Université de Montréal

Utilisation des citations pour le résumé automatique de la contribution d'articles scientifiques

par
Bruno Malenfant

Département d'informatique
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

décembre, 2016

RÉSUMÉ

Cette thèse cherche à construire des outils pour la communauté scientifique. Une des tâches d'un chercheur est la lecture d'articles scientifiques, que ce soit pour les comparer, pour identifier de nouveaux problèmes, pour situer son travail dans la littérature courante ou pour définir des propositions de recherche. Nous avons appliqué, combiné et modifié des techniques de résumé automatique pour la littérature scientifique. L'idée est de construire le résumé à partir de l'information que d'autres chercheurs ont retenue d'un l'article de référence. Plus particulièrement, le texte des citations vers l'article de référence est utilisé pour constituer la base du résumé. Ce résumé est donc construit à partir de l'analyse de plusieurs autres qui le citent.

Une citation est un élément qu'un autre auteur (ou le même) a retenu en lisant l'article. À l'intérieur d'une citation, il y a une description des liens entre plusieurs articles. Cette information n'étant pas disponible lors de l'écriture de l'article, cela lui ajoute un niveau d'interprétation et nous donne un indice sur l'apport de l'article à la communauté scientifique.

Pour construire le résumé d'un article, nous trouvons tous les articles qui lui font référence à l'aide d'une base de données `RDF` construite à partir des données de l'*ACL Anthology Network*. Ensuite, les citations sont extraites et classées selon leur contexte rhétorique. Afin de construire le résumé à l'aide de l'information trouvée, une technique basée sur la *Maximal Marginal Relevance* choisit certaines phrases parmi les citations en évitant la redondance. Finalement, le résumé est amélioré à l'aide d'extraits du texte original.

Mots clés : informatique, linguistique, langue naturelle, résumé automatique, analyse d'articles scientifiques.

ABSTRACT

The goals of this thesis are to build and improve tools for the scientific community. One of the tasks of a researcher is to read scientific papers, in order to compare them, identify new problems, place the work within the current literature or define new research proposals. We applied, combined and modified techniques of automatic summarization for the scientific literature. The underlying idea is to build the summary from the information that other researchers retained from a given paper called a reference paper. More particularly, the text of citations towards the reference paper is used for the base of the summary. The summary of the reference paper will thus be built from the analysis from several others who quote it.

A citation is an element which another author (or the same) remembered from reading the paper. Inside a citation, there is a description of the links between several papers. This information was not available when writing the original paper, it thus adds a level of interpretation to the paper. It gives an indication of the contribution of the paper to the scientific community. The set of citations reflects the opinion of the scientific community (community insight).

To build the summary of a paper, we find papers which reference to it. For this, we use a `RDF` database built from the data from the ACL Anthology Network. Then citations are extracted and classified according to their rhetorical context. To build the summary, we use a Maximal Marginal Relevance based technique to choose sentences among citations while avoiding the redundancy. Finally, the summary is improved by adding extracts from the original text.

Keywords: computer science, linguistic, natural language, automatic summarization, scientific paper analysis.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTINGS

LISTE DES SIGLES

AA	ACL Anthology
AAN	ACL Anthology Network
ACL	Association for Computational Linguistics
BiomedSumm 2014	Biomedical Summarization Track 2014
BOM	Byte Order Mark
CC · IDF	Common Citation × Inverse Document Frequency
CiTO	Citation Typing Ontology
CL	Computational Linguistics
DC	Dublin Core
EACL	Association for Computational Linguistis - European Chapter
EMNLP	Empirical Methods in Natural Language Processing
ESWC	European Semantic Web Conference
FOAF	Friend Of A Friend
HTML	HyperText Markup Language
ICCL	Int'l Committee on Computational Linguistics
IJCNLP	International Joint Conference on Natural Language Processing
JATS	Journal Article Tag Suite

JEP	Journées d'Études sur la Parole
LST	Lexique Scientifique Transdisciplinaire
MMR	Maximal Marginal Relevance
NAACL	The North American Chapter of the Association for Computational Linguistics
NIST	National Institute of Standards and Technology
NLTK	Natural Language Toolkit
OCR	Optical Character Recognition
PDF	Portable Document Forma
RDF	Resource Description Framework
RECITAL	Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues
RNC	RELAX NG Schema - Compact
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
SIG	Special Interest Group
SPAR	Semantic Publishing and Referencing Ontologies
SPARQL	SPARQL Protocol and RDF Query Language
TAC 2014	Text Analysis Conference
TALN	Traitement Automatique des Langues Naturelles
TF·IDF	Term Frequency × Inverse Document Frequency
UTF	Universal Character Set Transformation Format

WebNLG	International Workshop on Natural Language Generation and the Semantic Web
XML	Extensible Markup Language

REMERCIEMENTS

J'aimerais remercier mon directeur, Guy Lapalme, pour son enseignement et sa patience.

Merci à ma compagne Silvi pour ses encouragements et sa patience tout au long de mes études.

Merci à mes parents, Yolande et Laurien, pour leurs supports et conseils.

CHAPITRE 1

INTRODUCTION

Mon projet de doctorat avait pour objectif de construire et d'améliorer des outils pour la communauté scientifique. Une des tâches d'un chercheur est la lecture d'articles scientifiques, que ce soit pour les comparer, pour identifier de nouveaux problèmes, pour situer son travail dans la littérature courante ou pour définir des propositions de recherche [16]. Nous voulons appliquer, combiner et modifier des techniques de résumé automatique pour la littérature scientifique. L'idée est de construire le résumé à partir de l'information que d'autres chercheurs ont retenue d'un article de référence. Plus particulièrement, le texte des citations vers l'article de référence sera utilisé pour constituer la base du résumé. Le résumé d'un article sera donc construit à partir de l'analyse de plusieurs autres qui le citent. Ce résumé va refléter l'impact qu'un article à eu sur la communauté scientifique.

1.1 Définition du problème

> As the amount of on-line information increases, systems that can automatically summarize one or more documents become increasingly desirable.[29].

Cette phrase peut être lue en entête de la plupart des articles sur les résumés de texte automatique. Bien sûr, elle peut être reformulée autrement :

> With the mushrooming of the quantity of on-line text information, triggered in part by the growth of the World Wide Web, it is especially useful to have tools which can help users digest information content.[20]

Des articles du même domaine répètent souvent certaines informations. Pour trouver ce qu'un article ajoute au discours scientifique, un chercheur doit lire plusieurs sections qui contiennent de l'information déjà connue. Le travail d'un chercheur en devient plus ardu, que ce soit pour être à jour, pour trouver des références ou pour s'assurer que son travail n'a pas déjà été publié. Des revues de littérature sont souvent

construites par des chercheurs pour résumer des découvertes passées dans un domaine spécifique.

Plusieurs solutions informatiques sont utilisées pour aider les chercheurs. Les techniques de résumé automatique d'articles scientifiques simples ou multiples permettent de déterminer le sujet de l'article ou de plusieurs articles. Le résumé sur plusieurs articles n'est pas facile, comme les deux extraits du paragraphe précédent le montrent, deux phrases peuvent être très différentes et pourtant exprimer la même idée.

