilisons couramment diverses anaphores[1]. Le contexte entourant une phrase contenant une anaphore s'avère alors primordial, puisque sa compréhension dépend d'éléments d'information fournis par les phrases précédentes. Cela signifie que pour retenir les segments importants d'un texte, nous nous devons de conserver en plus des phrases désirées, celles qui sont nécessaires au contexte en les insérant entièrement aux côtés de la phrase retenue. Une autre façon de faire consiste à remplacer toutes les anaphores contenues dans la phrase par l'idée ou le mot auquel elles se rattachent. De cette manière, une phrase seule aura tout son sens.

Ce chapitre traite de l'influence du traitement anaphorique, plus particulièrement celui des pronoms, sur la sélection des phrases dans le but de produire un résumé. En effet, au lieu de retenir les phrases du texte original et d'ensuite traiter les pronoms afin de conserver le sens de la phrase, le traitement de ces pronoms est effectué dès le départ. Le document original est ainsi modifié et c'est parmi ces nouvelles phrases créées que les plus importantes seront choisies. De cette façon, nous obtenons des phrases qui, même employées seules, pourront donner au lecteur une bonne idée du contexte dans lequel elles étaient utilisées. De plus, il va sans dire qu'en modifiant dès le départ les anaphores du texte (dans le cas présent : les

[1]Élément de discours pour lequel il est nécessaire, si nous voulons pouvoir l'interpréter, de se reporter à un autre élément du même discours. *Source : http://www.granddictionnaire.com*

pronoms), nous modifions aussi potentiellement le nombre et la position des mots dans le texte et par conséquent la manière dont la phrase sera traitée par le logiciel de résumé. Ce chapitre démontrera donc si, dans le contexte bien précis à l'intérieur duquel CORTEX est utilisé, ce pré-traitement aide le logiciel à mieux sélectionner ses phrases.

## 6.2 Résolution des pronoms

Certaines méthodes classiques existent et servent expressément à effectuer automatiquement la substitution des pronoms (Lappin et Leass, 1994; Hobbs, 1986; Grosz et al., 1995; Ge et al., 1998). Il existe même certains logiciels, dont un utilisant la technique de Lappin et Leass[2], afin de résoudre les anaphores. Cependant, pour les fins des présents tests, la résolution des pronoms a été effectuée manuellement. La raison en est bien simple : nous cherchons ici à voir si un tel traitement contribuera à améliorer les performances du logiciel. Nous voulons donc préalablement tester son efficacité. Sachant qu'un logiciel automatique n'est certainement pas parfait mais qu'il permet de remplacer une grande partie des pronoms, nous obtiendrons ici des résultats provenant d'un cas idéal. Typiquement, avec l'état actuel de la technologie, on obtient une précision de plus de 90% avec les systèmes de traitement automatique des anaphores (Mitkov, 2002).

Voici maintenant l'idée principale qui a été utilisée pour traiter les pronoms au cours de ces expérimentations. Tous ceux se référant à une entité nommée ont été substitués. Par entité nommée, il faut comprendre tous les pronoms qui peuvent être directement remplacés par un nom ou par une courte expression. C'est donc dire que les pronoms se référant à des idées ou à des phrases antérieures entières

[2] `http://www.comp.nus.edu.sg/~qiul/NLPTools/JavaRAP.html`

n'ont pas été remplacés. Par exemple,

```
Le chien court dans la rue.
Sa balle a roulé très loin.
```

deviendrait

```
Le chien court dans la rue.
La balle du chien a roulé très loin.
```

Tandis que dans les phrases

```
Nous devrions aller manger.
C'est une très bonne idée.
```

le « *C'* » resterait inchangé. De plus, si le pronom réfère à une entité qui a déjà été nommée dans la même phrase, il ne sera pas remplacé afin d'éviter des répétitions qui pourraient rendre un texte complètement illisible.

L'idée de substituer les pronoms a pour but de permettre à une phrase de conserver son sens lorsqu'elle est employée seule ; ainsi, si le pronom réfère à une entité déjà nommée dans la même phrase, le sens de celui-ci est connu même si la phrase est employée en dehors de son contexte. Ensuite, si une anaphore se répète plusieurs fois dans une même phrase, il ne sera remplacé qu'une seule fois. Voici un dernier exemple pour illustrer ceci :

```
Le chien court dans la rue.
Sa balle a roulé très loin, puisqu'il aime se la faire lan-
cer très fort.
```

deviendra

```
Le chien court dans la rue.
La balle du chien a roulé très loin, puisqu'il aime se la faire
lancer très fort.
```

et **non**

```
Le chien court dans la rue.
La balle du chien a roulé très loin, puisque le chien aime se
faire lancer la balle très fort.
```

## 6.3 Résultats et analyse

Afin de bien cerner l'impact de la substitution des pronoms sur chacun des onze textes du corpus, regardons le tableau 6.1 qui nous montre le nombre total de phrases dans le texte, le nombre d'entre elles traitées pour les pronoms, ainsi que le pourcentage qui résulte de ces deux données.

Tableau 6.1 Caractéristiques des documents du corpus en rapport aux anaphores

| | Nombre total de phrases | Nombre de phrases traitées pour anaphores | Pourcentage des phrases traitées |
|---|---|---|---|
| Durham | 210 | 52 | 25 % |
| Épicier | 191 | 46 | 24 % |
| Kanthume | 230 | 38 | 17 % |
| Football | 102 | 36 | 35 % |
| J'Accuse | 207 | 49 | 24 % |
| Univers | 139 | 16 | 12 % |
| Opus Dei | 443 | 151 | 34 % |
| Cybermédias | 62 | 32 | 52 % |
| Sciences | 225 | 29 | 13 % |
| Sirène | 346 | 157 | 45 % |
| Travail | 244 | 21 | 9 % |

Le tableau 6.2 illustre à son tour de quelle manière se comporte CORTEX avec les phrases traitées. Selon le nombre total de phrases modifiées, nous verrons maintenant la proportion d'entre elles qui se retrouvent au résumé.

Tableau 6.2 Nombre de phrases traitées pour les anaphores et retenues pour former le résumé.

| | ETX | | FDL | | FDETLA | |
|---|---|---|---|---|---|---|
| /Tot. ph. traitées | Avant | Après | Avant | Après | Avant | Après |
| Durham /52 | 1 | 3 | 3 | 2 | 3 | 3 |
| Épicier /46 | 4 | 6 | 5 | 8 | 4 | 7 |
| Kanthume /38 | 8 | 9 | 6 | 7 | 7 | 9 |
| Football /36 | 6 | 8 | 6 | 7 | 5 | 8 |
| J'Accuse /49 | 5 | 6 | 7 | 11 | 5 | 9 |
| Univers /16 | 1 | 2 | 2 | 3 | 2 | 3 |
| Opus Dei /151 | 7 | 13 | 4 | 17 | 8 | 15 |
| Cybermédias /32 | 3 | 2 | 2 | 3 | 2 | 3 |
| Sciences /29 | 2 | 3 | 4 | 5 | 3 | 5 |
| Sirène /157 | 16 | 26 | 18 | 28 | 16 | 30 |
| Travail /21 | 1 | 1 | 2 | 3 | 2 | 2 |

Nous remarquons du premier coup d'œil que la très grande majorité du temps, le nombre de ces phrases est plus important dans le résumé obtenu après le traitement des pronoms. C'est donc dire que les phrases que nous traitons ont davantage tendance à faire partie du résumé par la suite. Nous pouvons poser comme hypothèse que la raison pour laquelle nous retrouvons un plus grand nombre de phrases une fois traitées est due au fait que lorsque nous remplaçons des pronoms dans une phrase pour y introduire d'autres mots, nous enlevons par le fait même des mots fonctionnels qui étaient de toute manière retirés au filtrage, lors du pré-traitement.

Ces mots sont ensuite remplacés par d'autres qui possèdent beaucoup plus de valeur et qui se retrouvent vraisemblablement dans la matrice de fréquences. Nous nous trouvons ainsi à augmenter le poids de la phrase et nous avons donc beaucoup plus de chances de la retrouver au sein du résumé final. Cependant, il paraît important de noter que cet état de fait ne s'avère pas nécessairement être une bonne chose. Le fait de retrouver les phrases traitées au niveau du résumé ne signifie pas que nous ajoutons des phrases d'importance au résumé. En effet, rien n'indique qu'une phrase traitée pour anaphore est pertinente au contexte du document.

À ce sujet, observons maintenant l'influence du traitement anaphorique sur les combinaisons de métriques présentées au chapitre 5 comme étant les meilleures. Voyons si, concrètement, les résultats s'en trouvent affectés.

Tableau 6.3 Résultats obtenus avec les trois métriques les plus avantageuses, avant et après le traitement des pronoms. Il s'agit d'une moyenne des notes des différents résumés manuels.

| | Moyenne (%) | | | | | |
|---|---|---|---|---|---|---|
| | ETX | | FDL | | FDETLA | |
| | Avant | Après | Avant | Après | Avant | Après |
| Durham | 41 | 43 | 37 | 35 | 35 | 37 |
| Épicier | 50 | 42 | 57 | 50 | 58 | 48 |
| Kanthume | 43 | 45 | 41 | 42 | 49 | 51 |
| Football | 55 | 55 | 51 | 45 | 55 | 51 |
| J'Accuse | 43 | 41 | 38 | 41 | 35 | 41 |
| Univers | 48 | 43 | 41 | 36 | 45 | 38 |
| Opus Dei | 40 | 42 | 47 | 49 | 46 | 48 |
| Cybermédias | 28 | 19 | 33 | 19 | 24 | 28 |
| Sciences | 48 | 48 | 55 | 52 | 61 | 55 |
| Sirène | 51 | 40 | 47 | 29 | 46 | 36 |
| Travail | 27 | 27 | 20 | 19 | 20 | 20 |
| Moyenne | 43 | 40 | 42 | 38 | 43 | 41 |

Nous remarquons dans le tableau 6.3 que les résultats tendent généralement à l'ambivalence. Alors que certains textes dont *Kanthume* ou *Opus Dei* dénotent une nette amélioration, d'autres tel *Épicier*, *Univers* ou *Sirène* subissent une dégringolade. Nous avons tenté de saisir le sens de ces résultats en observant à la fois les documents originaux ainsi que la constitution des phrases (nombre de mots, types de mots, etc.), sans y détecter pour autant de règles apparentes. Les autres documents varient généralement d'une métrique à une autre, quelques fois en bien, quelques fois en mal. Nous pouvons cependant noter, grâce à l'évolution de la moyenne, la tendance à la baisse des résultats généraux.

Nous avons constaté l'effet de la substitution des pronoms sur la moyenne des résultats obtenus par les évaluateurs humains. Regardons maintenant le tableau 6.4, qui montre les résultats obtenus, si, cette fois, au lieu de comparer avec la moyenne, nous prenons l'intersection des sélections établies par les trois évaluateurs (les phrases retenues simultanément par les trois personnes). De cette manière, nous obtiendrons uniquement les phrases qui importent vraiment, puisque chacun des trois juges a cru bon de les sélectionner.

Tableau 6.4 Résultats obtenus avec les trois métriques les plus avantageuses, avant et après le traitement des pronoms. Il s'agit de l'intersection entre les sélections des trois évaluateurs humains.

| | Intersection (%) | | | | | |
|---|---|---|---|---|---|---|
| | ETX | | FDL | | FDETLA | |
| | Avant | Après | Avant | Après | Avant | Après |
| Durham | 9 | 9 | 5 | 5 | 5 | 5 |
| Épicier | 20 | 10 | 25 | 25 | 30 | 25 |
| Kanthume | 4 | 4 | 4 | 0 | 8 | 8 |
| Football | 18 | 18 | 18 | 18 | 18 | 27 |
| J'Accuse | 10 | 19 | 14 | 24 | 10 | 24 |
| Univers | 7 | 7 | 7 | 7 | 14 | 7 |
| Opus Dei | 9 | 9 | 11 | 11 | 9 | 9 |
| Cybermédias | 0 | 0 | 0 | 0 | 0 | 0 |
| Sciences | 17 | 17 | 22 | 17 | 22 | 22 |
| Sirène | 26 | 11 | 31 | 14 | 26 | 17 |
| Travail | 4 | 4 | 4 | 4 | 4 | 4 |
| Moyenne | 11 | 10 | 13 | 11 | 13 | 13 |

Après l'analyse du tableau 6.4, nous arrivons à la même conclusion que celle établie pour le tableau 6.3. Règle générale, nous remarquons toujours une détérioration des

résultats, à la différence cette fois-ci que les documents qui paraissaient avantagés précédemment ne le sont plus en considérant l'intersection.

## 6.4 Insertion de phrases

Nous avons pu remarquer dans les sections précédentes la diminution, bien que très faible, de la qualité de sélection des phrases. Nous pouvons donc nous questionner à savoir s'il ne serait pas plus approprié de n'insérer que les phrases qui précèdent celles contenant les pronoms tel qu'expliqué à la section 1.5.1, afin d'en conserver le sens. Le tableau 6.5 montre les résultats obtenus en comparant ces deux techniques.

Tableau 6.5 Comparaison entre la méthode d'insertion des phrases précédentes et la substitution des pronoms

| | Moyenne (%) | | | | | |
|---|---|---|---|---|---|---|
| | ETX | | FDL | | FDETLA | |
| | Insertion | Substitition | Insertion | Substitition | Insertion | Substitition |
| Durham | 41 | 43 | 31 | 35 | 32 | 37 |
| Épicier | 50 | 42 | 57 | 50 | 58 | 48 |
| Kanthume | 46 | 45 | 36 | 42 | 46 | 51 |
| Football | 45 | 55 | 42 | 45 | 42 | 54 |
| J'Accuse | 41 | 41 | 36 | 41 | 36 | 41 |
| Univers | 38 | 43 | 31 | 36 | 38 | 38 |
| Opus Dei | 38 | 42 | 45 | 49 | 38 | 48 |
| Cybermédias | 19 | 19 | 29 | 19 | 28 | 28 |
| Sciences | 42 | 48 | 42 | 52 | 51 | 55 |
| Sirène | 40 | 40 | 33 | 29 | 39 | 36 |
| Travail | 31 | 27 | 25 | 19 | 25 | 20 |
| Moyenne | 39 | 40 | 37 | 38 | 39 | 41 |

Les résultats illustrent un fort avantage de la substitution des pronoms en ce qui touche au nombre de textes avantagés, en plus de produire une augmentation de la moyenne globale de tous les documents. Il nous porte donc à conclure qu'il s'agit d'une méthode beaucoup plus efficace que le simple ajout de la phrase précédente, insérée afin de conserver le sens du contexte. La raison de ce net avantage peut s'illustrer à l'aide d'un exemple. Supposons les trois phrases suivantes :

1. `Le chien court dans la rue.`
2. `Sa balle a roulé très loin et il n'a pu la retrouver.`
3. `Malchanceux, il a de plus laissé tomber durant sa course son jouet préféré dans l'égout donnant sur cette rue.`

Si on nous demandait de retenir seulement deux des trois phrases présentes, la plupart d'entres nous choisirions les phrases #2 et #3. Cependant, à elles seules elles ne permettent pas de conserver le contexte et il devient même impossible de savoir s'il est question d'un chien ! La phrase #1 devrait ainsi, pour le contexte, impérativement être retenue au détriment d'une des autres phrases fondamentalement plus pertinentes. Prenons maintenant les mêmes phrases, mais traitées pour anaphores. Le résumé admet maintenant la possibilité de ne contenir que les phrases #2 et #3 sans nécessiter le soutien de la phrase #1. Ainsi, nous retrouvons de l'espace pour inclure les deux phrases initialement désirées.