Il existe aussi plusieurs systèmes d'extraction de citations et de référence. Ils sont très utilisés par les sites de références croisées comme *CiteSeer*, *Microsoft Academic Search* et *Google Scholar*. Une autre suggestion est d'utiliser l'ensemble des citations qui font référence à un article spécifique pour en déduire le contenu important ou marquant. Une citation est un élément qu'un autre auteur (ou le même) a retenu en lisant l'article. Récemment nous observons un intérêt grandissant entourant les citations. Le défi proposé à la conférence ESWC-14 contenait une tâche dont l'objectif était de caractériser les citations d'articles scientifiques et de déterminer leur qualité. La compétition TAC 2014 proposait de générer des résumés automatique d'articles en biologie à l'aide des citations. À l'intérieur d'une citation, il y a une description des liens entre plusieurs articles. Ces articles sont comparés, commentés et combinés. Cette information n'était pas disponible lors de l'écriture de l'article, cela ajoute un niveau d'interprétation de l'article. Cela nous donne un indice sur l'apport de l'article à la communauté scientifique. L'ensemble des citations permettrait d'obtenir un résumé reflétant l'opinion de la communauté scientifique (community insight) [4]. Les compétitions CL-2014 et CL-2016 ont repris cette idée sur des corpus d'articles ayant comme sujet le traitement automatique de la langue naturelle. Comme le demontrent ces compétitions, il y a un intérêt grandissant pour l'analyse et l'extraction automatique d'information contenus dans les articles scientifiques.

Pour construire le résumé d'un article nous devons trouver tous les articles qui lui font référence (voir figure 1.1). Ensuite, il nous faut extraire les citations avec leurs contextes. Le terme *citance* à été proposé par Preslav I. Nakov, Ariel S. Schwartz et Marti A. Hearst pour décrire l'ensemble des phrases entourant une citation[23]. Le texte entourant une citation va souvent évoquer des informations traitées dans l'article de référence. Ces informations sont généralement énoncées de façon concise et peuvent ajouter de l'information non présente dans l'article cité. Prenons les phrases suivantes.

White [*32*] provides a good recent review of the field of citation analysis

(for a more thorough but less recent review of the field see [*22*]). White describes three major lines of research in the field of citation analysis.

La première phrase est une citation utilisant le marqueur [*32*]. Par contre, la phrase suivante ajoute de l'information sur la citation : *three major lines of research*. Aussi nous remarquons que la première phrase contient une deuxième citation entre parenthèses. Pour avoir seulement l'information liée à la première citation, nous devrons donc extraire la première partie de la citation et la deuxième phrase complétant la sitation. C'est l'ensemble de ces extraits que nous appelons *citance*.

Dans son étude, Simone Teufel [38] énumère différents contextes rhétoriques qui peuvent être attribués à une phrase. Parmi ces sections, nous trouvons des éléments de *contraste* (négatif), d'*approbation* (positif) et *descriptifs* (neutre) qui sont attribués aux phrases d'une citation. Ces indices du contexte de la citation nous donnent l'opinion de l'auteur (du citant) sur le document cité. Pour leur part, Cohen et al. proposent différentes facettes pour les citations dans la description de tâche pour TAC 2014 : HYPOTHÈSE, MÉTHODE, RÉSULTATS, IMPLICATION, DISCUSSION, et DONNÉES [4]. Ils proposent de construire un résumé pour chaque facette. Afin de construire le résumé à l'aide de l'information trouvée, nous devrons choisir certaines phrases parmi les citations en évitant la redondance et choisir l'ordre dans lequel les placer. Finalement, le résumé sera amélioré à l'aide d'extraits du texte original. Dans ce document nous allons utiliser une notation dérivée de celle utilisée pour TAC 2014 (voir figure 1.1), une compétition de résumés d'articles en biologie à laquelle nous avons participé.

U : une collection de documents.

RP : (reference paper) le document à résumer.

CP^i : (citing paper) les documents appartenant à U autres que RP, où $1 \leq i \leq N$ et N est le nombre de documents dans U. Un document est un ensemble d'extraits.

S : (summary) le résumé de RP.

e^i_j : extraits des segments de texte de CP^i, où $1 \leq j \leq n^i$ et n^i est le nombre d'extraits dans CP^i. Un extrait est une phrase ou sous-phrase contenant une idée complète. ($e^i_j \in \text{CP}^i$).

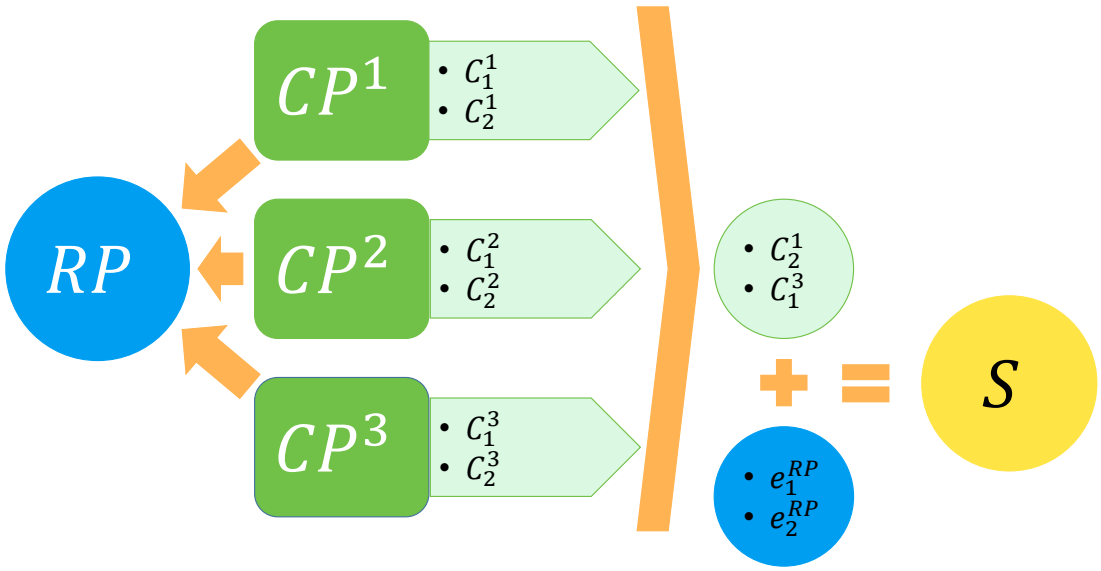

Figure 1.1 – Chaîne de traitement

c_j^i : les citances du document CP^i, où $1 \leq j \leq n_c^i$ et n_c^i est le nombre de citations du document CP^i, certaines faisant référence à RP. Une citance est un petit ensemble d'extraits ($c_j^i = \{e_k^i\}$).

Nous proposons d'utiliser un lexique spécialisé afin que notre système soit utilisable, peu importe le domaine des articles à résumer. Aussi, nous allons appliquer une technique de Maximal Marginal Relevance pour construire notre résumé à l'aide d'un mélange d'extraits de l'article et des citations vers cet article. Finalement, nous allons construire une interface permettant la consultation simultanée d'un article, son résumé et les articles le citant.

Nous avons construit un système complet (Figure 1.1). À partir d'un corpus d'articles composé d'un article de référence RP et d'articles qui le citent CP^i, `Citatum` va construire un résumé de RP.

Ce système (Figure 1.2) va transformer ces articles en deux bases de données. Une base de données sera constituée des articles modifiés en fichiers sous format JATS/XML, un format construit pour représenter l'information contenue dans les articles scientifiques. L'autre base de données va contenir les méta-informations liées

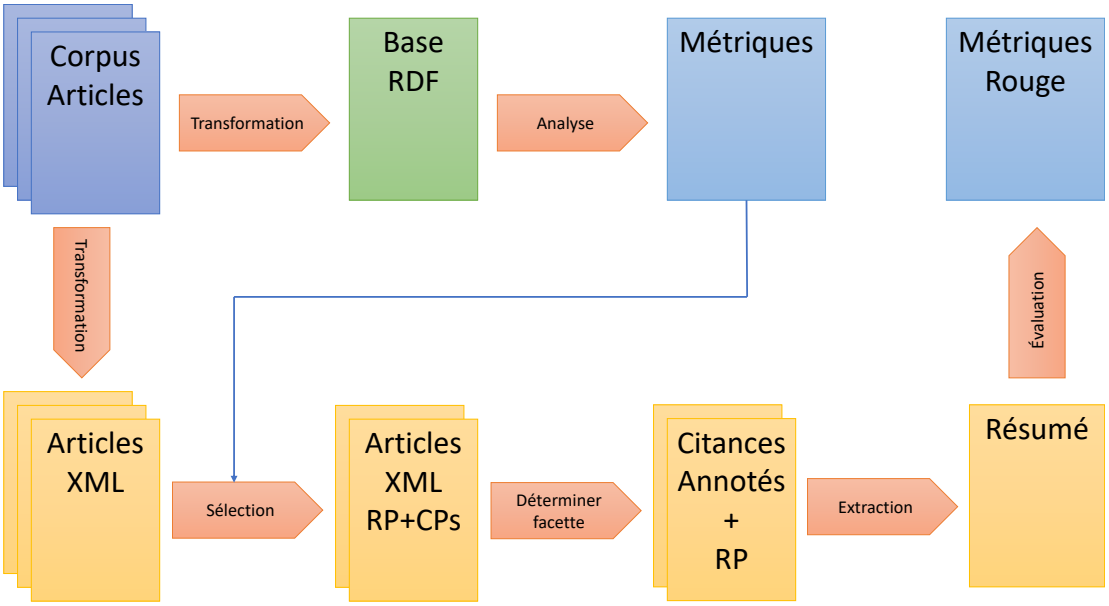

Figure 1.2 – Le chemin d'information de `Citatum`

aux articles. Cette information sera représentée en format RDF. Ce format fait partie du *Web Semantic* et représente l'information sous forme de relation. Cela va nous permettre de facilement construire des requêtes pour consulter la méta-information. Le chapitre 3 décrit les ensembles de données que nous avons utilisés. Aussi, il décrira les transformations effectuées par notre système pour construire les deux bases de données.

Ensuite, notre système analyse notre base de méta-information, plus particulièrement, le graphe de citation. Cette analyse nous permet, entre autres, d'extraire les articles qu'il serait le plus intéressant à résumer. Nous présentons comment cette analyse est faite au chapitre 4. Nous y présentons les techniques utilisées pour consulter nos bases de données à l'aide des langages SPARQL et Python. Aussi nous allons montrer comment nous avons reconstruit des métriques proposées par l'AAN. `Citatum` est loin d'être le premier à faire l'analyse d'un graphe de citation. Nous nous distinguons par l'inclusion de cette analyse dans un contexte plus grand : cette analyse est utilisée pour la sélection d'articles qui seront résumés par le même système.

Lorsque l'article à résumer et les articles le citant ont été sélectionnés, notre système doit traiter les citances pour les préparer à l'étape de résumé. Ce traitement

a été le sujet des compétitions TAC et CL. Nous devons déterminer la facette de chaque citance et trouver une phrase de l'article cité qui corresponde à la citation. Ces algorithmes et résultats sont présentés au chapitre 5. Bien que la détermination du contexte rhétorique d'une phrase soit un problème étudié par plusieurs équipes, pour notre système, ce résultat sera potentiellement utilisé pour construire des résumés d'articles dirigés vers des sujets plus précis : méthode, résultat, implication, hypothèse et objectif.

Ensuite, un résumé de l'article est construit à l'aide de l'information obtenue lors des étapes précédentes. Ce résumé est constitué de phrases provenant des citances et de phrases extraites de l'article résumé. Nous utilisons l'algorithme de Maximal Marginal Relevance qui était originalement conçu pour chercher de l'information à partir d'une requête. Cet algorithme va nous permettre d'éliminer la redondance dans le résumé final. Le chapitre 6 présente notre technique plus en détail. Le résumé que nous construisons diffère des résumés habituellement proposés par d'autres chercheurs [10, 28, 30]. Notre résumé inclut à la fois l'information venant des citations et l'information extraite de l'article. De tels résumés contiendront de l'information sur l'impact que l'article a eu sur la communauté complémentée par de l'information de l'article.

Finalement, notre système construit une interface HTML pour présenter les résultats et construit les fichiers sources pour permettre l'évaluation des résumés à l'aide des métriques ROUGE-2 et 4. Ces résultats seront présentés au chapitre 7.

Le chapitre suivant (Chapitre 2) présente l'état de l'art dans le domaine du traitement automatique de la langue appliqué aux textes scientifiques. Nous y présenterons aussi les techniques que notre système emprunte aux systèmes existants et les techniques que nous avons adaptées et modifiées.

CHAPITRE 2

REVUE DE LITTÉRATURE

Ce chapitre explore différentes technologies développées dans le domaine de l'analyse de la langue naturelle liées au sujet de notre doctorat. Puisque nous voulons construire un résumé à partir des citations de plusieurs articles, nous allons porter notre attention sur l'extraction des citations et l'automatisation de résumés d'articles multiples. Dans un premier temps, un rappel sur quelques principes de base du traitement des langues naturelles est donné (section 2.1). Suivra une section sur les différentes études sur la structure des articles scientifiques (section 2.2) seront présentées. Ensuite, les méthodes de résumé d'article seront présentées (section 2.3). Suivra une section sur l'extraction de citations et de leurs contextes (section 2.4). Finalement, quelques méthodes pour les résumés d'articles multiples seront décrites (section 2.5).

2.1 Principes de base

Une des caractéristiques importantes du traitement de la langue naturelle est la nature des données en entrées. Contrairement à d'autres logiciels, l'information fournie au système ne suit pas des règles strictes sur sa présentation et sa composition. Les systèmes construits doivent tenir compte de l'ambiguïté et la diversité des constructions linguistiques humaines.

Cette caractéristique impose deux idées de base. Premièrement, il est important d'utiliser des corpus d'information non synthétiques, mais plutôt de rechercher des données réelles, venant de l'extérieur. Au chapitre 3 nous présenterons le corpus que nous avons utilisé. Deuxièmement, les tests ne pourront pas donner des résultats exacts et précis. Il faut quand même évaluer les performances de nos systèmes. Des métriques simples, comme le rappel (pourcentage de l'information recherchée qui a été trouvée) et la précision (pourcentage de l'information trouvée qui est correcte) permettent d'évaluer certains systèmes d'extractions. Pour des besoins plus complexes, des métriques appropriées sont développées. L'appendice I présente la métrique ROUGE que nous avons utilisée pour évaluer nos résumés.

Cette imprécision des données implique souvent l'utilisation de statistiques. Les métriques TF $\cdot$ IDF sont importantes et à la base de plusieurs algorithmes présentés dans cet ouvrage. La métrique TF (Term Frequency) indique la fréquence d'un mot (terme) à l'intérieur d'un document. Lorsqu'un mot a une fréquence élevée dans un document, cela indique qu'il est important dans le contexte de ce document. Par exemple, une requête contenant ce mot devrait trouver les documents contenant une fréquence élevée de ce mot. Par contre, cette métrique ne tient pas compte des mots fréquents dans la majorité des documents. C'est pour pallier ce problème que la métrique IDF (Inverse Document Frequency) a été introduite. Pour un mot donné, cette métrique calcule l'inverse multiplicatif du nombre de documents où il apparait. Ainsi, un mot qui est présent dans seulement quelques documents sera plus important qu'un mot étant dans la majorité des documents.

La plupart des systèmes de résumé automatique sont développés pour résumer des textes de nouvelles. Dans notre travail, nous allons analyser des textes scientifiques. Les textes de nouvelles ont une structure particulière. Au début ils mentionnent l'événement le plus important, ensuite, chaque section approfondit différents éléments de la nouvelle. Cela leur donne une structure pyramidale, où chaque section est un étage supplémentaire qui développe le précédent.