## 6.5 Conclusion sur les pronoms

En conclusion de tous ces tests, nous pouvons affirmer que le traitement des anaphores et plus particulièrement des pronoms ne permet pas d'améliorer le choix des phrases effectué par CORTEX. Dans certaines situations, les résultats peuvent s'y trouver améliorés, mais dans un nombre équivalent, voire supérieur de situations, c'est l'inverse qui se produit alors qu'aucun paramètre ou caractéristique ne semble

responsable de cette variation de comportement. Nous avons discuté de résultats similaires à la section 1.5.2, nos résultats tendent donc à confirmer cette tendance.

Le présent document ne démontre en rien l'influence sur la qualité du résumé, mais bien sur la pondération des phrases et le choix de celles-ci. En effet, nous devons faire la distinction entre ces deux éléments puisqu'il semble possible, et même hautement probable, que le traitement des anaphores améliore la qualité du résumé sans pour autant améliorer le choix des phrases. En effet, nous retrouvons un document à partir duquel les phrases sont presque aussi bien sélectionnées, mais en plus, les phrases résultantes ne contiennent plus de pronoms et s'avèrent par conséquent plus lisibles.

Finalement, nous avons constaté que bien que la technique de substitution des pronoms cause souvent une diminution de la pertinence des phrases sélectionnées, cette technique s'avère tout de même moins désastreuse que la simple insertion de phrase dans le but de conserver le contexte des pronoms utilisés. Dépendamment de l'utilisation faite de CORTEX, si un traitement quelconque doit être effectué sur les pronoms, il vaut peut-être la peine d'opter sur la technique de substitution, bien qu'un peu plus complexe, si nous désirons affecter le moins possible la sélection de phrases.

# CHAPITRE 7

# CHOIX AUTOMATIQUE DES MÉTRIQUES

Nous avons constaté au chapitre 5 qu'il était souhaitable de laisser de côté certaines métriques. De plus, le même chapitre illustre que dépendamment du document traité, les métriques ne s'avéraient pas d'une efficacité constante. Considérant ces observations, il pourrait être intéressant de déterminer automatiquement, lors de l'exécution du programme, quelles sont les métriques les mieux adaptées pour le genre de texte considéré. Ainsi, la décision d'utiliser une seule, plusieurs ou toutes les métriques, serait prise au moment de l'exécution. Une approche par apprentissage pourrait s'avérer efficace dans ce type de situation, mais puisque nous désirons conserver CORTEX le plus indépendant possible des textes traités, nous devons songer à une autre solution.

De plus, il nous faut laisser de côté l'idée de comparer les métriques les unes aux autres. Nous aurions pu imaginer une technique comparant entre elles les mesures établies pour chacune des métriques afin de déterminer lesquelles possèdent potentiellement plus d'influence que les autres. Cependant, le choix de laisser tomber cette idée repose sur le fait que toutes les métriques sont constituées de caractéristiques bien distinctes. Par conséquent, il se trouve difficile de départager, uniquement à partir du poids des phrases, moyennes ou écart-types, les métriques qui s'avèrent potentiellement pertinentes et celles qui ne le sont pas. Bien que les valeurs soient normalisées, chacune des métriques possède des caractéristiques bien particulières et la répartition des poids est difficilement comparable malgré le fait que les valeurs se situent entre les bornes $[0, 1]$. Par exemple, regardons la métrique de l'angle entre le titre de section et la phrase (métrique A) : la grande majorité

des phrases possèdent une note située entre 0.0 et 0.1 alors que seulement quelques-unes (les phrases incluant certains mots du titre) possède une note supérieure à ces valeurs. La figure 7.1 illustre un exemple de la distribution de la métrique A pour le texte *Cybermédias.*

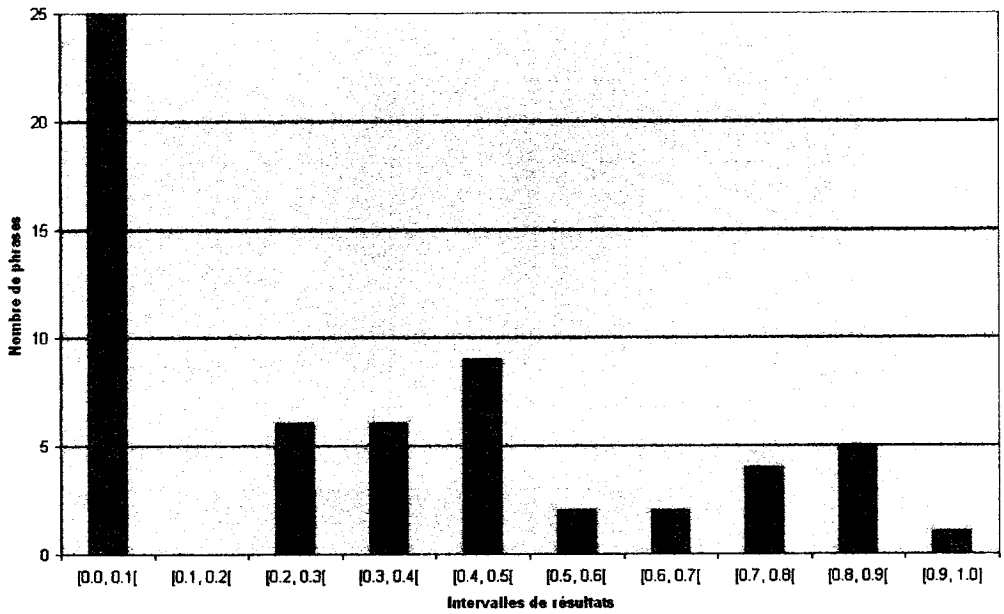


Figure 7.1 Distribution des valeurs normalisées pour la métrique A, pour le texte *Cybermédias.*

À l'opposé maintenant, prenons la métrique de fréquence des mots (métrique F). Nous remarquons que presque toutes les phrases ont une note supérieure à 0.1. La figure 7.2 illustre un exemple de la distribuion de la métrique F, toujours pour le texte *Cybermédias.*

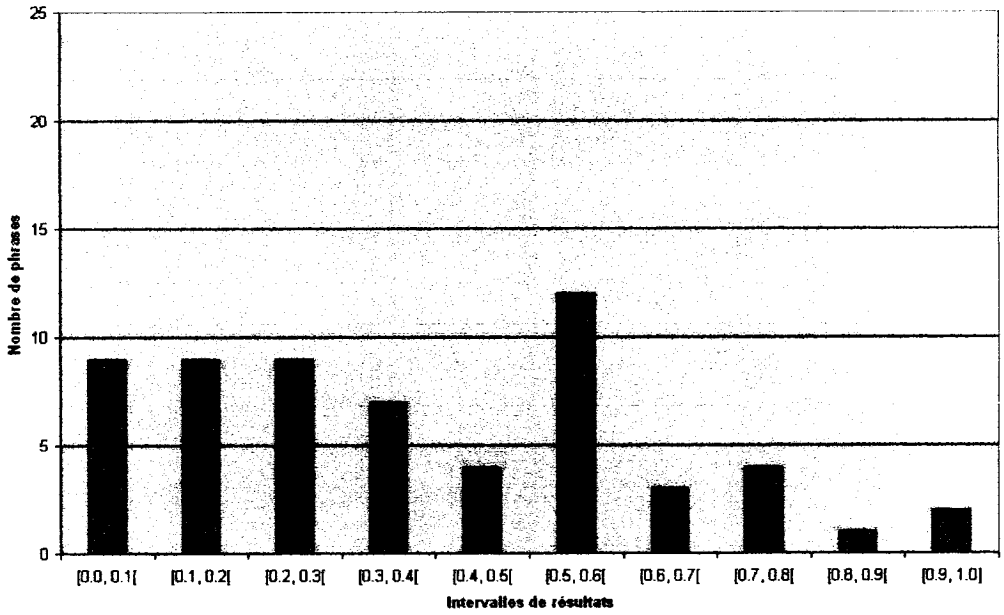


Figure 7.2 Distribution des valeurs normalisées pour la métrique F, pour le texte *Cybermédias.*

Avec les exemples illustrés aux figures 7.1 et 7.2, il devient évident qu'il est difficile de comparer directement les deux métriques puisque bien qu'elles soient normalisées, leur distributions sont totalement différentes. Il en est de même pour la majorité d'entre elles.

Puisque nous tentons d'éviter d'utiliser des méthodes par apprentissage et que les seules informations que nous possédons lors de l'exécution sont des valeurs numériques associées aux phrases du texte, l'idée retenue consiste à utiliser une méthode heuristique pour résoudre le problème. Deux approches ont été abordées, malheureusement les sections à venir nous montreront qu'une seule s'est avérée concluante. Nous verrons dans les lignes qui suivent la description du cheminement suivi pour l'élaboration de ces méthodes.

## 7.1 Analyse des poids calculés pour chacune des métriques

Tel que nous l'avons vu au chapitre 2, afin de produire un résumé, un poids est donné pour toutes les métriques, et ce pour chacune des phrases du texte. Ces valeurs sont calculées en compilant pour chacune des phrases les résultats obtenus à partir des différentes métriques ; il s'avère ensuite possible de déterminer quelles phrases représentent les plus importantes du document. À l'opposé, dans ce chapitre nous essayerons de démarquer non pas les phrases les plus pertinentes, mais les métriques les plus pertinentes. Nous compilerons les résultats d'une autre manière, soit en représentant sous forme de moyenne les valeurs pour chacune des métriques et non chacune des phrases. Nous pouvons voir ce processus comme étant la recherche des métriques les plus utiles plutôt que les phrases les plus pertinentes !

L'intention derrière cette première technique consiste à observer un à un les poids attribués par les différentes métriques aux phrases. Ensuite, la valeur numérique calculée par la métrique est comparée avec un seuil choisi dans le but de décider si la métrique doit être retenue ou non pour cette phrase. Le seuil de chaque métrique est différent et propre à chacune d'elles. Le seuil des métriques pourrait être déterminé au fil des expériences.

Regardons maintenant la moyenne obtenue pour chacune des métriques lors du résumé des onzes textes (tableau 7.1), ainsi que l'écart-type de ces moyennes (tableau 7.2).

Tableau 7.1 Moyenne des poids normalisés obtenue pour chacune des métriques.

| Métriques | F | D | E | T | L | A | X |
|---|---|---|---|---|---|---|---|
| Durham | 0.2930 | 0.1907 | 0.2883 | 0.2803 | 0.1524 | 0.02945 | 0.3365 |
| Épicier | 0.2288 | 0.1604 | 0.3010 | 0.3127 | 0.1180 | 0.02103 | 0.3453 |
| Kanthume | 0.2304 | 0.2205 | 0.2170 | 0.2113 | 0.07257 | 0.1284 | 0.3614 |
| Football | 0.1612 | 0.2023 | 0.2537 | 0.3040 | 0.09481 | 0.09569 | 0.3405 |
| J'Accuse | 0.2386 | 0.2165 | 0.2535 | 0.2475 | 0.09568 | 0.03286 | 0.3423 |
| Univers | 0.2949 | 0.3320 | 0.3406 | 0.3392 | 0.1748 | 0.09052 | 0.3384 |
| Opus Dei | 0.2272 | 0.2743 | 0.2966 | 0.2696 | 0.1359 | 0.1227 | 0.3741 |
| Cybermédias | 0.3672 | 0.3586 | 0.3660 | 0.3405 | 0.2264 | 0.3037 | 0.3259 |
| Sciences | 0.2332 | 0.2318 | 0.3370 | 0.3475 | 0.1407 | 0.1145 | 0.3347 |
| Sirène | 0.2545 | 0.2866 | 0.3137 | 0.3241 | 0.1278 | 0.06375 | 0.3345 |
| Travail | 0.2326 | 0.2086 | 0.3431 | 0.3641 | 0.1671 | 0.1424 | 0.3629 |

Tableau 7.2 Écart-type des poids normalisés obtenu pour chacune des métriques.

| Métriques | F | D | E | T | L | A | X |
|---|---|---|---|---|---|---|---|
| Durham | 0.1846 | 0.1541 | 0.1924 | 0.1975 | 0.1759 | 0.1207 | 0.3010 |
| Épicier | 0.2005 | 0.1313 | 0.2191 | 0.2278 | 0.1602 | 0.09546 | 0.3051 |
| Kanthume | 0.1382 | 0.1492 | 0.1314 | 0.1362 | 0.09445 | 0.2030 | 0.3235 |
| Football | 0.1410 | 0.1675 | 0.1828 | 0.2317 | 0.1372 | 0.1870 | 0.3055 |
| J'Accuse | 0.1922 | 0.1984 | 0.2003 | 0.2009 | 0.1385 | 0.1244 | 0.3046 |
| Univers | 0.2011 | 0.2088 | 0.2061 | 0.2087 | 0.1913 | 0.1689 | 0.3022 |
| Opus Dei | 0.1316 | 0.2190 | 0.1996 | 0.2170 | 0.1478 | 0.2578 | 0.3347 |
| Cybermédias | 0.2465 | 0.2471 | 0.2347 | 0.2290 | 0.2303 | 0.3093 | 0.3039 |
| Sciences | 0.1656 | 0.2003 | 0.2152 | 0.2399 | 0.1629 | 0.1588 | 0.2993 |
| Sirène | 0.1714 | 0.2009 | 0.2017 | 0.2158 | 0.1449 | 0.1672 | 0.3000 |
| Travail | 0.1357 | 0.1550 | 0.1799 | 0.2056 | 0.1580 | 0.1522 | 0.3238 |

Évidemment, tel qu'expliqué au début de ce chapitre, il s'avère peu utile d'utiliser ces poids moyens pour affirmer qu'une certaine métrique semble plus importante qu'une autre, uniquement sur la base du poids moyen. Cependant, nous verrons que ces valeurs peuvent s'avérer utiles pour notre heuristique.

Il pourrait être intéressant de comparer ces résultats avec les évaluations subjectives des résumés effectués à la main. Nous identifierons, pour chacun des textes, les métriques qui ont fourni les meilleurs résultats et nous regarderons ensuite la caractéristique de ces métriques au point de vue des poids du tableau 7.1 et 7.2. Si nous trouvons une corrélation, nous pourrions ainsi « prédire » quelles métriques seront les plus favorables en utilisant uniquement les valeurs des tableaux 7.1 et 7.2 qui, rappelons-le, se trouvent disponibles lors de l'exécution, contrairement à l'évaluation de la qualité du résumé qui elle, n'est connue qu'à la suite d'une analyse humaine.