Pour les discours scientifiques, la structure est dirigée par le besoin de promouvoir les recherches de l'auteur. C'est une suite d'arguments contenant des citations des autres articles scientifiques, positive ou négative. La section suivante donne plus d'information sur cette structure.

2.2 Structure d'un article scientifique

> Ensuring coherence is difficult, because this in principle requires some understanding of the content of each passage and knowledge about the structure of discourse.[29]

Dans son étude, Teufel [37] énumère différents contextes rhétoriques qui peuvent être attribués à une phrase dans un article scientifique. Par contexte rhétorique d'une phrase, nous entendons sa fonction communicationnelle dans le contexte de l'article scientifique. Le terme facette est utilisé pour désigner cet attribut qu'est le contexte rhétorique d'une phrase (Par exemple : le but, les méthodes, les bases scientifiques). Les phrases extraites d'un texte perdent leur contexte, elles ne sont

plus entourées des autres phrases. Le rôle d'une facette est de pallier à ce manque en ajoutant de l'information sur le contexte rhétorique de la phrase. Simone Teufel détecte les éléments suivants dans un article : la structure d'un problème (l'objectif de la recherche, les méthodes utilisées pour résoudre le problème et les résultats), les attributs intellectuels (ce qui est nouveau dans l'article comparé aux anciennes découvertes), la valorisation scientifique (argumentation pour faire accepter l'article et ces faits) et la perception du travail d'autrui (une approche contradictoire, qui contient une erreur ou qui aide les recherches présentes). En se basant sur ses observations, elle définit différentes facettes pour chaque phrase d'un article scientifique :

AIM : Les objectifs de l'article courant.

TEXTUAL : des indicateurs de structure du texte, section, sous-section ...

OWN : des descriptions de ce qui est présenté dans l'article : méthodologie, résultat et discussion qui ne sont pas identifiés comme AIM ou TEXTUAL.

BACKGROUND : des descriptions des connaissances scientifiques acceptées dans le domaine.

OTHER : citations décrivant les réalisations d'un autre chercheur.

CONTRAST : éléments faibles/différents de la recherche citée.

BASIS : éléments acceptés des autres recherches qui serviront de fondement pour la recherche présente.

Pour les citations, ce sont les facettes OTHER, CONTRAST et BASIS qui vont être les plus importants.

Dans leur article, Teufel et Moens [38] présentent une méthode pour extraire le contexte rhétorique des phrases d'un texte scientifique. Pour cela, ils utilisent un classifieur naïf de Bayes basé sur des articles annotés manuellement. Les caractéristiques suivantes sont utilisées par le classifieur :

- Position absolue et relative de la phrase.
- Longueur de la phrase et sa composition en termes significatifs. Utilisation de la mesure $\mathrm{TF} \cdot \mathrm{IDF}$ pour les termes.

- La syntaxe des verbes, leurs temps.
- Le type de la citation.
- La catégorie de la phrase précédente.
- Sorte d'agent et d'action.

L'information résultante sert autant à construire un résumé [38] qu'à classifier les citations. Dans notre cas, elle sera utilisée pour classer les citations ce qui, avec leurs contextes, déterminera le rôle de chaque citation.

Si la citation parle de l'article à résumer, alors BASIS et OTHER indiquent une citation contenant de l'information qui est directement utilisable dans le résumé final. CONTRAST donnera de l'information sur ce qu'il est possible d'améliorer. Par contre, si la citation est dans l'article à résumer alors CONTRAST nous donne le sujet de l'article et BASIS nous donne le domaine dans lequel la recherche s'inscrit. Dans ce cas, OTHER ne contiens pas d'information pertinente.

Cela donne un ensemble de trois facettes pour l'ensemble des citations. Le *Citation Typing Ontology* (CiTO) propose un plus grand choix pour qualifier une citation. Construit par Shotton [35], c'est une ontologie pour décrire des citations scientifiques. Étant une ontologie, le CiTO s'inscrit dans le Web Semantique[1]. Bien que construit pour les articles en biologie, il est applicable sans problème à d'autres domaines. Les facettes proposées représentent la relation entre l'article citant et l'article cité. Par exemple, CONFIRMS indique que la citation confirme ce qui est dit dans l'article cité. Des relations décrivent la fréquence des citations vers l'article cité. Par exemple, CITO :GLOBALCITATIONFREQUENCY est une relation donnant le nombre de fois que l'article citant cite l'article référé. Finalement, l'ontologie contient des relations (comme CITO :PATENTDOCUMENT) qualifiant l'article cité. Cet ensemble de facettes est très complet, mais les facettes sont trop précises pour nos besoins puisque nous cherchons à construire un résumé par facette. Elles deviendraient intéressantes dans un contexte de génération de revue de littérature, où elles expliqueraient les différents liens entre l'information extraite des articles inclues.

Un ensemble différent de caractéristiques est proposé pour la compétition TAC 2014. Cohen et al. 2014 [4] utilisent des facettes pour décrire le rôle d'une citation. Leur objectif est de trouver l'extrait du document de référence qui est lié à la citation.

[1] `http://purl.org/spar/cito/`

Les facettes qu'ils proposent reflètent un tel choix : HYPOTHESIS, METHOD, RESULTS, IMPLICATION, DISCUSSION et DATA. Puisque nous avons participé à ces compétitions, cela nous a permis de recueillir des citations préannotées avec ces facettes. C'est pour cette raison que notre système va les utiliser. Ces compétitions utilisent des corpus d'articles provenant du domaine de l'informatique-linguistique. Nos expériences vont utiliser ces corpus. À plus long terme, il serait important et intéressant d'étudier et traiter des corpus issus d'autres domaines afin de déterminer comment les citations sont utilisées dans ces autres contextes.

2.3 Résumé d'article

Dans son article de 1999, Jones [36] propose un modèle abstrait décrivant l'automatisation des résumés. Cette abstraction comprend trois étapes :

I : Interprétation du texte source et transformation vers une représentation abstraite de la source.

T : Transformation de la représentation abstraite de la source vers une représentation abstraite du résumé.

G : Génération du résumé à partir de sa représentation abstraite.

Les prochaines sections décrivent les méthodes extractives et abstractives pour la construction automatique de résumé.

2.3.1 Méthodes extractives

Ces méthodes de résumé consistent à extraire l'information pertinente du texte et à construire le résumé sans transformer cette information. Différentes techniques sont utilisées pour trouver le texte à extraire.

Edmundson [9] propose l'utilisation d'une mesure d'importance associée aux phrases à extraire. Pour cela, il va extraire manuellement des phrases de plusieurs textes. Cette extraction est basée sur plusieurs critères : le sujet du texte, sa raison d'être, les méthodes utilisées, les conclusions, la généralisation et les implications, les

recommandations et les suggestions. Il va aussi minimiser l'interdépendance (ne pas avoir de phrases qui dépendent d'une autre) et maximiser la cohérence.

Son système d'extraction construit quatre métriques.

Cue : des points sont donnés aux phrases qui contiennent des mots clefs identifiés au préalable. Ces mots clefs ont été choisis parmi les phrases extraites manuellement. Il y a 783 mots à contribution positive et 73 mots à contribution négative. Il a aussi identifié 139 mots qui ont une contribution nulle, ils seront utilisés pour les autres métriques.

Key : utilise la fréquence des mots dans le texte. Les mots les plus fréquents donnent des points. Les mots clefs identifiés par la première méthode sont exclus.

Title : donne des points aux mots qui apparaissent dans les entêtes de section et dans le titre. Les mots à contribution nulle sont exclus.

Location : cette métrique se base sur la position des mots. Les mots au début (dans l'introduction) et à la fin (dans la conclusion) reçoivent une contribution positive. De plus, les phrases du premier et dernier paragraphe ainsi que la première et dernière de chaque paragraphe reçoivent des points.

Une somme pondérée est calculée pour chaque phrase et celles ayant les valeurs les plus élevées sont choisies. Après quelques tests, il a conclu que la métrique *Key* n'aidait pas à obtenir de meilleurs résultats.

Kupiec, Pedersen et Chen utilisent des principes similaires pour construire un extracteur auquel il vont ajouter l'apprentissage machine [18]. Leur système calcule la probabilité qu'une phrase appartienne à un résumé en fonction des propriétés suivantes :

- Nombre de mots dans la phrase. C'est une valeur booléenne qui devient vraie si le nombre de mots est supérieur à un seuil.
- Une valeur booléenne indiquant si la phrase contient un des 26 segments de phrase préidentifiés.
- Pour les dix premiers paragraphes et les cinq derniers, chaque phrase reçoit un attribut indiquant si elle est au début, à la fin, ou au milieu du paragraphe.

- Les mots les plus fréquents (thématiques) sont identifiés.
- Les mots de plus d'une lettre, qui ne commencent pas une phrase et qui commencent par une majuscule sont identifiés. Seuls ceux apparaissant plus d'une fois dans le texte sont retenus.

Une probabilité est calculée pour chaque phrase. Cette probabilité indique les chances qu'a une phrase (e_j^s) d'apparaître dans le résumé final.

$$P(e_j^{\mathrm{RP}} \in S|F_1, F_2, \ldots F_m) \quad = \quad \frac{\Pi_{i=1}^m P(F_i|e_j^{\mathrm{RP}} \in S) P(e_j^{\mathrm{RP}} \in S)}{\Pi_{i=1}^m P(F_i)}$$

où les F_i représentent les m propriétés. Les $P(e_j^{\mathrm{RP}} \in S)$ sont des constantes, $P(F_i|e_j^{\mathrm{RP}} \in S)$ et $P(F_i)$ sont estimés à partir de l'ensemble d'entrainement.

Conroy et O'Leary [5] ont proposé une méthode utilisant les chaînes de Markov pour décider si une phrase devrait figurer dans un résumé en se basant sur la précédente. Leur modèle utilise les facteurs suivants pour classer une phrase :

- La position de la phrase dans le document.
- Le nombre de termes dans la phrase. Ils utilisent le logarithme du nombre de mots dans la phrase.
- La chance qu'un terme soit dans une phrase i spécifique s'il est dans le document.

Les trois méthodes que nous venons de décrire (Edmundson [9], Kupiec, Pedersen et Chen [18], Conroy et O'Leary [5]) permettent l'extraction d'information importante. Par contre elles résument un seul document. Aucune de ces méthodes n'élimine la redondance.

Filatova et Hatzivassiloglou [11] recherchent les événements atomiques dans un texte (ou un ensemble de textes). Ensuite, le résumé est construit à l'aide des éléments les plus mentionnés. Une liste d'événements atomiques est affectée à chaque phrase du texte. Un événement atomique est un lien entre les constituants d'une action. Les actions sont représentées ou décrites par les verbes. En général, les constituants

principaux sont représentés par des entités nommées. Il ne reste plus qu'à trouver les événements atomiques les plus fréquents.

Pour trouver la couverture maximale des événements, ils utilisent un algorithme vorace :

1. Calculer le poids de chaque unité textuelle en sommant le poids des concepts qu'ils couvrent.
2. Trouver les unités textuelles de plus grand poids qui ne sont pas encore dans la solution. Choisir celle avec le plus grand poids. L'ajouter à la solution et ajouter les concepts à la liste des concepts couverts.
3. Recalculer le poids des unités textuelles en enlevant le poids des concepts déjà couverts.
4. Si la longueur du résumé voulue n'est pas atteinte alors retourner à l'étape 2.

Parveen et Strube [25] construisent un résumé à l'aide d'un graphe. Ils utilisent un graphe biparti, d'un côté les noeuds représentent des phrases et de l'autre ils représentent des entités. Un arc existe entre une phrase et une entité si l'entité est mentionnée dans la phrase.

Le graphe est ensuite utilisé pour calculer une valeur de cohérence. La cohérence d'une séquence de phrases est augmentée si elles partagent des entités. Le résumé est finalement construit en choisissant la séquence de phrases qui optimise la cohérence du résultat.

Les deux dernières techniques présentées (Filatova et Hatzivassiloglou [11], Parveen et Strube [25]) sont intéressantes. Elles utilisent des manipulations plus avancées sur les données, soit en les divisant en éléments plus petits [11], soit en les ajoutant de l'information sur les entitées [25]. Pour notre projet, nous voulons un algorithme qui travaille sur un grand nombre d'articles. Nous cherchons donc des techniques plus légères sur l'utilisation des ressources. À plus longue échelle, il serait intéressant d'insérer de telle analyse dans notre projet.

2.3.2 Méthodes abstractives

Dans la section précédente, nous avons vu quelques techniques de résumé qui cherchent des phrases dans un texte, les extraient et construisent un résumé contenant ces phrases. Les techniques de résumé par abstraction vont construire de nouvelles phrases à partir de l'information extraite. Cela implique une manipulation syntaxique (section 2.3.3) ou sémantique du texte choisi. Le résumé sera donc construit à l'aide de concept tirés du texte original qui seront présentés différemment.

Genest et Lapalme [12] construisent une représentation intermédiaire du texte afin d'analyser sa sémantique. Chaque phrase du texte est transformée en un graphe où les arcs représentent la dépendance entre les mots d'une phrase. Leur objectif est d'extraire des informations précises sur un événement : lieu, victime, etc.. Afin de trouver l'information la plus pertinente possible, ils utilisent un système de règles logique. Ces règles sont des patrons à remplir en utilisant les graphes représentant les phrases. Cette sélection permet d'éliminer la redondance et d'extraire l'information la plus pertinente. Finalement, les patrons permettent aussi la génération des phrases composants le résumé qui sera complété par des phrases extraites du texte.

Dans le contexte de cette recherche, nous allons nous concentrer sur les méthodes extractives, les méthodes sémantiques demandant un ensemble différent de manipulations.

2.3.3 Manipulation syntaxique

Knight propose une manipulation afin de réduire la longueur des phrases d'un texte [17]. Il transforme le texte en un arbre et transforme les arbres afin d'obtenir des arbres plus petits. Les transformations sont dirigées par un système stochastique dont les valeurs sont trouvées à l'aide de phrases déjà réduites. Il considère les phrases longues comme ayant été construites à partir de phrases plus courtes et tente de trouver comment elles ont été construites afin de retrouver les phrases de départ.

Dans un premier temps, son système construit des arbres représentant les phrases longues. Ensuite, plusieurs petits arbres qui pourraient représenter la phrase d'origine sont construits. Le système reconstruit l'arbre de la grande phrase en appliquant des transformations sur les arbres des plus petites phrases. Les transformations sont basées sur des modèles qui contiennent un noeud de l'arbre et ses enfants. Chaque résultat est classé et le meilleur est choisi.

Pour leur part, Mani, Gates et Bloedorn [21] proposent une technique où un résumé de départ est amélioré afin d'obtenir le résumé final. Ils utilisent des règles qui enlèvent et ajoutent de l'information au résumé [6]. Ils transforment le texte en un arbre syntaxique. Pour construire le résumé de base, ils vont associer à chaque phrase une valeur indiquant son importance. Ensuite, ils choisissent les phrases de plus haute importance jusqu'à ce qu'ils aient atteint le taux de compression désiré. Lorsque le résumé de départ est terminé, ils appliquent chaque règle aussi souvent qu'ils le peuvent. Le programme se termine lorsqu'il n'y a plus de règles à appliquer ou si le résumé a atteint la taille désirée.