Pour établir la corrélation, définissons maintenant ce qui constitue une métrique favorable pour un document. Les étapes suivantes ont été suivies afin de quantifier l'importance des métriques :

- Pour chacun des textes, nous classons chacune des combinaisons de métriques (127 différentes au total) par ordre de performance. Plus une combinaison nous donne une évaluation favorable (selon les évaluations manuelles), plus elle se situe dans un rang élevé de liste.
- Une fois la liste triée, nous normalisons les évaluations pour les situer entre 0 et 1. Donc, chacune des 127 combinaisons possède maintenant une valeur d'évaluation qui se situe dans cette intervalle.
- Afin de voir l'importance d'une seule métrique et non d'une combinaison de celles-ci, une somme est effectuée pour chacune des 7 métriques. Nous prenons une à une les 127 combinaisons et nous ajoutons la valeur normalisée de la combinaison traitée à la somme de chacune des métriques faisant partie de cette

même combinaison. Par exemple, si la combinaison de métrique $FEX$ possède une valeur normalisée de 0.792, le programme ajoute $F+ = 0.792$, $E+ = 0.792$ et $X+ = 0.792$.

- Pour finir et pour aider la comparaison des différents résultats, les valeurs de chacune des 7 métriques sont divisées par la valeur de la métrique la plus élevée.

En résumé, dans le tableau 7.3, plus une métrique se rapproche de la valeur 1, plus elle s'avère importante (se retrouve fréquemment) à la contribution des meilleures combinaisons de métriques.

Tableau 7.3 Fréquence d'apparition de chacune des métriques dans l'évaluation subjective de la qualité des résumés.

| Métriques | F | D | E | T | L | A | X |
|---|---|---|---|---|---|---|---|
| Durham | 0.9007 | 0.8911 | 0.9211 | 0.8921 | 0.8909 | 1.0000 | 0.8907 |
| Épicier | 0.9793 | 0.9885 | 0.9532 | 0.9526 | 1.0000 | 0.9188 | 0.8064 |
| Kanthume | 0.8971 | 0.9160 | 0.9561 | 0.9620 | 0.9549 | 0.9185 | 1.0000 |
| Football | 0.9764 | 0.9701 | 0.9654 | 0.9749 | 1.0000 | 0.8978 | 0.8664 |
| J'Accuse | 0.9298 | 1.0000 | 0.9790 | 0.9702 | 0.9645 | 0.8718 | 0.9593 |
| Univers | 0.9584 | 0.9766 | 1.0000 | 0.9941 | 0.9824 | 0.9317 | 0.8848 |
| Opus Dei | 0.9879 | 0.9625 | 0.9564 | 0.9580 | 0.9970 | 1.0000 | 0.9605 |
| Cybermédias | 0.8049 | 0.8567 | 0.8262 | 0.8018 | 0.7957 | 1.0000 | 0.9177 |
| Sciences | 0.9628 | 1.0000 | 0.9155 | 0.9204 | 0.9715 | 0.9435 | 0.7481 |
| Sirène | 0.9551 | 0.9549 | 0.9736 | 0.9669 | 1.0000 | 0.9370 | 0.9425 |
| Travail | 0.8113 | 0.8079 | 0.8328 | 0.8195 | 0.8079 | 0.7467 | 1.0000 |

### 7.1.1 Analyse selon les métriques

Les figures 7.3 à 7.9 illustrent les trois tableaux précédents. Les histogrammes représentent moyennes et écart-types, tandis que la courbe indique l'importance de

la métrique.

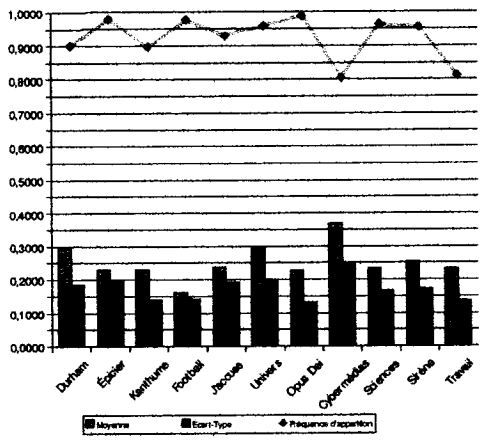


Figure 7.3 **Métrique F.** Évolution de la cote attribuée à la métrique F, en comparaison à la moyenne et à l'écart-type du poids attribué aux phrases.

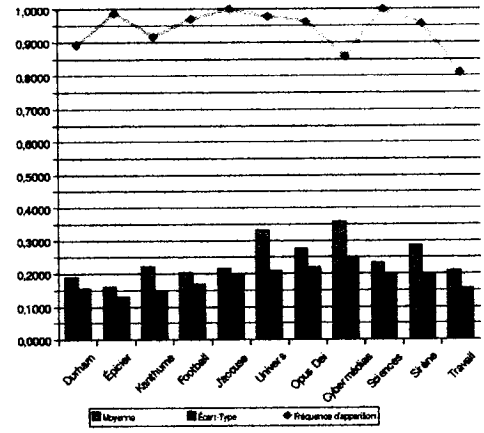


Figure 7.4 **Métrique D.** Évolution de la cote attribuée à la métrique D, en comparaison à la moyenne et à l'écart-type du poids attribué aux phrases.

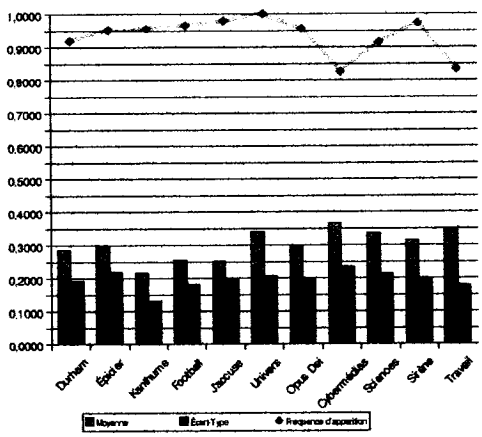


Figure 7.5 **Métrique E.** Évolution de la cote attribuée à la métrique E, en comparaison à la moyenne et à l'écart-type du poids attribué aux phrases.

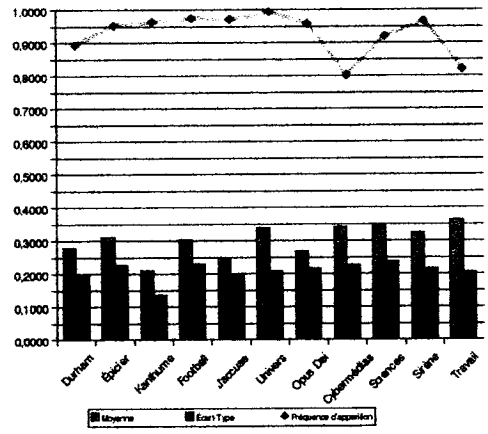


Figure 7.6 **Métrique T.** Évolution de la cote attribuée à la métrique T, en comparaison à la moyenne et à l'écart-type du poids attribué aux phrases.

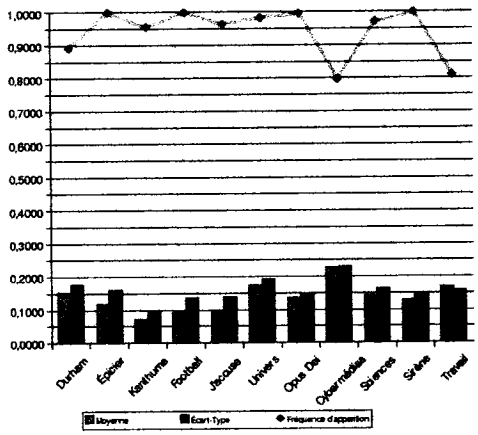


Figure 7.7 **Métrique L.** Évolution de la cote attribuée à la métrique L, en comparaison à la moyenne et à l'écart-type du poids attribué aux phrases.

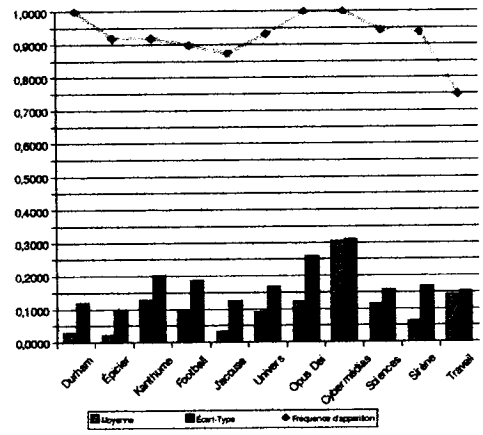


Figure 7.8 **Métrique A.** Évolution de la cote attribuée à la métrique A, en comparaison à la moyenne et à l'écart-type du poids attribué aux phrases.

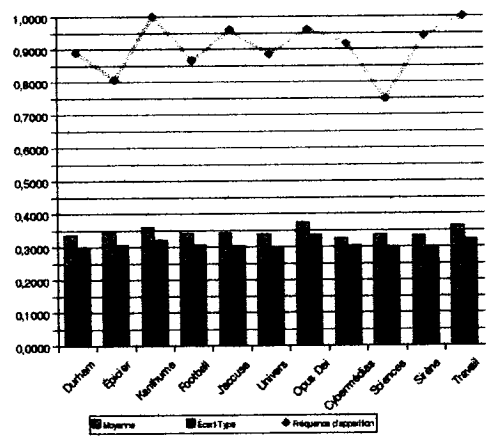


Figure 7.9 **Métrique X.** Évolution de la cote attribuée à la métrique X, en comparaison à la moyenne et à l'écart-type du poids attribué aux phrases.

Prenons tout d'abord la métrique F et regardons pour quels textes elle semble avoir été un bon choix. De toute évidence, pour les documents Opus Dei et Football, nous pouvons affirmer que la métrique a été utile (figure 7.3). Si nous regardons du côté de la moyenne des poids normalisés dans cette même figure, nous remarquons que ces deux mêmes textes ont obtenus les plus faibles résultats pour la métrique F. Essayons de comprendre la signification d'un faible résultat pour cette métrique dans le tableau 7.1 : chacune des phrases se voit attribuer un résultat par CORTEX. Pour la métrique de fréquence, ce résultat est incrémenté de 1 pour chaque mot contenu dans la matrice d'occurence des mots (équation 2.1, chapitre 2), par conséquent, plus une phrase est longue et qu'elle contient de termes (autres que des articles ou verbes de soutien), plus elle risque de posséder un résultat important. Lorsque toutes les notes ont été attribuées, une normalisation est effectuée et tous les résultats sont ramenés entre 0 et 1. Supposons maintenant un texte constitué de phrases uniformes, ayant toutes des longueurs similaires. Une fois normalisée, la moyenne des poids attribués à chacune des phrases se situera aux alentours de 0.5 puisque chacun des résultats aura été normalisé entre 0 et 1. Il va de soi que pour un tel texte, l'information sur la fréquence des mots n'est pas pertinente. Puisque chacune des phrases contient un nombre similaire de mots pertinents, il est inutile de tenter de choisir les phrases pertinentes sur cette base. Un raisonnement analogue peut être fait pour des moyennes supérieures ou inférieures à 0.5. En s'éloignant de cette valeur soit vers le haut ou vers le bas, nous pouvons en tirer certaines informations. Si nous nous approchons de la valeur 1, cela signifie que le texte possède plusieurs phrases, constituées chacune d'une multitude de mots importants ; versus quelques rares phrases n'en possédant que très peu. Si nous posons l'hypothèse que dans un tel texte, les phrases courtes apportent une faible quantité d'informations versus les phrases plus longues, nous pouvons affirmer que la métrique F s'avère utile pour éliminer les phrases courtes. À l'opposé, si la moyenne s'approche de 0, nous sommes vraisemblablement en présence d'un texte possédant plusieurs phrases courtes avec

de rares phrases longues, ces dernières prennent potentiellement beaucoup d'importance. Une autre hypothèse tout aussi probable propose simplement le fait que les résultats de la métrique F confirment la tendance des évaluateurs à sélectionner les phrases les plus longues.

Nous avons vu qu'il est possible d'analyser le comportement de la métrique F selon les critères de moyennes et d'écart-types des valeurs attribuées aux phrases. Malheureusement, il ne s'agit là que d'une exception. En effet, en observant les résultats obtenus pour les autres métriques, aucune relation évidente ne peut être tirée, mis à part le fait que certaines métriques semblent réagir de façon similaire d'un texte à l'autre. Si nous regardons l'évolution de la courbe des métriques E et T, nous pouvons poser l'hypothèse que ces deux métriques sont pertinentes dans les mêmes situations et devraient par conséquent, s'utiliser dans les mêmes circonstances.

## 7.2 Métrique différente pour chacune des phrases

L'autre technique expérimentée dans le but d'établir le choix des métriques vise pour objectif d'utiliser les métriques, non pas pour le texte en entier, mais pour chacune des phrases prises individuellement : chaque phrase possède sa propre métrique. L'intuition de cette méthode est venue lors de l'observation de la répartition des poids d'une phrase à l'autre. En effet, prenons en considération les sept métriques distinctes **FDETLAX** présentées à la section 5.2.2. Le fonctionnement de CORTEX fait en sorte que chacune des métriques est évaluée à la base indépendamment pour chacune des phrases. Nous discuterons un peu plus loin dans cette section du fait que les résulats donnés aux phrases tendent généralement à être relativement constants.

En remplaçant la technique expliquée à la section 2.1.3, qui permet de combiner les valeurs obtenues des différentes métriques, nous pourrions modifier la manière dont CORTEX traite les résultats provenant des métriques en conservant uniquement la note maximale donnée par la métrique la plus performante, et ce pour chacune des phrases du texte.

Par exemple, supposons le texte suivant de 4 phrases :

1. `Le chien, meilleur ami du monde, est mort, ce qui est un tourment pour l'humanité.`
2. `Vu sa grande notoriété, il a été placé en orbite terrestre.`
3. `Il est maintenant, à proprement parler, un satellite.`
4. `Toute l'humanité est en larmes, larmes, larmes,...`

Et considérons maintenant les métriques fictives **Q**, **R** et **S**, utilisant des critères quelconques (peu importants pour l'exemple), qui auraient évalué le précédent texte comme suit :

| Exemple de résultats d'évaluation fictive | | | |
|---|---|---|---|
| Phrases | Métrique **Q** | Métrique **R** | Métrique **S** |
| Phrase 1 | 0.6 | 0.2 | 0.4 |
| Phrase 2 | 0.5 | 0.1 | 0.4 |
| Phrase 3 | 0.6 | 0.8 | 0.3 |
| Phrase 4 | 0.6 | 0.2 | 0.5 |

Ainsi, au lieu de combiner ces résultats pour chacune des phrases par la méthode traditionnelle, l'idée est de ne conserver que le maximum obtenu pour ces mêmes phrases :

Phrase 1 : 0.6 (**Q**)

Phrase 2 : 0.5 (**Q**)

Phrase 3 : 0.8 (**R**)

Phrase 4 : 0.6 (**Q**)

Ensuite, nous ne retenons que les phrases ayant une forme de discontinuité avec les autres, c'est-à-dire celles qui sortent de la distribution régulière des résultats. En effet, la section II.1 de l'annexe nous rapporte un exemple de répartition pour le texte *Cybermédias* ; nous remarquons que les phrases réagissent à peu près toutes de la même manière aux métriques sauf quelques exceptions. L'idée est de ne retenir que ces exceptions ; ces phrases seront réputées contenir une certaine forme d'information différente des autres. Ainsi, selon l'exemple fictif précédent, la phrase numéro 3 aurait été retenue puisqu'elle est affectée par la métrique **R** contrairement à la tendance normale vers la métrique **Q**. De plus, le maximum obtenu se démarque lui aussi des autres phrases. Ce sont ces deux caractéristiques qui sont observées pour en arriver à retenir les phrases. Plus les caractéristiques sont différentes d'une phrase à l'autre, plus la phrase se verra attribuer de l'importance. Selon ce degré d'importance et le nombre de phrases que l'utilisateur désire retenir, CORTEX effectue sa sélection.