Il y a trois types de règles de révision :

1. Éliminer des sections d'une phrase : parenthèses, un mot comme *In particular*, *Accordingly* et *In conclusion* au début d'une phrase.
2. Combiner deux sections de phrases différentes. Dans ce cas une section est déjà dans le résumé et l'autre n'y est pas encore. Les groupes de verbes de deux phrases sont combinés si elles ont une co-référence en commun dans leurs groupes noms.
3. Modifier une phrase pour l'améliorer. Ce sont des modifications de style dans la phrase. Ces modifications permettent de réduire la taille d'une phrase et d'améliorer son style. À la fin de ces opérations, un dernier type de modification est appliqué. Ces dernières modifications consistent à améliorer la cohérence des phrases du résumé. Cela consiste à faire des références à des éléments antérieurs. Par exemple, changer un nom propre pour un alias si le nom apparait dans une phrase antérieure.

2.4 Extraction des citations et références

Dans le champ de l'analyse d'information scientifique, nous nous sommes intéressé à l'analyse des citations et des co-citations. L'indexation automatique des citations a été proposée par Garfield en 1965. Cette indexation fut inspirée par l'indexation des ressources légales que Frank Shepard a initié en 1873.

Wan et.al. [39] ont consulté des chercheurs pour comprendre leurs besoins lors de la recherche d'articles scientifiques(2009). Ils en sont venus à la conclusion qu'il y avait deux éléments clefs pour assister le lecteur lorsqu'une citation est rencontrée :

les métadonnées et un pré-visionnement contextuel. Ces informations ont pour but d'aider le chercheur à déterminer si l'article cité devrait être consulté.

2.4.1 Extraction et segmentation des références

Un premier champ d'intérêt dans le domaine est l'extraction des références bibliographiques. L'activité consiste à extraire chaque référence et à les diviser en composantes (auteurs, titre, année ...).

Besagni, Belaïd, et Benet [1] présentent une méthode basée sur la reconnaissance des *parties d'un discours*. Ils traitent des articles numérisés et sauvegardés en XML et identifient différents mots du texte selon les caractères les composant (lettre, chiffre, case ...). Ensuite, ils utilisent des patrons pour identifier les différentes sections d'une référence. Une fois certaines sections identifiées, le système peut déduire les sections restantes.

Qazvinian et Radev [27] ont développé un système où chaque phrase est représentée par une variable aléatoire indiquant si la phrase appartient à une citance. Ils vont utiliser des champs aléatoires de Markov comme modèle de graphe pour représenter les liens entre les variables aléatoires. Leur graphe lie les variables entre elles lorsque les phrases du texte sont voisines. Un algorithme de propagation des croyances (*Belief Propagation*) à travers le graphe fait converger les valeurs (pointages).

2.4.2 Extraction des références/citations simultanément

Afin d'augmenter la précision de l'extraction des citations et des références dans un article scientifique, Powley et Dale proposent un système qui utilise l'information partielle d'une tâche afin d'aider l'autre, ainsi l'extraction des citations utilise l'information des références et vice-versa [26]. Leur système commence par extraire les citations en cherchant des nombres pouvant représenter des années. Ils vont aussi extraire tous les mots précédant l'année et qui commencent par une lettre majuscule. Ensuite, le système parcourt la section des références et cherche les mots commençant par une majuscule et les compare à ceux trouvés dans les citations, cela leur permet d'identifier les noms d'auteurs. Finalement, les groupes de noms identifiés dans la section des références avec leurs années de publication sont utilisés pour séparer chaque référence.

Pour le projet présent, c'est cette technique qui sera utilisée. Elle nous donne toute l'information dont nous avons besoin sans demander de manipulation complexe.

2.4.3 Site internet avec des systèmes d'extraction de citations

CiteSeer utilise des logiciels d'exploration (*crawlers*) du web pour trouver des articles scientifiques. CiteSeer extrait les références et les citations de l'article. Les citations sont accompagnées de leurs contextes.

Un objectif important de CiteSeer est d'être entièrement automatique (sans intervention humaine) et de pouvoir intégrer les nouveaux articles le plus tôt possible [13]. Lorsque l'information a été extraite, il faut associer les articles similaires à l'aide d'une mesure de similarité.

La première métrique utilise la construction d'un vecteur représentant chaque document. Les valeurs des composantes de chaque élément d'un vecteur sont des calculs de poids pour les 20 termes les plus importants du document ($\mathrm{TF} \cdot \mathrm{IDF}$). La distance entre deux documents devient le produit scalaire entre les vecteurs les représentant. Plus précisément, le poids associé à un lemme t_a pour un document CP^i est calculé comme suit :

$$\begin{aligned} w(t_a, \mathrm{CP}^i) &= \frac{\left(0.5 + 0.5 \frac{\mathrm{tf}(t_a, \mathrm{CP}^i)}{\mathrm{tf}(t_{\max}, \mathrm{CP}^i)}\right) \mathrm{idf}_a}{\sqrt{\sum_{t_j \in \mathrm{CP}^i} \left(\left(0.5 + 0.5 \frac{\mathrm{tf}(t_j, \mathrm{CP}^i)}{\mathrm{tf}(t_{\max}, \mathrm{CP}^i)}\right)^2 \left(\mathrm{idf}_j\right)^2\right)}} \\ \mathrm{idf}_j &= \log \frac{N}{\mathrm{df}_a} \end{aligned}$$

où

$\mathrm{tf}(t_j, \mathrm{CP}^i)$: est la fréquence du radical t_j dans le document CP^i.

$\mathrm{tf}(t_{\mathbf{max}}, \mathrm{CP}^i)$: est la plus grande fréquence d'un terme dans le document CP^i.

df_j : est le nombre de documents contenant le radical t_j.

Une deuxième mesure utilisée est la distance *LikeIt* entre chaînes de caractères. Elle est utilisée pour comparer les entêtes des articles. L'entête d'un article est le texte avant la section *Abstract* (résumé signalétique). *LikeIt* mesure la similarité entre une chaîne de caractères et un ensemble de chaînes de caractères afin de trouver des chaînes presques identiques malgré leurs différences. Cette technique utilise trois filtres successifs afin de réduire l'ensemble de départ et obtenir un sous-ensemble de chaînes similaires [44].

Enfin pour pouvoir suggérer des documents similaires à celui choisi par un utilisateur, CiteSeer utilise une dernière métrique : CC · IDF (Common Citation x Inverse Document Frequency). Pour cela, un poids est associé à chaque citation. Ce poids est l'inverse multiplicatif de la fréquence de la citation dans la base de données. Ensuite, pour chaque citation d'un document, il calcule l'ensemble de documents qui ont une citation commune. La similarité d'un document avec un autre devient la somme des poids des citations partagées. Pour avoir une métrique unique, la somme pondérée des trois métriques est effectuée.

Dans le cadre de cette recherche, c'est le résultat obtenu sur ces sites qui est important. Il est possible de les utiliser pour extraire les articles citant et leurs citations pour les besoins de nos expérimentations. Nous voulons concentrer notre travail sur le résumé plutôt que sur l'extraction de citations.

2.5 Résumé d'articles multiples

Puisque nous voulons construire le résumé d'un article à partir des citations extraites de plusieurs autres articles, nous utiliserons des techniques similaires au résumé d'article multiple. Le résumé d'articles multiples apporte trois problèmes supplémentaires à résoudre : la redondance, l'identification des différences importantes entre les documents et la cohérence des résumés alors que l'information vient de plusieurs sources [29].