### 7.2.1 Comment discriminer les phrases

Lorsque nous observons les résultats réels, par exemple ceux de la section II.1 de l'annexe, nous voyons une forte relation entre la discontinuité dont nous avons précédemment discuté et les phrases retenues par les évaluations humaines (tableau I.1 en annexe). Observons en exemple les phrases #8 et #13 à la figure 7.10, toutes deux retenues par chacun des évaluateurs humains. Nous remarquons ce type de distribution :

...
Phrase #7, Valeur Maximale = 0.42, Métrique associee = F
Phrase #8, Valeur Maximale = 0.64, Métrique associee = A
Phrase #9, Valeur Maximale = 0.25, Métrique associee = X
...
Phrase #12, Valeur Maximale = 0.17, Métrique associee = F
Phrase #13, Valeur Maximale = 0.62, Métrique associee = D
Phrase #14, Valeur Maximale = 0.25, Métrique associee = F
...

Figure 7.10 Données tirées de l'annexe II.1 illustrant un exemple de distribution des valeurs.

Comment serait-il possible de modéliser cette observation pour être en mesure d'en tirer profit dans le logiciel ? Premièrement, rappelons-nous que les caractéristiques pour une phrase donnée sont comparées à la phrase précédente et la suivante. Que faire, donc, avec les première et dernière phrases du texte ? La solution retenue pour ces deux cas particuliers consiste à les retenir systématiquement, puisque la métrique X (section 5.1) assigne la valeur maximale de 1.0 à ces deux phrases. Il sera d'ailleurs considéré que si une phrase se voit attribuer une valeur très élevée (au delà de 0.95), elle sera automatiquement intégrée au résumé, peu importe les phrases précédentes et suivantes : ces cas se trouvent relativement peu fréquents. La majeure partie du temps, chacune des métriques attribue la valeur 1 à au moins une seule phrase (deux pour la métrique X) et les autres phrases du document se situent sous la barre des 0.95.

Maintenant que nous avons discuté des cas particuliers, nous verrons maintenant comment il s'avère possible de soutirer l'information sur la discontinuité pour les phrases restantes.

#### 7.2.1.1 Comment définir la discontinuité

Quatre données sont observées pour définir la discontinuité des résultats. Concrètement, seulement deux types distincts d'observations font partie de l'heuristique utilisée. La première : la séquence de métrique. CORTEX note pour chaque phrase si la précédente est marquée de la même métrique ; *idem* pour la métrique affectée à la phrase suivante. Nous obtenons ainsi deux données à observer. Ensuite, la différence entre la valeur maximale attribuée à la phrase courante et à la phrase qui précède, et entre la phrase courante et celle qui suit. Si la valeur de la phrase courante se trouve significativement supérieure à ses voisines, il est fort probable qu'elle contienne une information différente. Ces quatres données seront donc utilisées dans une heuristique qui permettra de discriminer des phrases possédant moins d'attributs.

Il existe une multitude de méthodes pour combiner ces caractéristiques. Comme toute heuristique, il s'agit d'un processus d'essais-erreurs qui suit une certaine logique selon les observations, sans pour autant avoir de fondement mathématique ou logique mieux établi. Nous verrons ici les résultats provenant des tests les plus probants, les autres résultats intermédiaires pouvant être consultés à l'annexe II.

Tout d'abord, il nous faut déterminer quel seuil doit être minimalement requis entre les résultats de deux phrases successives, pour affirmer que l'une d'elles possède un résultat significativement supérieur. Pour les tests initiaux, une valeur de 0.2 d'écart a arbitrairement été posée. À partir de là, nous possédons une méthode pour évaluer les quatre caractéristiques identifiées précédemment, il nous faut donc ensuite trouver une méthode pour les combiner. Nous pouvons aisément admettre qu'une phrase possédant les quatres caractéristiques simultanément (elle utilise une métrique différente de la phrase précédente et la phrase suivante, en plus d'avoir une valeur supérieure à celles-ci), admet une plus grande probabilité de succès qu'une

autre n'en possédant que deux ou trois. Nous établirons donc une sorte de hiérarchie dans CORTEX pour lui permettre de choisir parmi plusieurs phrases, advenant le cas où il devrait restreindre ses choix pour compresser le document. Ainsi, en ordre de priorité nous retrouverons les phrases :

1. situées en début et en fin de document
2. possédant les quatre caractéristiques
3. ayant au minimum les deux écarts requis avec les phrases voisines
4. constituées de trois des quatres caractéristiques
5. affectées par une métrique différente des deux phrases voisines
6. constituées de deux des quatres caractéristiques
7. restantes, classées par ordre de résultats

Maintenant que nous avons défini ces règles, nous pouvons récolter les premiers résultats de CORTEX avec les métriques automatiques. Le tableau II.1 en annexe montre des résultats plutôt décevants. Généralement plus hauts que les mesures de base, les résultats ne sont pas comparables à ceux de CORTEX utilisant les métriques traditionnelles. Dans les prochaines sections, nous essayerons de trouver un moyen d'améliorer ces performances. Le tableau 7.4 de la prochaine section résumera les résultats que nous retrouvons aux tableaux II.1 à II.7.

### 7.2.2 Améliorations à l'heuristique initiale

Les résultats préliminaires sont en soi surprenants compte tenu du fait qu'à observer les résultats et tenter l'expérience approximativement tout en suivant les critères établis, la méthode nous semble efficace. Essayons d'ajouter un nouveau critère à ceux précédemment mentionnés. Peut-être serait-il intéressant de bonifier davantage les phrases se démarquant d'une manière significative. En effet, en observant les

résultats, nous remarquons que plusieurs phrases correspondent aux critères mentionnés. En réalité, trop de phrases correspondent, nous pourrions donc ajouter un critère additionnel (ajoutant ainsi deux caractéristiques : phrases suivantes et précédentes) qui considère en plus du seuil minimal, les phrases qui franchissent le seuil d'une manière significative. Plus concrètement, essayons de doubler le seuil en deux nouvelles caractéristiques et insérons ce nouvel élément entre les priorités numéros 1 et 2 de la section précédente. Ainsi, si la phrase précédente et la phrase suivante sont distantes du double du seuil minimal, la phrase courante sera insérée entre les priorités 1 et 2. Le tableau II.2 en annexe montre les résultats ainsi obtenus : toujours décevants en comparaison aux résultats originaux, mais encourageants dans le sens où nous améliorons ceux établis automatiquement précédemment.

Pourquoi alors ne pas regarder le comportement en augmentant le facteur multiplicatif? Le tableau II.3 en annexe nous rappporte les résultats en considérant un facteur de 2.5x au lieu de 2x pour le nouveau critère introduit. Malheureusement, les résultats se stabilisent ou même se dégradent dans plusieurs cas. Signe que la technique se limite ici, nous devrons trouver autre chose pour en arriver à des résultats concluants...

Laissons tomber le nouveau critère introduit dans cette section et revenons à la hiérarchie décrite en 7.2.1.1. Le seuil de 0.2 pour désigner l'écart minimal des résultats entre deux phrases n'est fort probablement pas optimal. Essayons de déplacer ce dernier. Pourquoi ne pas essayer de diminuer la valeur puisque le seuil établi à 0.2 semble restreindre plusieurs phrases? Effectuons les tests avec un seuil minimal placé à 0.1, les résultats sont présentés dans le tableau II.4 en annexe : nous n'arrivons pas encore à des résultats intéressants. Voyons ce qui se produit en réintroduisant les dernières caractéristiques que nous venons de laisser tomber, c'est-à-dire tenir compte en plus du seuil minimum, d'un nouveau critère qui établit un nouveau seuil par un facteur multiplicatif sur le seuil minimum. Encore une fois,

diverses valeurs peuvent y être multipliées. Cette fois-ci, les tableaux II.5 et II.6 en annexe, illustrant respectivement des facteurs de 2x et 3x, démontrent une nette amélioration. Cependant, nous verrons dans la prochaine section que c'est avec un facteur multiplicatif de 4x que nous obtenons les résultats les plus intéressants, dépassée cette valeur (tableau II.7 en annexe), les résultats se dégradent. Le tableau 7.4 qui suit résume les résultats que nous venons de décrire.

Tableau 7.4 Résumé des résultats obtenus avec les tests effectuées dont les valeurs se retrouvent aux tableaux II.1 à II.7

| Test effectué | Nombre de textes améliorés | Nombre de textes déteriorés | Moyenne avant | Moyenne après |
|---|---|---|---|---|
| Écart 0.2 | 1 | 10 | 43 | 32 |
| Écart 0.2 et 0.4 | 1 | 10 | 43 | 34 |
| Écart 0.2 et 0.5 | 1 | 10 | 43 | 33 |
| Écart 0.1 | 1 | 10 | 43 | 32 |
| Écart 0.1 et 0.2 | 1 | 10 | 43 | 31 |
| Écart 0.1 et 0.3 | 1 | 10 | 43 | 33 |
| Écart 0.1 et 0.5 | 1 | 10 | 43 | 34 |
| 77 textes (7 tests x 11 documents) | 7 | 70 | 43 | 33 |

### 7.2.3 Résultats finaux et analyse

Le tableau 7.5 rapporte maintenant les résultats obtenus à l'aide de cette nouvelle heuristique, selon les critères élaborés dans les sections précédentes, en comparaison aux différentes mesures de base.

Tableau 7.5 Choix automatique des métriques en comparaison aux méthodes de base

| | Métriques auto. (moyenne des évaluateurs A B C) | Phrases longues | Premières et dernières phrases | Hasard |
|---|---|---|---|---|
| Durham | 35 % | 44 % | 27 % | 20 % |
| Épicier | 37 % | 50 % | 32 % | 23 % |
| Kanthume | 33 % | 39 % | 32 % | 20 % |
| Football | 52 % | 42 % | 27 % | 25 % |
| J'Accuse | 30 % | 33 % | 11 % | 13 % |
| Univers | 43 % | 43 % | 28 % | 18 % |
| Opus Dei | 35 % | 44 % | 35 % | 22 % |
| Cybermédias | 38 % | 43 % | 33 % | 26 % |
| Sciences | 33 % | 39 % | 29 % | 22 % |
| Sirène | 28 % | 40 % | 30 % | 23 % |
| Travail | 23 % | 21 % | 21 % | 18 % |
| Moyenne | 35 % | 40 % | 28 % | 21 % |

Nous remarquons que la technique offre des résultats inférieurs à ceux produits par la sélection des phrases les plus longues, mais s'avère tout de même supérieure aux

deux autres mesures de base. Bien que diverses tentatives aient été effectuées en vue d'améliorer l'heuristique utilisée, il est hautement probable que l'ajout de certaines informations additionnelles à celle-ci permettrait d'améliorer les résultats davantage, au point d'en dépasser toutes combinaisons statiques. En effet, l'heuristique ajoute l'opportunité de ne conserver que les métriques ayant des comportements particuliers sur le texte traité, ce que ne permet pas un choix de métrique fixe. Le choix automatique de métriques pourrait par conséquent, si l'heuristique était améliorée, permettre de s'adapter beaucoup mieux aux différents genres de documents traités. Des tests plus approfondis pourraient être effectués sur ce sujet, nous en discuterons davantage à la conclusion de ce mémoire dans la section des travaux futurs.

# CHAPITRE 8

# UTILISATION D'UNE GRAMMAIRE DE LA LANGUE FRANÇAISE

Jusqu'à maintenant, des méthodes purement statistiques ont été utilisées pour faire fonctionner CORTEX et obtenir des résumés. Dans ce chapitre, nous verrons comment l'ajout d'un module de pré-traitement linguistique permet d'améliorer les performances de CORTEX. Ce module réalise une analyse grammaticale de la phrase, ce qui a pour avantage de fournir une information supplémentaire au logiciel quant à la structure de la phrase, à savoir où se trouvent les verbes, les sujets, les compléments du nom, etc. De plus, ces informations pourront être utilisées pour effectuer un pré-filtrage en éliminant certaines parties de phrases moins importantes. De cette manière, nous pourrons accorder plus de poids aux termes importants, tout en en supprimant d'autres de moindre importance, tels que des circonstanciels de lieu ou de temps par exemple.

## 8.1 Analyseur syntaxique

L'analyseur syntaxique utilisé est celui qui se trouve à la base du logiciel « Le Correcteur 101™ ». La sortie obtenue à l'aide de cet analyseur est représentée sous la forme d'arbre. Le programme prend un texte brut en entrée et fournit en sortie les phrases analysées sous forme d'arbre, où chacun des noeuds de l'arbre représente un mot du texte associé à sa catégorie (Nom, Verbe, Adjectif, ...), et où chacun des liens entre les noeuds est formé de la relation qui associe ceux-ci dans la phrase (Complément, Déterminant, Sujet, ...). Voici un exemple d'analyse, légèrement

simplifié, tiré de la seconde partie de la phrase # 59 du texte *Cybermédias* (contenu intégral, annexe I.1)

```
Les médias sont-ils responsables
de l'efficacité des publicités
qu'ils véhiculent ?

Arbre sont/verbe
 Sujet les médias/nom
 RepriseSujet ils/pronPers
 Attrib responsables/adj
  ComplAdj de l'efficacité/nom
   ComplNom des publicités/nom
    Relat véhiculent/verbe
     ObjetDirect qu'/pronRelat
     Sujet ils/pronPers
 FinProp ?/ponctFinale
```

Figure 8.1 Exemple d'arbre de dépendance produit par le module d'analyse syntaxique. Quelques sous-arbres sont simplifiés.

Pour ce qui est de l'analyse en elle-même, elle ne s'effectue pas en une seule itération sur le texte en entier. L'analyseur produit plutôt un découpage sur le texte qui lui est fourni. Règle générale, les phrases sont traitées indépendamment les unes des autres et nous obtenons ainsi un arbre distinct pour chacune des phrases du document. Il existe cependant une exception non négligeable à cette règle : un choix de conception de 101 fait intervenir un découpage plus fin dans certaines situations. Par exemple, si la phrase traitée contient des points-virgules ou des deux-points, elle sera scindée en plusieurs petits morceaux. Nous obtenons alors, pour chaque segment de la phrase, une analyse et un arbre distincts. Cette segmentation des phrases peut poser problème. En effet, si l'analyseur n'effectue pas la même division que celle initialement réalisée par CORTEX, les phrases ne concorderont plus lorsque nous comparerons deux résultats. Nous verrons comment palier ce problème à la

section 8.2.