Carbonell et Goldstein [3] proposent une technique pour extraire l'information d'articles multiples avec plusieurs répétitions. Dans une collection U de documents, leur système va permettre de répondre à une requête Q. Ils vont calculer une métrique, *maximal marginal relevance* (MMR), pour chaque phrase. Cette métrique calcule la pertinence d'une phrase par rapport à une requête.

$$\text{MMR} \stackrel{\text{def}}{=} \text{Argmax}_{\text{CP}^i \in R \setminus V} \left[\lambda(\text{Sim}_1(\text{CP}^i, Q) - (1-\lambda) \max_{\text{CP}^j \in V} \text{Sim}_2(\text{CP}^i, \text{CP}^j)) \right]$$

où $R = IR(U, Q, \theta)$ est la liste des documents extraits par un système d'extraction d'information, θ est un seuil pour choisir les documents pertinents, V est l'ensemble des documents de R déjà présentés à l'utilisateur et Sim_i une métrique de similarité entre deux documents. Dans leurs tests, ils ont utilisé le cosinus comme métrique de similarité. Le deuxième terme de l'équation $((1-\lambda) \max_{\text{CP}^j \in V} \text{Sim}_2(\text{CP}^i, \text{CP}^j))$ est soustrait. Ce terme compare une phrase candidate avec les phrases déjà incluses dans le résumé. Le pointage de la phrase la plus similaire est soustrait, pénalisant les phrases redondantes au résumé déjà construit.

White et Cardie proposent une procédure de recherche locale pour choisir un ensemble de phrases en optimisant la somme des rangs de chaque phrase [41]. Leur algorithme favorise l'inclusion de phrases adjacentes et défavorise les phrases répétitives. Afin de détecter les phrases similaires (répétitives), ils utilisent une mesure générée par l'outil SimFinder [14].

Dans un premier temps, leur système extrait l'information des textes originaux pour les représenter en XML avec des annotations sur les rôles des parties d'une phrase. Ensuite, les sorties de la première analyse sont regroupées en structures d'événements où les faits sont comparables. Une valeur est assignée à chaque phrase indiquant la position de la phrase dans le document, la date de publication, la présence de guillemets, la moyenne de la similarité avec chaque phrase. Aux groupes sémantiques est assignée une valeur représentant la somme des valeurs de chaque phrase du groupe. Ces valeurs sont normalisées et servent ensuite à augmenter la valeur de chaque phrase qui est dans un groupe plus important qu'un autre.

Un aperçu est construit en choisissant un ensemble de phrases. Pour cela, ils utilisent une recherche aléatoire locale. La somme des valeurs de ces phrases est construite et à cette somme est appliquée une pénalité lorsque des phrases similaires sont dans l'ensemble. Un bonus est ajouté à une phrase et à sa précédente si elle commençait par un pronom ou un marqueur rhétorique important.

Ensuite, l'algorithme extrait une phrase des textes et en élimine possiblement quelques-unes du résumé pour revenir à la taille voulue pour le résumé. Il y a deux façons de choisir une phrase, soit avec un saut aléatoire, soit avec un algorithme vorace.

Le choix entre les deux est aléatoire. Cela est répété jusqu'à ce que l'algorithme vorace ne réussisse pas à trouver une phrase qui augmente la valeur totale de l'ensemble. Finalement, le résumé en format hypertexte est construit à partir de l'ensemble résultant.

Dans leur article, Radev, Jing, Sty et Tam [30] présentent une méthode basée sur les centroïdes. Un centroïde est un ensemble de mots qui sont statistiquement importants pour un ensemble de documents. Chaque document est représenté par un vecteur de valeurs de poids TF · IDF. L'outil CIDR calcule un centroïde à partir du premier document (CP^1) de l'ensemble de documents (U). Ensuite, les valeurs TF · IDF sont calculées pour chaque document et sont comparées à celle du centroïde en utilisant la formule suivante :

$$\mathrm{sim}(\mathrm{CP}^i, \mathrm{CP}^1) = \frac{\sum_{t_j \in U} (\mathrm{tf}(t_j, \mathrm{CP}^i) \times \mathrm{tf}(t_j, \mathrm{CP}^1) \times \mathrm{idf}_j)}{\sqrt{\sum_{t_j \in U} (\mathrm{tf}(t_j, \mathrm{CP}^i))^2} \sqrt{\sum_{t_j \in U} (\mathrm{tf}(t_j, \mathrm{CP}^1))^2}}$$

Si la mesure de similarité est inférieure à un certain seuil alors le document est ajouté à l'ensemble. À chaque document ajouté à un centroïde, le logiciel recalcule la valeur IDF de l'ensemble. Leur hypothèse est qu'une phrase qui contient des mots d'un centroïde parle d'un sujet similaire au reste de l'ensemble.

Ensuite, trois métriques sont calculées pour mesurer la qualité d'une phrase.

1. La valeur centroïde C^i d'une phrase e^i_j est la somme des centroïdes de ses mots $t_k \in e^i_j$.

$$C^i(e^i_j) = \sum_{t_k \in e^i_j} C^i(t_k)$$

2. Une valeur est attribuée à chaque phrase selon sa position dans le document. La première phrase du document aura une valeur égale à la valeur la plus élevée parmi les phrases du document $P(e^i_1) = \max\{C^i(e^i_j) | e^i_j \in \mathrm{CP}^i\}$). Les valeurs des autres phrases seront calculées comme suit :

$$P(e^i_j) \quad = \quad \frac{n^i - j + 1}{n^i} \times P(e^i_1)$$

où n^i est le nombre de phrases du document CP^i.

3. Une valeur représentera la similarité entre chaque phrase et la première phrase du document. Pour cela, le produit scalaire entre les vecteurs représentant chaque phrase est calculé. Chaque élément du vecteur représente un mot du document. Le vecteur représentant une phrase contient à chaque position le nombre d'occurrences de ce mot dans la phrase (représentation en sac de mots).

$$F(e^i_j) \quad = \quad \vec{e^i_1}\vec{e^i_j}$$

La métrique totale associée à chaque phrase est une somme pondérée des trois valeurs précédentes :

$$\mathrm{score}(e^i_j) \quad = \quad w_c C(e^i_j) + w_p P(e^i_j) + w_f F(e^i_j)$$

Ensuite, une pénalité est donnée aux phrases (e^i_a) qui ont une section commune avec une phrase qui a une valeur plus élevée (e^i_b). La pénalité due aux répétitions est calculée comme suit :

$$\begin{aligned} \mathrm{score}'(e^i_a) &= \mathrm{score}(e^i_a) - w_r R(e^i_a) \\ w_r &= \max_{e^i_j \in \mathrm{CP}^i} (\mathrm{score}(e^i_j)) \\ R(e^i_a) &= 2 \times \frac{|\mathrm{words}(e^i_a) \cap \mathrm{words}(e^i_b)|}{|\mathrm{words}(e^i_a)| + |\mathrm{words}(e^i_b)|} \end{aligned}$$

Finalement, un nouvel ensemble de phrases est extrait à partir des phrases ayant la plus haute valeur. Le procédé est répété jusqu'à ce qu'il n'y ait plus de nouvelles phrases extraites.