## 8.2 Découpage en sections

Dans le souci de conserver la même segmentation et la même numérotation des phrases, les textes segmentés en XML (section 2.1.1) ont été insérés dans CORTEX via un script qui lance l'analyse de chacune des phrases une à une. Il s'avère nécessaire d'utiliser le texte dans son format XML (post-segmentation) pour s'assurer que la numérotation appliquée aux phrases traitées demeure la même. En effet, il a été précédemment discuté du fait que l'analyseur grammatical effectue un découpage beaucoup plus fin pour ses analyses que l'utilisation d'une phrase entière, par conséquent il apparaît nécessaire de marquer la division des phrases pour CORTEX avant de les introduire dans l'analyseur. Par cette astuce, bien qu'une phrase unique puisse se retrouver divisée, toutes les sections éclatées pourront à nouveau être regroupées à la suite de l'analyse.

## 8.3 Fusion des expressions de plusieurs mots

Nous retrouvons très régulièrement, dans un document, une série de mots utilisés conjointement pour représenter une expression, un nom, un lieu, etc. Par exemple, « John Doe », « à jamais », « pour toujours », « World Wide Web », « École Polytechnique », « Canadien Français », ..., sont tous des regroupements de mots formant une expression. L'analyse syntaxique offerte par 101 se trouve en mesure de reconnaître ces expressions. À partir de cette information, les mots sont reliés par le caractère '_' et sont donnés sous cette forme à CORTEX. La section 2.1.1 faisait mention du fait que le pré-traitement de Cortex est en mesure de réaliser cette opération, cependant celle-ci est réalisée via une banque de mots connus.

L'avantage ici se trouve dans le fait qu'il n'y a nul besoin de connaître toutes les expressions pour être en mesure de les remplacer. En effet, dans les exemples précédents, un nom propre tel « John Doe » ne se retrouve vraisemblablement pas dans une banque d'expressions, alors que l'analyseur est tout de même en mesure de le reconnaître. Cela évite que CORTEX interprète et enregistre distinctement chacun des mots de l'expression dans sa matrice de fréquences. Notons que pour le texte *Cybermédias*, 37 expressions ont été reconnues, alors que CORTEX, sans ce module, n'en repère que 20. En plus, nous avons constaté à la figure 2.3 de la page 41 que lors de l'étape de la fusion des mots composés, l'utilisation d'un dictionnaire pour la reconnaissance d'expressions a entraîné la détection d'un faux positif. En effet, le dictionnaire d'expression contenait le mot « bien-être » et ce mot fut reconnu, alors que les mots « bien » et « être » étaient utilisés dans un tout autre contexte. Les faux positifs sont beaucoup plus fréquents en utilisant une simple banque de mots plutôt qu'une analyse de la phrase. Voyons maintenant le tableau 8.1, qui nous rapporte les résultats provenant de l'utilisation de CORTEX avec et sans l'utilisation de ce module.

Tableau 8.1 Comparaison de résultats démontrant l'efficacité de la fusion des expressions par l'analyseur

| | ETX (%) | | | | ETX+Lemm | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 59 | 55 | 18 | 44 |
| Épicier | 60 | 50 | 40 | **50** | 55 | 55 | 45 | 52 |
| Kanthume | 46 | 58 | 25 | **43** | 46 | 58 | 29 | 44 |
| Football | 73 | 36 | 55 | **55** | 82 | 45 | 73 | 67 |
| J'Accuse | 67 | 33 | 29 | **43** | 73 | 36 | 55 | 55 |
| Univers | 43 | 50 | 50 | **48** | 36 | 29 | 29 | 31 |
| Opus Dei | 36 | 40 | 44 | **40** | 40 | 40 | 49 | 43 |
| Cybermédias | 14 | 57 | 14 | **28** | 14 | 57 | 29 | 33 |
| Sciences | 39 | 52 | 52 | **48** | 43 | 48 | 43 | 45 |
| Sirène | 63 | 40 | 51 | **51** | 31 | 31 | 34 | 32 |
| Travail | 36 | 16 | 28 | **27** | 40 | 20 | 32 | 31 |
| Moyenne | | | | 43 | | | | 43 |

Le tableau 8.1 montre clairement un net avantage de la majorité des documents face à la reconnaissance des expressions. Malgré tout, la moyenne globale de tous les textes n'a pas évolué, mais nous remarquons aisément que deux d'entre eux : *Univers* et *Sirène* sont responsables de cette stabilité. À eux seuls les résultats obtenus par ces textes compensent le gain établi par tous les autres. Nous pouvons tout de même conclure à l'efficacité de la méthode puisqu'une grande proportion des documents en tire profit.

## 8.4 Filtrage

Nous avons discuté à la section 8.1 de la manière dont les arbres sont construits par 101. Comment pourrions-nous profiter de ces structures pour améliorer les résumés produits par CORTEX ? L'idée de base expérimentée est de réduire les arbres au maximum dans le but de ne conserver que les informations essentielles. Ainsi, pour une phrase donnée, toutes les relations jugées non-essentielles à son sens et à sa compréhension sont retirées. Nous avons expérimenté deux types de filtrages qui seront détaillées dans les sections à venir.

### 8.4.1 Filtrage simple

En premier lieu, nous nous sommes demandé : qu'advient-il si nous filtrons un minimum d'information ? Pour ces tests, nous avons conservé toutes les catégories de mots et seules quelques relations sont filtrées : subordonnées, appositions, incises et circonstantielles. Puisqu'il s'agit d'un arbre, lorsqu'une de ces relations est rencontrée, tout le sous-arbre est supprimé. La figure 8.2 illustre le résultat du filtrage sur la phrase #2 du texte *Cybermédia* (annexe I.1). Nous y retrouvons trois sections de phrases filtrées : un adverbe jouant le rôle d'une circonstantielle, une locution adverbiale de lieu (circonstancielle) et finalement, une apposition.

{Pourtant,}$_{CircAdv}$ {dans le monde en pleine effervescence d'Internet, }$_{LocAdv}$ l'arrivée de HotWired marque le début de la cybermédiatisation { , le premier véritable média sur Internet}$_{App}$.

*devient après filtrage :*

L'arrivée de HotWired marque le début de la cybermédiatisation.

Figure 8.2 Exemple de filtrage d'apposition et de circonstancielles.

### 8.4.2 Filtrage agressif

Une autre technique en remplacement à un filtrage simple pourrait se trouver intéressante. Au lieu de filtrer les quelques relations décrites à la section 8.4.1, essayons d'élargir notre filtre. De cette façon, les liens suivants ont été supprimés : verbes auxiliaires, toutes relations circonstantielles, compléments d'adverbes, dates, lieux, appositions, subordonnées, épithètes et locutions de tous genres. Encore une fois, lorsqu'une relation rencontrée doit être éliminée, tout le sous-arbre est supprimé. De plus, outre les relations, nous retrouvons certaines catégories de mots qui ont aussi été filtrées. Parmi celles-ci nous notons toutes les formes de pronoms et d'adverbes détectés.

### 8.4.3 Résultats des filtrages

Pour prouver ou infirmer l'efficacité des techniques proposées, les tableaux 8.2 et 8.3 présentent les résultats de CORTEX utilisant sa combinaison de métrique la plus performante telle que décrite au chapitre 5, en opposition à la même combinaison appliquée au document filtré ; les autres combinaisons décrites au chapitre 5 se

comportent de manière similaire. Les notes sont présentées pour les 3 évaluateurs A, B et C et la moyenne obtenue pour chacun des textes est comparée. Le tableau suivant expose les résultats de l'application d'un filtrage simple.

Tableau 8.2 Tableau comparatif des résultats avec l'utilisation d'un filtrage

| | ETX (%) | | | | ETX+Filtre simple (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 59 | 55 | 9 | 41 |
| Épicier | 60 | 50 | 40 | **50** | 45 | 50 | 30 | 42 |
| Kanthume | 46 | 58 | 25 | **43** | 50 | 58 | 29 | 46 |
| Football | 73 | 36 | 55 | **55** | 82 | 36 | 55 | 58 |
| J'Accuse | 67 | 33 | 29 | **43** | 67 | 33 | 33 | 44 |
| Univers | 43 | 50 | 50 | **48** | 57 | 29 | 21 | 36 |
| Opus Dei | 36 | 40 | 44 | **40** | 33 | 49 | 44 | 42 |
| Cybermédias | 14 | 57 | 14 | **28** | 43 | 57 | 14 | 38 |
| Sciences | 39 | 52 | 52 | **47** | 48 | 61 | 48 | 52 |
| Sirène | 63 | 40 | 51 | **51** | 29 | 37 | 40 | 35 |
| Travail | 36 | 16 | 28 | **27** | 32 | 16 | 44 | 30 |
| Moyenne | | | | **43** | | | | 42 |

Le tableau 8.2 affiche une amélioration sur sept des onze documents alors que seuls trois d'entre eux se trouvent déficitaires. Nous pouvons ainsi conclure que filtrer les informations moins porteuses de sens au sein d'une phrase constitue une technique fort appréciable, cependant il ne faut pas ignorer la faible diminution de la moyenne totale de tous les textes, déficitaire d'un point. Voyons avec le tableau 8.3, si un filtrage plus agressif nous donne de meilleurs résultats.

Tableau 8.3 Tableau comparatif des résultats avec l'utilisation d'un filtrage agressif

| | ETX (%) | | | | ETX+Filtre agressif(%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 41 | 36 | 9 | 29 |
| Épicier | 60 | 50 | 40 | **50** | 45 | 55 | 30 | 43 |
| Kanthume | 46 | 58 | 25 | **43** | 42 | 58 | 33 | 44 |
| Football | 73 | 36 | 55 | **55** | 82 | 45 | 73 | 67 |
| J'Accuse | 67 | 33 | 29 | **43** | 57 | 33 | 24 | 38 |
| Univers | 43 | 50 | 50 | **48** | 36 | 29 | 43 | 36 |
| Opus Dei | 36 | 40 | 44 | **40** | 38 | 56 | 38 | 44 |
| Cybermédias | 14 | 57 | 14 | **28** | 43 | 57 | 29 | 43 |
| Sciences | 39 | 52 | 52 | **47** | 39 | 43 | 48 | 43 |
| Sirène | 63 | 40 | 51 | **51** | 46 | 49 | 51 | 49 |
| Travail | 36 | 16 | 28 | **27** | 40 | 24 | 44 | 36 |
| Moyenne | | | | **43** | | | | 42 |

Nous remarquons dans le tableau 8.3 que cinq textes sur les onze réagissent de manière intéressante, par contre, les six autres subissent en moyenne une perte de valeur. Ceci signifie que les résumés de ces six textes se révèlent être de qualité inférieure aux résumés des documents sans filtrage. Nous ne pouvons donc pas conclure aux bienfaits d'un filtrage imposant, puisque les résultats sont ambivalents. De plus, le filtrage plus simple semblait nous fournir des résultats équivalents, voire meilleurs. Il semble ainsi peu pertinent d'appliquer systématiquement un filtrage syntaxique, puisque les résultats qui en découlent s'avèrent incertains. Nous devons aussi considérer que le filtrage tel qu'utilisé comporte certaines limites. Bien que l'analyseur soit très robuste puisqu'il est utilisé dans la correction d'orthographe et de grammaire de documents, certains points doivent être considérés. Quelques détails linguistiques sont ignorés dans l'analyseur lorsqu'ils n'ont aucun effet sur

la correction. Par exemple, 101 marque toutes les prépositions qui sont placées en tant que compléments du verbe tels les lieux, le temps, etc. de la même relation circonstancielle. Cette décision a été prise lors du développement de la grammaire dans le but de réduire le nombre d'analyses possibles, sachant que cette ambiguïté n'influencerait en rien la correction de la phrase.

Il n'est pas non plus exclu que l'une des raisons expliquant les performances mitigées du filtrage provienne du fait que nous éliminons ainsi une grande quantité de mots et nous perdons vraisemblablement de l'information statistique. En effet, pour que les techniques statistiques soient intéressantes, il s'avère nécessaire de posséder le plus d'informations possibles. Il s'agit donc de faire un compromis entre informations pertinentes et accumulation de mots pour les métriques statistiques.

## 8.5 Bonus pour certaines relations ou catégories de mots

L'analyse syntaxique retourne plusieurs informations intéressantes sur une phrase donnée. Jusqu'à maintenant, nous avons utilisé ces informations afin d'établir un filtre pour éliminer certains mots. Serait-il aussi avantageux, au lieu d'éliminer certaines relations et conséquemment certains passages, d'attribuer un bonus pour des relations ou des catégories de mots plus significatives au sein d'une phrase ? Par exemple, attribuer une importance plus élevée aux sujets des verbes ou encore aux verbes principaux ? La section 8.4.1 montrait une amélioration en utilisant un filtrage, qu'en est-il d'un filtrage combiné aux bonifications ?

### 8.5.1 Résultats de la bonification des relations et catégories de mots

Pour réaliser les tests, les mots bonifiés ont été dupliqués à l'intérieur des phrases. Par exemple, pour avantager les sujets des verbes, chacune des relations identifiées voyait la fréquence de son mot associé multipliée par deux. Pour les verbes, la même astuce est utilisée ; cependant, seuls les verbes principaux ont été dupliqués et ce pour une fréquence multipliée par trois. La figure 8.3 montre un exemple de phrase où aucun mot n'est filtré et où nous allouons un bonus de 2 pour les sujets et de 3 pour les verbes principaux.

Pourtant, dans le monde en pleine effervescence d'Internet, l'*arrivée* de HotWired *marque* le début de la cybermédiatisation, le premier véritable média sur Internet.

*devient après bonification :*

Pourtant, dans le monde en pleine effervescence d'Internet, l'*arrivée arrivée* de HotWired *marque marque marque* le début de la cybermédiatisation, le premier véritable média sur Internet.

Figure 8.3 Exemple de bonification des verbes principaux et des sujets.

Les tableaux des divers tests réalisés de trouvent en annexe III, ils ne sont pas reportés ici puisque les résultats obtenus ne sont pas concluants. Cependant, le tableau 8.4 résume les échecs constatés sur l'ensemble des textes. Que ce soit l'application d'un bonus sur les verbes sans effectuer de filtrage (tableau III.1), avec un filtrage (tableau III.2), sur les sujets et les verbes avec ou sans filtrage (tableaux III.3 et III.4) ou encore sur les sujets du verbe avec filtrage (tableau III.5), les résultats se ressemblent. Dans aucun des cas, nous arrivons à obtenir des résultats supérieurs ni même équivalents à ceux établis à la section 8.4.1 par le filtrage simple à lui seul.

Tableau 8.4 Résumé des résultats obtenus avec les tests effectuées dont les valeurs se retrouvent aux tableaux III.1 à III.5

| Test effectué | Nombre de textes améliorés | Nombre de textes déteriorés | Moyenne avant | Moyenne après |
|---|---|---|---|---|
| Bonus verbes | 4 | 6 | 43 | 39 |
| Filtre et Bonus verbes | 6 | 4 | 43 | 42 |
| Bonus verbes + suj. | 4 | 5 | 43 | 40 |
| Filtre et Bonus verbes + suj. | 5 | 5 | 43 | 41 |
| Filtre et Bonus suj. | 6 | 5 | 43 | 42 |
| 55 textes (5 tests x 11 documents) | 25 | 25 | 43 | 41 |

## 8.6 Conclusion sur l'analyse grammaticale

Il a été démontré que le filtrage d'informations, s'il est employé modérément, peut avantager CORTEX. Cependant, lorsque nous filtrons trop, les métriques statistiques

perdent de leur efficacité puisque les données se font plus rares et proportionnellement, le bruit plus important. D'un autre côté, le présent chapitre souligne le fait qu'il s'avère inutile, voir même pire pour l'obtention d'un résumé de qualité, de bonifier les sujets ou les verbes principaux d'une phrase. En effet, dans le meilleur des cas, nous ne faisons qu'avantager certains textes au détriment des autres et dans le pire des cas, la majorité des documents traités produit de moins bons résultats.

De plus, voici une information non négligeable à prendre en considération : l'analyse utilisée dans le présent chapitre est une opération très coûteuse en frais de temps. En effet, le temps d'exécution normal de CORTEX, incluant le pré-traitement et les calculs statistiques qui s'effectuent en moins de 10 secondes. Si nous comparons ce temps avec celui de l'analyse syntaxique, nous nous retrouvons loin derrière avec un temps non négligeable de 10 à 15 minutes en moyenne, pour les textes faisant l'objet de ce mémoire. Il s'agit là d'un temps relativement élevé que nous nous devons de considérer si le temps de calcul constitue un critère pour l'usager. Dans le cas contraire, nous pourrions affirmer qu'effectuer un filtrage simple avant même l'analyse des phrases par CORTEX peut se montrer fort utile.

# CONCLUSION

Dans ce mémoire, nous avons étudié diverses méthodes en vue de l'amélioration des résultats produits par CORTEX. Nous avons d'abord apporté de légères améliorations à ce qui existait déjà, en ajoutant une nouvelle métrique basée sur le positionnement des phrases dans le document, et en étudiant l'utilisation de diverses combinaisons de métriques afin de trouver la combinaison idéale. Par la suite, nous avons observé l'influence de la substitution des pronoms sur la sélection des phrases effectuée par CORTEX, la possibilité de choisir automatiquement les métriques idéales et finalement l'incorporation d'une analyse syntaxique.

Pour en arriver à exécuter ces tests, nous avons dû modifier la structure existante du code de CORTEX afin de la rendre plus lisible et facilement modifiable. Nous sommes passés d'un logiciel difficilement compréhensible écrit en langage C, à un système beaucoup plus complet et aisément configurable, écrit en C++. Profitant de techniques de programmation telles le polymorphisme et l'héritage, les ajouts au code source de CORTEX sont facilités. De plus, nous avons ajouté plusieurs modules afin de pouvoir interfacer des composantes externes et ainsi réaliser maintes expérimentations.

Nous pouvons constater après ces divers tests que malgré le fait que quelques légers progrès ont été réalisés, les méthodes statistiques utilisées dans CORTEX sont poussées à leurs limites. Sans l'utilisation de nouvelles données provenant de sources plus intelligentes telles que la sémantique des phrases ou certaines bases de connaissances externes (ontologies, etc.), nous pouvons difficilement effectuer de gains majeurs.

Malgré le fait que nous ne pouvons affirmer révolutionner CORTEX avec les tests effectués, nous avons tout de même pu établir certaines bases de fonctionnement

intéressantes. Dorénavant, il peut être exécuté avec une série de métriques que nous savons adéquates dans la plupart des situations, contrairement aux anciennes données récoltées par CORTEX, qui employait alors toutes les métriques simultanément. De plus, une métrique additionnelle basée sur la position des phrases dans le texte s'avère bien souvent pertinente dans le traitement de plusieurs documents.

Nous avons par la suite constaté que la substitution des pronoms des phrases par les entités qu'ils représentent n'aide pas la sélection des phrases réalisée automatiquement. Cependant, les expériences ont démontré que cette substitution demeure tout de même préférable à l'insertion de phrases additionnelles. Dépendamment du contexte d'utilisation, si les pronoms nous embêtent, il vaut mieux se tourner vers l'utilisation de la substitution des pronoms pour ne pas affecter la sélection automatique de phrases.

Quant à l'utilisation de certaines heuristiques pour établir le choix des métriques à utiliser, nous n'avons malheureusement pas pu conclure à l'efficacité des méthodes proposées. Cependant, l'heuristique implémentée dans CORTEX pourra être retravaillée aisément pour en arriver, peut-être, à des résultats concluant.

Nos derniers tests étaient consacrés à l'utilisation d'une grammaire de la langue française, dans le but d'établir une analyse syntaxique pour nous permettre de reconnaître les expressions fractionnées en plusieurs termes, les diverses catégories de mots, ainsi que les relations qui les relient. Nous n'avons malheureusement pas su tirer avantage d'un filtrage de mots ni de la bonification de relations jugées plus importantes en rapport avec la moyenne des résultats. Cependant, l'idée paraît prometteure si nous considérons le nombre de résumés améliorés. De plus, la reconnaissance d'expressions scindées en plusieurs mots, grâce au lemmatiseur, s'est avérée fort utile et ce, malgré le fait que CORTEX possédait déjà un module équivalent basé sur un dictionnaire. L'utilisation de l'analyseur syntaxique a permis

d'augmenter la qualité des sélections de phrases ; il s'agit là d'une preuve que l'analyseur apporte de précieuses informations grâce, entre autres, à la lemmatisation.

### Travaux futurs

Parmi les expériences futures intéressantes, nous pourrions mentionner l'utilité d'effectuer davantage de tests sur les heuristiques établies. Il existe très certainement une manière d'agencer les données pour permettre à CORTEX de sélectionner les métriques automatiquement suivant le texte traité. Plusieurs autres tests pourraient être tentés en modifiant les poids, les caractéristiques, etc.

Les expériences effectuées avec l'analyseur syntaxique pourraient elles aussi être étendues de diverses façons. Nous avons vu que le lemmatiseur à lui seul permet à CORTEX de gagner en performance, il serait intéressant de trouver certaines méthodes, possiblement autres que celles tentées dans ce mémoire, afin de tirer avantage de la connaissance des catégories des différents mots qui constituent le document ainsi que des relations qui les relient. L'outil serait de plus idéal pour utiliser des techniques similaires à celles employées dans le logiciel COLUMBIA NEWSBLASTER (section 1.2.2.5), soit d'utiliser les informations sur les couples sujet-verbes présents dans le document.

Notons aussi un perfectionnement possible de la métrique X, qui fonctionne actuellement en avantageant les phrases aux extrémités de sections (les sections étant séparées par les titres et sous-titres du document). Il pourrait s'avérer intéressant de voir les résultats obtenus en utilisant la métrique non pas avec les divisions des titres, mais plutôt avec la segmentation en paragraphes.

Une analyse demandant énormément d'expérimentations, mais qui pourrait s'avérer fort avantageuse, serait d'ajouter un poids aux onze métriques présentes dans le

logiciel. Plutôt que de juger si elles doivent ou non être présentes, il serait intéressant de leur ajouter un facteur multiplicatif permettant d'allouer une importance plus grande à certaines. Nous pourrions, par exemple, décider qu'une métrique donnée R doit être présente, mais seulement à 50%, alors qu'une autre, S, doit être utilisée à 90%. Ainsi, lors du passage dans l'algorithme de décision, ces poids influenceraient les valeurs finales. Une autre technique serait d'attribuer un poids important à la métrique F, dont nous avons observé les résultats élevés dans ce mémoire, pour lui ajouter graduellement les dix autres métriques à de plus faibles pourcentages afin d'arriver à des valeurs idéales. Cette technique pourrait aussi être réutilisée avec les heuristiques pour le choix automatique de métriques, ce qui permettrait une plus grande flexibilité pour l'élaboration d'une stratégie efficace.

L'algorithme de décision n'a pas fait l'étude de critiques dans ce mémoire. Il pourrait être intéressant d'y consacrer quelques expériences dans le but de voir, par exemple, s'il s'avère effectivement plus efficace qu'une simple moyenne. Nous pourrions aussi déterminer s'il n'existerait pas de techniques autres qui permettraient de combiner les différentes valeurs retournées par les métriques.

Finalement, notons quelques autres techniques qui pourraient s'adapter relativement simplement au fonctionnement actuel du logiciel. L'utilisation de la technique LSA (abordée à la section 1.2.2.2) ou encore plus simplement, l'utilisation des expressions indicatives pour ajouter un bonus à certaines phrases (tel que vu à la section 1.3.1.1) pourraient fort probablement permettre une amélioration aux résultats produits par CORTEX.

# RÉFÉRENCES

BREMDAL, B. A. (2000). Corporum summariser. Technical Report Document version 1.02, CognIT a.s. August 2000.

CHUANG, W. T. et YANG, J. (2000). Extracting sentence segments for text summarization : a machine learning approach. In *SIGIR '00 : Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, pages 152–159. ACM Press.

COPERNIC (2004). *Copernic Summarizer - Feature List.* `http://www.copernic.com/en/products/summarizer/features.html`.

DALIANIS, H. (2000). Swesum - a text summarizer for swedish. Technical Report TRITA-NA-P0015, IPLab-174, NADA, KTH, `http://www.nada.kth.se/~hercules/Textsumsummary.html`. October 2000.

EDMUNDSON, H. P. (1969). New methods in automatic extracting. *Journal of the ACM (JACM)*, **16**(2), 264–285. April 1969.

GAGNON, M. et SYLVA, L. D. (2005). Text summarization by sentence extraction and syntactic pruning. In *Proceedings of Computational Linguistics in the North-East.* Université du Québec en Outaouais, Gatineau, août 2005, 8 pages.

GE, N., HALE, J., et CHARNIAK, E. (1998). A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–171.

GROSZ, B. J., JOSHI, A. K., et WEINSTEIN, S. (1995). Centering : a framework for modeling the local coherence of discourse. *Computational Linguistics*, **21**(2), 203–226.

HASSEL, M. (2000). Pronominal resolution in automatic text summarization. Master's thesis, Stockholm University. June 2000.

HASSEL, M. (2003). Exploitation of Named Entities in Automatic Text Summarization for Swedish. In *Proceedings of NODALIDA'03 - 14th Nordic Conference on Computational Linguistics*, Reykjavik, Iceland.

HASSEL, M. (2004). Evaluation of automatic text summarization - a practical implementation. Master's thesis, Stockholm University. Licentiate Thesis, KTH NADA.

HOBBS, J. (1986). Resolving pronoun references. *Readings in natural language processing*, pages 339–352.

HOVY, E. et LIN, C.-Y. (1999). Automating text summarization in summarist. *Advances in automatic text summarization*, pages 81–94. The MIT Press.

INXIGHT (2002). Inxight summarizer : Managing the information deluge. Technical Report, Insight Software. `http://www.inxight.com/pdfs/summarizer_managing_deluge.pdf` (Page consultée le 04 août 2004).

LANDAUER, T. K., FOLTZ, P. W., et LAHAM, D. (1998). Introduction to latent semantic analysis. *Discourse Processes*, **25**, 259–284.

LAPPIN, S. et LEASS, H. J. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*, **20**(4), 535–561.

LIE, D. (1998). Sumatra : A system for automatic summary generation. In *Proceedings of the 14th Twente Workshop on Language Technology*, pages 173–176.

LIN, C.-Y. (1999). Machine translation for information access across the language barrier : the must system. *Proceedings of the Machine Translation Summit VII*. 13-17 September 1999.

LIN, C.-Y. (2001). Summary evaluation environment. `http://www.isi.edu/~cyl/SEE`.

LUHN, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal*, pages 159–165. April 1958.

MANI, I. (2001a). *Automatic Summarization*. John Benjamins B.V.

MANI, I. (2001b). Summarization evaluation : An overview. *In Proceedings of the second NTCIR Workshop Meeting on Evaluation of Chinese and Japanese Text*.

MANI, I. et MAYBURY, M. T. (1999). *Advances in automatic text summarization*. The MIT Press, U.S.A.

MARCU, D. et GERBER, L. (2001). An inquiry into the nature of multidocument abstracts, extracts, and their evaluation. In *Proceedings of the NAACL-2001 Workshop on Automatic Summarization*, Pittsburgh, PA. June 3, 2001.

MCCAGAR, V. (2004). Statistical approaches to automatic text summarization. *Bulletin of the American Society for Information Science and Technology*, **30**(4). April/May 2004.

MILLER, T. (2003). Latent semantic analysis and the construction of coherent extracts. In ANGELOVA, G., BONTCHEVA, K., MITKOV, R., NICOLOV, N., et NIKOLOV, N., editors, *Proceedings of the International Conference RANLP-2003 (Recent Advances in Natural Language Processing)*, pages 270–277. September 2003.

MINDS (1997). Hypertext summary extraction for fast document browsing. Technical Report, Pertinence Mining. `http://crl.nmsu.edu/Research/Projects/minds/core_summarizer/talk/talk1.%htm` (Page consultée le 27 mars 2005).

MINEL, J.-L., DESCLÉS, J.-P., CARTIER, E., CRISPINO, G., BEN HAZEZ, S., et JACKIEWICZ, A. (2001). Résumé automatique par filtrage sémantique d'informations dans des textes. *Revue Technique et Science Informatiques no. 3.*

MITKOV, R. (2002). *Anaphora Resolution.* Longman.

MORRIS, A., KASPER, G., et ADAMS, D. (1999). The effects ans limitations of automated text condensing on reading comprehension performance. *Advances in automatic text summarization*, pages 305–323. The MIT Press.

PERTINENCE (2004). Fiche descriptive : pertinence summarizer. Technical Report, Pertinence Mining. `http://www.pertinence.net/ps/PSummarizer.pdf` (Page consultée le 04 août 2004).

RATH, G. J., RESNICK, A., et SAVAGE, T. R. (1999). The formation of abstracts by the selection of sentences. *Advances in automatic text summarization*, pages 287–291. The MIT Press, U.S.A.

SAGGION, H. et LAPALME, G. (2000). Selective analysis for automatic abstracting : Evaluating indicativeness and acceptability. *RIAO'2000 (Recherche d'Informations Assistée par Ordinateur).* 12-14 April 2000.

SALTON, G. et MCGILL, M. J. (1986). *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA.

SCHIFFMAN, B., NENKOVA, A., et MCKEOWN, K. (2002). Experiments in multidocument summarization. *Proceedings of HLT Human Language Technology Conference.*

SPARCK-JONES, K. (1999). Automatic summarizing : factors and directions. *Advances in automatic text summarization*, pages 1–12. The MIT Press, U.S.A.

ST-ONGE, P.-L. (2003). Cortex. Technical Report version 3.5.2, École Polytechnique de Montréal. 9 octobre 2003.

TORRES-MORENO, J.-M., VAZQUEZ, R., BELLOT, P., EL-BEZE, M., ST-ONGE, P.-L., et GAGNON, M. (2004). Coupling an automatic-summarization system with a question-answering system (qaas). Soumis pour publication, 2004.

TORRES-MORENO, J.-M., VELÁZQUEZ-MORALES, P., et MEUNIER, J.-G. (2001). Cortex : un algorithme pour la condensation automatique des textes. In *La cognition entre individu et société*, volume 2, ISC-Lyon France, page 365. ARCo 2001.

TORRES-MORENO, J.-M., VELÁZQUEZ-MORALES, P., et MEUNIER, J.-G. (2002). Condensés de textes par des méthodes numériques. *JADT (Journées internationales d'Analyse statistique des Données Textuelles) 2002*, **2**, 723–734.

WU, C.-W. et LIU, C.-L. (2003). Ontology-based text summarization for business news articles. *Proceedings of the ISCA Eighteenth International Conference on Computers and Their Applications (CATA'03)*, pages 389–392. 26-28 March 2003.

# ANNEXE I

# DONNÉES EN RAPPORT AU TEXTE *CYBERMÉDIAS*

## I.1 Contenu intégral du texte *Cybermédias*

[*Titre*] Médias : La croissance des cybermédias et la publicité sur Internet

[0] Le 27 octobre 1994, le magazine Wired lançait un rejeton, HotWired sur le World Wide Web d'Internet.

[1] Une naissance beaucoup moins spectaculaire que ne le furent celles de l'imprimerie, la radio ou la télévision.

[2] Pourtant, dans le monde en pleine effervescence d'Internet, l'arrivée de HotWired marque le début de la cybermédiatisation, le premier véritable média sur Internet.

[3] Non pas qu'il s'agisse du premier à diffuser de l'information.

[4] Ce qui fait de HotWired un précurseur est qu'il fut le premier à répondre à la définition des médias modernes : des véhicules qui transportent à la fois du contenu et de la publicité.

[5] En effet, dès son lancement, HotWired présentait des publicités d'annonceurs comme ATT, Sprint ou Volvo.

[6] Quelques semaines plus tard, Time Warner lançait Pathfinder, le site regroupant des versions électroniques de la plupart des propriétés médias du groupe comme Time, People, Entertainment Weekly ou Sports Illustrated.

[7] Comme HotWired, Pathfinder accordait aussi une grande place aux bandeaux publicitaires de ses annonceurs.

[8] Dans la course pour livrer du contenu sur Internet, les médias traditionnels déjà en place - surtout les imprimés - ont toujours eu une longueur d'avance.

[9] Leurs infrastructures nécessitaient un minimum d'adaptation pour que leur contenu soit rediffusé sur le Web.

[10] Mais en arrivant sur Internet, ces grands médias ont vite essayé d'adapter leur modèle "commercial" au réseau des réseaux.

[11] Et dans ce modèle, sans publicité, il ne peut y avoir de contenu.

[12] Bien entendu, le modèle a évolué au fil des mois.

[13] À part quelques cas isolés comme le site Sportszone de la chaîne américaine spécialisée en sports ESPN, les sites des grands médias ne sont pas encore rentables.

[14] Time Warner, pour ne nommer que celui-là, engloutit chaque année des millions de dollars dans son site.

[15] La publicité soulage, mais n'est pas un remède au malaise provoqué par des revenus largement inférieurs aux dépenses.

[16] D'autres ont essayé de transposer sur le Web un modèle d'abonnement.

[17] L'échec le plus notoire fut sans doute celui du quotidien national américain USA Today qui, dans sa première version électronique, exigeait un abonnement de 14,95 $US par mois.

[18] Le site a rapidemement fermé ses portes pour rouvrir, quelques mois plus tard, accessible gratuitement et financé largement par la publicité.

[19] Le magazine Playboy retarde depuis plusieurs mois la mise en place d'un système d'abonnement.

[20] ESPN réserve quelques sections de son site à des abonnés payants.

[21] Enfin, le Wall Street Journal Interactive vient tout juste de célébrer son premier anniversaire.

[22] Accessible uniquement par abonnement, ses dirigeants estiment qu'il sera rentable dans deux ans.

[23] Ce n'est donc pas la promesse de revenus faramineux qui pousse les médias sur le Web.

[24] En fait, la formule gagnante n'a toujours pas été trouvée.

[25] On peut toutefois prévoir qu'au fur et à mesure que la portée d'Internet va grandir, les sites qui exigent un abonnement et ceux qui vendent de la publicité verront leurs revenus croître substantiellement.

[26] À condition que soient réglées les différentes questions entourant la mesure de l'achalandage des sites et la mesure de visibilité des publicités.

[27] Dans les médias traditionnels, le coût d'une publicité dépend avant tout du nombre de lecteurs, d'auditeurs ou de téléspectateurs qui a l'occasion de lire, d'entendre ou de voir cette publicité.

[28] D'où l'importance des mesures d'auditoires comme Nielsen à la télévision, BBM à la radio ou PMB pour les magazines.

[29] Ces mesures garantissent aux annonceurs que leur publicité rejoint non seulement un nombre prédéterminé d'auditeurs, mais également que ce public correspond à un certain profil sociodémographique susceptible d'intéresser tel ou tel annonceur.

[30] Sur le Web, rien de tout cela.

[31] En fait, les premiers médias sur Internet ont capitalisé sur leur notoriété en dehors du cyberespace pour attirer les annonceurs.

[32] Ou encore, comme ESPN ou HotWired, ils ciblent des publics tellement précis que les annonceurs ne peuvent tout simplement pas passer à côté.

[33] Il règne une grande confusion autour de tout ce qui concerne la mesure des auditoires sur Internet.

[34] On commence à peine à différencier le nombre de "hits" du nombre de visiteurs sur un site, alors que cette distinction est fondamentale pour tout acheteur de publicité.

[35] Et, paradoxalement, Internet est le média où les données sont excessivement faciles à obtenir, à partir du serveur même.

[36] Des logiciels abordables facilitent grandement cette tâche d'analyse, de sorte que la plupart des sites pourraient fournir à leurs annonceurs des chiffres sérieux.

[37] Au Québec, le Groupe de travail sur les applications publicitaires d'Internet réunit des représentants d'agences de publicité, des acheteurs médias, des annonceurs et des éditeurs de sites.

[38] Ensemble, ils cherchent à harmoniser les différentes normes publicitaires afin de faciliter la vente et l'achat de publicité sur les sites Web.

[39] Récemment, l'agence Cossette Interactif et le GTAPI ont dévoilé une vaste étude sur la mesure interactive, un premier pas vers l'établissement de bases solides pour éduquer et guider les entreprises qui désirent offrir ou acheter de la publicité sur Internet.

[40] Accessible sur le site du Mondial de la publicité francophone, l'étude fournit, entre autres, un lexique complet des différents termes employés dans le domaine.

[41] On y trouve également une revue sommaire des logiciels qui analysent le journal des serveurs.

[42] Une fois ces éléments bien compris, il reste la question de la validation des données.

[43] Les annonceurs aiment bien être rassurés, ne pas se lancer à l'aveuglette dans un projet publicitaire, même si les chiffres qu'on leur fournit semblent crédibles.

[44] Les firmes comme BBM ou Nielsen offrent déjà des services de validation de ces données et, à moins de trouver une norme à laquelle tous les sites vont se conformer, l'influence des BBM ou Nielsen ne fera que grandir.

[45] Évidemment, il serait tellement simple de passer du modèle des médias traditionnels.

[46] Mais le Web n'est pas si simple.

[47] Dans quelque média que ce soit, on ne peut jamais être sûr à 100 % qu'un message publicitaire sera lu, entendu ou vu par la cible qu'il vise.

[48] C'est un risque que les annonceurs ont toujours été prêts à prendre, faute de mieux.

[49] Internet, par contre, est une tout autre histoire.

[50] Ayant longtemps vanté leurs médias comme étant véritablement "instantanés" et "interactifs", les éditeurs de sites Web sont aujourd'hui aux prises avec des annonceurs qui veulent profiter de cette interactivité.

[51] En clair : sur Internet, un annonceur peut, s'il le veut, exiger de payer seulement pour la proportion des gens qui "clique" sur sa publicité, pas pour l'ensemble des gens qui risquent de la voir.

[52] La différence est énorme.

[53] Quand Procter Gamble, le plus grand annonceur, tous médias confondus, de la planète, a commencé à acheter de la publicité sur des sites Internet, il a exigé d'être facturé au "clic".

[54] Et ça ne s'arrête pas là.

[55] Peu avant le lancement de la version 4.0 de son fureteur Internet Explorer, le fabricant Microsoft aurait exigé des éditeurs de sites d'être facturé non plus au "clic" mais au téléchargement.

[56] Prenons un exemple dans les médias traditionnels : c'est l'équivalent, pour un concessionnaire automobile, de payer sa publicité dans un quotidien seulement si celle-ci lui rapporte des ventes.

[57] Télévision, radio et imprimés traditionnels ne pourront jamais livrer concurrence au Web sur cette base.

[58] Mais pour les sites, c'est un couteau à deux tranchants.

[59] Car ce genre de procédé soulève des questions plus fondamentales : les médias sont-ils responsables de l'efficacité des publicités qu'ils véhiculent ?

[60] Ne devraient-ils pas seulement fournir un environnement auquel la publicité doit s'accommoder ?

[61] Il faut croire qu'avec l'arrivée des cybermédias, bien des règles vont changer.

## I.2 Choix des évaluateurs pour le texte *Cybermédias*

Tableau I.1 Tableau répertoriant le choix des évaluateurs pour le texte *Cybermédias*

| Évaluateur A | Évaluateur B | Évaluateur C |
|---|---|---|
| 4 8 9 13 15 23 24 | 4 8 13 25 26 27 28 | 2 4 8 13 15 27 30 |
| 28 29 33 36 39 44 | 33 34 35 43 44 50 | 39 51 55 58 61 |
| 51 56 57 59 | 51 53 55 59 60 61 | |

# ANNEXE II

# RÉSULTATS DES TESTS SUR L'HEURISTIQUE POUR LE CHOIX DES MÉTRIQUES

## II.1 Distribution des valeurs maximales et des métriques pour le texte *Cybermédias*

Phrase #0, Valeur Maximale = 1.00, Métrique associee = X
Phrase #1, Valeur Maximale = 0.88, Métrique associee = X
Phrase #2, Valeur Maximale = 0.85, Métrique associee = A
Phrase #4, Valeur Maximale = 0.76, Métrique associee = A
Phrase #5, Valeur Maximale = 0.49, Métrique associee = X
Phrase #6, Valeur Maximale = 0.83, Métrique associee = F
Phrase #7, Valeur Maximale = 0.42, Métrique associee = F
Phrase #8, Valeur Maximale = 0.64, Métrique associee = A
Phrase #9, Valeur Maximale = 0.25, Métrique associee = X
Phrase #10, Valeur Maximale = 0.58, Métrique associee = F
Phrase #11, Valeur Maximale = 0.48, Métrique associee = A
Phrase #12, Valeur Maximale = 0.17, Métrique associee = F
Phrase #13, Valeur Maximale = 0.62, Métrique associee = D
Phrase #14, Valeur Maximale = 0.25, Métrique associee = F
Phrase #15, Valeur Maximale = 0.48, Métrique associee = A
Phrase #16, Valeur Maximale = 0.25, Métrique associee = F
Phrase #17, Valeur Maximale = 0.50, Métrique associee = F
Phrase #18, Valeur Maximale = 0.50, Métrique associee = D
Phrase #19, Valeur Maximale = 0.25, Métrique associee = F
Phrase #20, Valeur Maximale = 0.31, Métrique associee = E
Phrase #21, Valeur Maximale = 0.08, Métrique associee = F
Phrase #22, Valeur Maximale = 0.17, Métrique associee = F
Phrase #23, Valeur Maximale = 0.48, Métrique associee = A
Phrase #24, Valeur Maximale = 0.00, Métrique associee = X
Phrase #25, Valeur Maximale = 1.00, Métrique associee = L
Phrase #26, Valeur Maximale = 0.60, Métrique associee = D
Phrase #27, Valeur Maximale = 0.80, Métrique associee = D
Phrase #28, Valeur Maximale = 0.58, Métrique associee = F

Phrase #29, Valeur Maximale = 0.52, Métrique associee = E
Phrase #30, Valeur Maximale = 0.07, Métrique associee = T
Phrase #31, Valeur Maximale = 1.00, Métrique associee = A
Phrase #32, Valeur Maximale = 0.33, Métrique associee = F
Phrase #33, Valeur Maximale = 0.42, Métrique associee = A
Phrase #34, Valeur Maximale = 0.57, Métrique associee = D
Phrase #35, Valeur Maximale = 0.76, Métrique associee = A
Phrase #36, Valeur Maximale = 0.50, Métrique associee = F
Phrase #37, Valeur Maximale = 1.00, Métrique associee = D
Phrase #38, Valeur Maximale = 0.71, Métrique associee = E
Phrase #39, Valeur Maximale = 0.75, Métrique associee = F
Phrase #40, Valeur Maximale = 0.50, Métrique associee = D
Phrase #41, Valeur Maximale = 0.33, Métrique associee = F
Phrase #42, Valeur Maximale = 0.17, Métrique associee = F
Phrase #43, Valeur Maximale = 0.33, Métrique associee = F
Phrase #44, Valeur Maximale = 1.00, Métrique associee = F
Phrase #45, Valeur Maximale = 0.42, Métrique associee = A
Phrase #46, Valeur Maximale = 0.07, Métrique associee = T
Phrase #47, Valeur Maximale = 0.37, Métrique associee = A
Phrase #48, Valeur Maximale = 0.17, Métrique associee = F
Phrase #49, Valeur Maximale = 0.85, Métrique associee = A
Phrase #50, Valeur Maximale = 0.77, Métrique associee = D
Phrase #51, Valeur Maximale = 0.75, Métrique associee = F
Phrase #52, Valeur Maximale = 0.25, Métrique associee = X
Phrase #53, Valeur Maximale = 0.91, Métrique associee = D
Phrase #55, Valeur Maximale = 0.56, Métrique associee = E
Phrase #56, Valeur Maximale = 0.64, Métrique associee = A
Phrase #57, Valeur Maximale = 0.57, Métrique associee = X
Phrase #58, Valeur Maximale = 0.66, Métrique associee = X
Phrase #59, Valeur Maximale = 0.85, Métrique associee = A
Phrase #60, Valeur Maximale = 0.88, Métrique associee = X
Phrase #61, Valeur Maximale = 1.00, Métrique associee = X

*NB : les phrases #3 et #54 sont absentes puisqu'elles ne contiennent aucun mot retenu dans la matrice de fréquences !*

## II.2 Tableaux de résultats des tests sur l'heuristique

Tableau II.1 Tableau comparatif des résultats entre la combinaison ETX et l'utilisation de l'heuristique, écart fixé à 0.2

| | ETX (%) | | | | Heuristique avec seuil minimum à 0.2 (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 36 | 45 | 9 | 30 |
| Épicier | 60 | 50 | 40 | **50** | 30 | 30 | 25 | 28 |
| Kanthume | 46 | 58 | 25 | **43** | 25 | 42 | 21 | 29 |
| Football | 73 | 36 | 55 | **55** | 55 | 45 | 27 | 42 |
| J'Accuse | 67 | 33 | 29 | **43** | 33 | 24 | 19 | 25 |
| Univers | 43 | 50 | 50 | **48** | 29 | 36 | 36 | 34 |
| Opus Dei | 36 | 40 | 44 | **40** | 40 | 29 | 36 | 35 |
| Cybermédias | 14 | 57 | 14 | **28** | 29 | 57 | 29 | 38 |
| Sciences | 39 | 52 | 52 | **47** | 39 | 30 | 39 | 36 |
| Sirène | 63 | 40 | 51 | **51** | 37 | 26 | 17 | 27 |
| Travail | 36 | 16 | 28 | **27** | 28 | 12 | 28 | 23 |
| Moyenne | | | | 43 | | | | 32 |

Tableau II.2 Tableau comparatif des résultats entre la combinaison ETX et l'utilisation de l'heuristique, écart fixé à 0.2 et seuil additionnel à 0.4

| | ETX (%) | | | | Heuristique avec seuil minimum à 0.2 et seuil supérieur à 0.4 (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 36 | 45 | 9 | 30 |
| Épicier | 60 | 50 | 40 | **50** | 30 | 40 | 30 | 33 |
| Kanthume | 46 | 58 | 25 | **43** | 25 | 46 | 29 | 33 |
| Football | 73 | 36 | 55 | **55** | 55 | 55 | 45 | 52 |
| J'Accuse | 67 | 33 | 29 | **43** | 33 | 24 | 24 | 27 |
| Univers | 43 | 50 | 50 | **48** | 36 | 36 | 36 | 36 |
| Opus Dei | 36 | 40 | 44 | **40** | 40 | 29 | 36 | 35 |
| Cybermédias | 14 | 57 | 14 | **28** | 29 | 57 | 29 | 38 |
| Sciences | 39 | 52 | 52 | **47** | 39 | 30 | 39 | 36 |
| Sirène | 63 | 40 | 51 | **51** | 43 | 29 | 23 | 32 |
| Travail | 36 | 16 | 28 | **27** | 28 | 12 | 28 | 23 |
| Moyenne | | | | 43 | | | | 34 |

Tableau II.3 Tableau comparatif des résultats entre la combinaison ETX et l'utilisation de l'heuristique, écart fixé à 0.2 et seuil additionnel à 0.5

| | ETX (%) | | | | Heuristique avec seuil minimum à 0.2 et seuil supérieur à 0.5 (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 36 | 45 | 9 | 30 |
| Épicier | 60 | 50 | 40 | **50** | 30 | 30 | 25 | 28 |
| Kanthume | 46 | 58 | 25 | **43** | 29 | 46 | 25 | 33 |
| Football | 73 | 36 | 55 | **55** | 45 | 55 | 45 | 48 |
| J'Accuse | 67 | 33 | 29 | **43** | 33 | 24 | 19 | 25 |
| Univers | 43 | 50 | 50 | **48** | 36 | 43 | 43 | 41 |
| Opus Dei | 36 | 40 | 44 | **40** | 40 | 29 | 36 | 35 |
| Cybermédias | 14 | 57 | 14 | **28** | 29 | 57 | 29 | 38 |
| Sciences | 39 | 52 | 52 | **47** | 39 | 30 | 39 | 36 |
| Sirène | 63 | 40 | 51 | **51** | 34 | 29 | 17 | 27 |
| Travail | 36 | 16 | 28 | **27** | 28 | 12 | 28 | 23 |
| Moyenne | | | | 43 | | | | 33 |

Tableau II.4 Tableau comparatif des résultats entre la combinaison ETX et l'utilisation de l'heuristique, écart fixé à 0.1

| | ETX (%) | | | | Heuristique avec seuil minimum à 0.1 | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 50 | 45 | 9 | 35 |
| Épicier | 60 | 50 | 40 | **50** | 35 | 45 | 30 | 37 |
| Kanthume | 46 | 58 | 25 | **43** | 25 | 42 | 21 | 29 |
| Football | 73 | 36 | 55 | **55** | 45 | 36 | 27 | 36 |
| J'Accuse | 67 | 33 | 29 | **43** | 33 | 19 | 24 | 25 |
| Univers | 43 | 50 | 50 | **48** | 29 | 43 | 29 | 34 |
| Opus Dei | 36 | 40 | 44 | **40** | 40 | 29 | 36 | 35 |
| Cybermédias | 14 | 57 | 14 | **28** | 29 | 57 | 29 | 38 |
| Sciences | 39 | 52 | 52 | **47** | 39 | 26 | 35 | 33 |
| Sirène | 63 | 40 | 51 | **51** | 37 | 23 | 17 | 26 |
| Travail | 36 | 16 | 28 | **27** | 28 | 12 | 28 | 23 |
| Moyenne | | | | 43 | | | | 32 |

Tableau II.5 Tableau comparatif des résultats entre la combinaison ETX et l'utilisation de l'heuristique, écart fixé à 0.1 et seuil additionnel à 0.2

| | ETX (%) | | | | Heuristique avec seuil minimum à 0.1 et seuil supérieur à 0.2 (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 41 | 41 | 9 | 30 |
| Épicier | 60 | 50 | 40 | **50** | 25 | 35 | 30 | 30 |
| Kanthume | 46 | 58 | 25 | **43** | 21 | 42 | 25 | 29 |
| Football | 73 | 36 | 55 | **55** | 36 | 27 | 27 | 30 |
| J'Accuse | 67 | 33 | 29 | **43** | 33 | 29 | 19 | 27 |
| Univers | 43 | 50 | 50 | **48** | 29 | 36 | 29 | 31 |
| Opus Dei | 36 | 40 | 44 | **40** | 40 | 29 | 36 | 35 |
| Cybermédias | 14 | 57 | 14 | **28** | 29 | 57 | 29 | 38 |
| Sciences | 39 | 52 | 52 | **47** | 35 | 30 | 39 | 35 |
| Sirène | 63 | 40 | 51 | **51** | 46 | 29 | 23 | 33 |
| Travail | 36 | 16 | 28 | **27** | 28 | 12 | 28 | 23 |
| Moyenne | | | | 43 | | | | 31 |

Tableau II.6 Tableau comparatif des résultats entre la combinaison ETX et l'utilisation de l'heuristique, écart fixé à 0.1 et seuil additionnel à 0.3

| | ETX (%) | | | | Heuristique avec seuil minimum à 0.1 et seuil supérieur à 0.3 (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 50 | 45 | 5 | 33 |
| Épicier | 60 | 50 | 40 | **50** | 30 | 35 | 30 | 32 |
| Kanthume | 46 | 58 | 25 | **43** | 21 | 42 | 25 | 29 |
| Football | 73 | 36 | 55 | **55** | 45 | 45 | 36 | 42 |
| J'Accuse | 67 | 33 | 29 | **43** | 38 | 24 | 24 | 29 |
| Univers | 43 | 50 | 50 | **48** | 36 | 43 | 36 | 38 |
| Opus Dei | 36 | 40 | 44 | **40** | 40 | 29 | 36 | 35 |
| Cybermédias | 14 | 57 | 14 | **28** | 29 | 57 | 29 | 38 |
| Sciences | 39 | 52 | 52 | **47** | 39 | 26 | 35 | 33 |
| Sirène | 63 | 40 | 51 | **51** | 40 | 29 | 23 | 31 |
| Travail | 36 | 16 | 28 | **27** | 28 | 12 | 28 | 23 |
| Moyenne | | | | 43 | | | | 33 |

Tableau II.7 Tableau comparatif des résultats entre la combinaison ETX et l'utilisation de l'heuristique, écart fixé à 0.1 et seuil additionnel à 0.5

| | ETX (%) | | | | Heuristique avec seuil minimum à 0.1 et seuil supérieur à 0.5 (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 50 | 45 | 9 | 35 |
| Épicier | 60 | 50 | 40 | **50** | 35 | 45 | 30 | 37 |
| Kanthume | 46 | 58 | 25 | **43** | 29 | 46 | 25 | 33 |
| Football | 73 | 36 | 55 | **55** | 45 | 45 | 45 | 45 |
| J'Accuse | 67 | 33 | 29 | **43** | 33 | 19 | 24 | 25 |
| Univers | 43 | 50 | 50 | **48** | 43 | 43 | 43 | 43 |
| Opus Dei | 36 | 40 | 44 | **40** | 40 | 29 | 36 | 35 |
| Cybermédias | 14 | 57 | 14 | **28** | 29 | 57 | 29 | 38 |
| Sciences | 39 | 52 | 52 | **47** | 39 | 26 | 35 | 33 |
| Sirène | 63 | 40 | 51 | **51** | 34 | 23 | 14 | 24 |
| Travail | 36 | 16 | 28 | **27** | 28 | 12 | 28 | 23 |
| Moyenne | | | | 43 | | | | 34 |

# ANNEXE III

## RÉSULTATS DE L'ANALYSE SYNTAXIQUE

### III.1 Tableaux de résultats issus de la bonification de certains mots

Tableau III.1 Tableau comparatif des résultats avec l'utilisation d'un bonus pour les verbes principaux

| | ETX (%) | | | | ETX+Verbe (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 59 | 55 | 18 | 44 |
| Épicier | 60 | 50 | 40 | **50** | 50 | 60 | 40 | 50 |
| Kanthume | 46 | 58 | 25 | **43** | 46 | 58 | 29 | 44 |
| Football | 73 | 36 | 55 | **55** | 73 | 27 | 45 | 48 |
| J'Accuse | 67 | 33 | 29 | **43** | 57 | 29 | 19 | 35 |
| Univers | 43 | 50 | 50 | **48** | 36 | 29 | 29 | 31 |
| Opus Dei | 36 | 40 | 44 | **40** | 38 | 40 | 49 | 42 |
| Cybermédias | 14 | 57 | 14 | **28** | 0 | 43 | 29 | 24 |
| Sciences | 39 | 52 | 52 | **47** | 39 | 48 | 48 | 45 |
| Sirène | 63 | 40 | 51 | **51** | 31 | 31 | 34 | 32 |
| Travail | 36 | 16 | 28 | **27** | 40 | 20 | 32 | 31 |
| Moyenne | | | | 43 | | | | 39 |

Tableau III.2 Tableau comparatif des résultats avec l'utilisation d'un filtrage en plus d'un bonus pour les verbes principaux

| | ETX (%) | | | | ETX+Filtre+Verbe (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 55 | 50 | 9 | 38 |
| Épicier | 60 | 50 | 40 | **50** | 50 | 55 | 30 | 45 |
| Kanthume | 46 | 58 | 25 | **43** | 50 | 58 | 29 | 46 |
| Football | 73 | 36 | 55 | **55** | 82 | 36 | 55 | 58 |
| J'Accuse | 67 | 33 | 29 | **43** | 67 | 33 | 33 | 44 |
| Univers | 43 | 50 | 50 | **48** | 57 | 21 | 29 | 36 |
| Opus Dei | 36 | 40 | 44 | **40** | 31 | 47 | 42 | 40 |
| Cybermédias | 14 | 57 | 14 | **28** | 43 | 57 | 14 | 38 |
| Sciences | 39 | 52 | 52 | **47** | 43 | 61 | 48 | 51 |
| Sirène | 63 | 40 | 51 | **51** | 29 | 37 | 40 | 35 |
| Travail | 36 | 16 | 28 | **27** | 32 | 16 | 44 | 31 |
| Moyenne | | | | 43 | | | | 42 |

Tableau III.3 Tableau comparatif des résultats avec l'utilisation d'un bonus pour les verbes principaux et les sujets

| | ETX (%) | | | | ETX+Sujet+Verbe (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 59 | 55 | 18 | 44 |
| Épicier | 60 | 50 | 40 | **50** | 55 | 55 | 40 | 50 |
| Kanthume | 46 | 58 | 25 | **43** | 46 | 58 | 29 | 44 |
| Football | 73 | 36 | 55 | **55** | 73 | 36 | 55 | 55 |
| J'Accuse | 67 | 33 | 29 | **43** | 57 | 33 | 24 | 38 |
| Univers | 43 | 50 | 50 | **48** | 36 | 29 | 29 | 31 |
| Opus Dei | 36 | 40 | 44 | **40** | 38 | 40 | 49 | 42 |
| Cybermédias | 14 | 57 | 14 | **28** | 0 | 43 | 29 | 24 |
| Sciences | 39 | 52 | 52 | **47** | 39 | 48 | 48 | 45 |
| Sirène | 63 | 40 | 51 | **51** | 31 | 31 | 34 | 32 |
| Travail | 36 | 16 | 28 | **27** | 40 | 20 | 32 | 31 |
| Moyenne | | | | 43 | | | | 40 |

Tableau III.4 Tableau comparatif des résultats avec l'utilisation filtrage en plus d'un bonus pour les verbes principaux et les sujets

| | ETX (%) | | | | ETX+Filtre+ Filtrage+Sujet+Verbe (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 55 | 50 | 9 | 38 |
| Épicier | 60 | 50 | 40 | **50** | 45 | 55 | 30 | 43 |
| Kanthume | 46 | 58 | 25 | **43** | 50 | 58 | 29 | 46 |
| Football | 73 | 36 | 55 | **55** | 82 | 36 | 55 | 58 |
| J'Accuse | 67 | 33 | 29 | **43** | 62 | 33 | 29 | 41 |
| Univers | 43 | 50 | 50 | **48** | 57 | 21 | 29 | 36 |
| Opus Dei | 36 | 40 | 44 | **40** | 29 | 47 | 44 | 40 |
| Cybermédias | 14 | 57 | 14 | **28** | 43 | 43 | 14 | 33 |
| Sciences | 39 | 52 | 52 | **47** | 52 | 57 | 43 | 51 |
| Sirène | 63 | 40 | 51 | **51** | 29 | 34 | 37 | 33 |
| Travail | 36 | 16 | 28 | **27** | 32 | 16 | 44 | 31 |
| Moyenne | | | | 43 | | | | 41 |

Tableau III.5 Tableau comparatif des résultats avec l'utilisation d'un filtrage en plus d'un bonus pour les sujets

| | ETX (%) | | | | ETX+Filtre+Sujet (%) | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | Moyenne | A | B | C | Moyenne |
| Durham | 55 | 50 | 18 | **41** | 55 | 50 | 9 | 38 |
| Épicier | 60 | 50 | 40 | **50** | 50 | 55 | 30 | 45 |
| Kanthume | 46 | 58 | 25 | **43** | 50 | 58 | 29 | 46 |
| Football | 73 | 36 | 55 | **55** | 82 | 45 | 64 | 64 |
| J'Accuse | 67 | 33 | 29 | **43** | 62 | 33 | 29 | 41 |
| Univers | 43 | 50 | 50 | **48** | 50 | 36 | 21 | 36 |
| Opus Dei | 36 | 40 | 44 | **40** | 33 | 49 | 44 | 42 |
| Cybermédias | 14 | 57 | 14 | **28** | 43 | 57 | 14 | 38 |
| Sciences | 39 | 52 | 52 | **47** | 48 | 61 | 48 | 52 |
| Sirène | 63 | 40 | 51 | **51** | 29 | 34 | 37 | 33 |
| Travail | 36 | 16 | 28 | **27** | 32 | 16 | 44 | 31 |
| Moyenne | | | | 43 | | | | 42 |