Erkan utilise des principes similaires [10]. Il va construire une métrique de *centralité* pour chaque phrase dans un regroupement et extraire la plus importante pour le résumé. Le résumé sera composé des phrases les plus importantes de chaque regroupement. Il commence par construire une représentation par sac de mots pour chaque phrase. La similarité sera mesurée par le cosinus des deux vecteurs.

$$\widehat{idf}(\vec{e_a^i}, \vec{e_b^i}) = \frac{\sum_{t_w \in e_a^i, e_b^i} \mathrm{tf}(t_w, e_a^i) \times \mathrm{tf}(t_w, e_b^i) \times (\mathrm{idf}_w)^2}{\sqrt{\sum_{t_i \in e_a^i} (\mathrm{tf}(t_i, e_a^i) \times \mathrm{idf}_i)^2} \times \sqrt{\sum_{t_i \in e_b^i} (\mathrm{tf}(t_i, e_b^i) \times \mathrm{idf}_i)^2}} \quad (2.1)$$

où $\mathrm{tf}(t_w, e_j^i)$ est le nombre d'occurrences du mot t_w dans la phrase e_j^i. Ensuite, un graphe est construit en utilisant les phrases comme noeuds. Pour chaque paire de phrases dont la similarité est supérieure à un certain seuil, un arc est ajouté avec l'indice de similarité comme poids. Il calcule la *centralité* pour chaque phrase du graphe en utilisant cette formule qui tient compte du nombre de phrases adjacentes ($adj[e_b^i]$) et de leurs qualités (*centralité* :$p(e_a^i)$).

$$p(e_a^i) = \frac{d}{K} + (1-d) \sum_{e_b^i \in adj[e_a^i]} \frac{\widehat{idf}(e_a^i, e_b^i)}{\sum_{e_c^i \in adj[e_b^i]} \widehat{idf}(e_c^i, e_b^i)} p(e_b^i) \quad (2.2)$$

où d est utilisé pour ajuster les valeurs et K est le nombre de noeuds du graphe.

2.5.1 Analyse des citations

Qazvinian, et al. [28] proposent une technique utilisant les citations et le résumé signalétique d'un article pour construire un résumé. Ils modélisent l'ensemble des phrases citant un article choisi comme le graphe des citations de l'article (Citation Summary Network). Les arcs de ce graphe vont être décorés par une valeur de similarité des citations. Ils utilisent quatre méthodes pour choisir les phrases pour les résumés à partir de ce graphe : C-LexRank, C-RR, LexRank et MASCS. Ensuite, ils vont comparer les résultats avec un résumé écrit par un humain et un autre composé de phrases choisies aléatoirement. Ils utilisent trois sources d'informations différentes par article :

- Un premier résumé à partir de l'article complet.
- Un autre utilisant le résumé signalétique.
- Un dernier à partir des citations.

Ils en concluent que les citations et les résumés signalétiques contiennent plus d'information unique et utile que le corps de l'article.

La construction de résumé que nous proposons poursuit ces travaux qui ont montré que les citations d'un article contiennent de l'information pertinente pour construire un résumé de l'article cité. Cette information peut être obtenue assez simplement, car plusieurs sites internet contiennent des articles pour lesquels les références ont déjà été extraites (Citeseer, Google Scholar et Microsoft Academic Search)(section 2.4). Nous allons donc utiliser les citations venant de plusieurs articles pour construire le résumé de l'article cité. Pour éviter de traiter des phrases complexes (contenant plus d'une idée) certains auteurs réduisent la taille des phrases, soit en cherchant les événements atomiques ou en retrouvant la section centrale de la phrase(section 2.3.3).

Il faut ensuite construire le résumé à l'aide de l'information trouvée. Nous devrons choisir certaines phrases parmi les citations en évitant la redondance. Pour cela, plusieurs auteurs utilisent des métriques de similarité : MMR, utilisation de SimFinder, et mesure de centralité(section 2.5). Il est aussi proposé d'utiliser des règles afin d'améliorer un résumé(section 2.3.3). Le chapitre suivant va présenter notre proposition de recherche qui s'inspire de ces travaux.

2.6 Métrique mesurant la pertinence d'un article

Il existe déjà des métriques mesurant la pertinence ou l'importance d'un article scientifique. Brin et Page [2] ont construit une métrique permettant d'analyser les liens entre articles afin d'attribuer une valeur d'importance à l'article. Cette métrique assigne une valeur selon le nombre de citations que reçoit un article et selon l'importance des articles citant. Nous représentons les articles comme les noeuds (V) d'un graphe (G) et les citations comme étant les arcs (E) de ce graphe. Les arcs sont dirigés de l'article citant vers l'article cité. Définissons les deux fonctions suivantes :

- $\text{In}(V_i) = \{V_j | (V_j, V_i) \in E\}$

- $\text{Out}(V_i) = \{V_j | (V_i, V_j) \in E\}$

La valeur d'importance (PageRank) d'un article sera calculée de la façon suivante :

$$P(V_i) = \frac{(1-d)}{|V|} + d \times \sum_{j \in \text{In}(V_i)} \frac{1}{|\text{Out}(V_j)|} P(V_j) \tag{2.3}$$

où d est un facteur d'amortissement compris entre 0 et 1 (en général, $d = 0.85$ est choisi). Les valeurs d'importance sont recalculées jusqu'à ce qu'elles convergent. Il est aussi possible d'associer un poids (w_{ij}) aux arcs. Dans ce cas la formule devient :

$$W(V_i) = \frac{(1-d)}{|V|} + d \times \sum_{j \in \text{In}(V_i)} \frac{w_{ji}}{\sum_{V_k \in \text{Out}(V_j)} w_{jk}} W(V_j) \tag{2.4}$$

Nous venons d'explorer différentes technologies développées dans le domaine de l'analyse de la langue naturelle. Plusieurs des techniques de résumés présentées sont utilisées dans le domaine des articles de nouvelles. Dans notre cas, nous allons résumer des articles scientifiques. Un ensemble d'articles sur un sujet particulier représente un discours entre plusieurs auteurs. C'est une suite d'arguments dont la continuité est représentée par les citations entre articles.

Nous avons présenté les études de Simone Teufel [37] sur les constituants d'un article, plus particulièrement, l'annotation des citations et les contextes rhétoriques. Ces contextes ajoutent de l'information sur la direction que prend l'argumentation scientifique. Aussi, nous avons présenté différentes techniques de résumé automatique. Les premières techniques présentées ne tennaient pas compte de la répétition. Ensuite, nous avons présenté des techniques utilisant des manipulations plus avancées, Genest et Lapalme [12], Knight [17], Mani, Gates et Bloedorn [21]). Ces techniques, bien qu'intéressantes, ne nous conviennent pas puique nous cherchons une technique plus simple afin de traiter beaucoup d'information.

Bien que la première version de `Citatum` ne fasse pas l'extraction des citations, nous avons présenté des techniques d'extractions existantes que nous pourrons éventuellement inclure dans notre système. Finalement, nous avons présenté des

techniques de résumé d'articles multiples. Ces techniques, plus adaptées à nos besoin, permettent d'éliminer la redondance. Notre système, `Citatum`, va utiliser une méthode extractive. La technique que nous avons choisie, celle de Carbonell et Goldstein [3], permet de choisir des phrases tout en éliminant la redondance. Au chapitre suivant, nous présenterons les corpus que nous avons utilisés.

CHAPITRE 3

DESCRIPTION DES DONNÉES

Notre premier problème est de réunir un corpus d'articles scientifiques pour entrainer et tester notre système. Dans ce chapitre, nous allons introduire les données utilisées pour notre thèse. La première étape de notre système va transformer les données en format XML et RDF. Ces formats vont permettre aux autres modules de notre système de faire des requêtes XPath et SPARQL(voir Figure 3.1).

À la section 3.1, nous présentons les données de l'AAN qui seront utilisées au chapitre 4. Ensuite, la section 3.2 explique les transformations en triplets RDF utilisant les vocabulaires Dublin Core et Friend Of A Friend. Aussi, nous y présentons comment les textes des articles scientifiques sont découpés en sections et transformés en format JATS/XML.

La section 3.3 présente les corpus des compétitions TAC2014, CL-2014 et CL-2016. Ce sont les données utilisées pour l'apprentissage (Chapitre 6), la construction des résumés (Chapitre 7) et l'évaluation de notre système(Chapitre 8).

Il existe différents types de corpus en fonction des prétraitements appliqués aux fichiers.

- Balisage : un prétraitement a identifié et balisé de l'information dans le texte. Cette information peut être :
  - Méta-informations : titre, auteurs, nom de la revue, ...
  - Sections : identification des entêtes de section.
  - Citations : identification des marqueurs de citation dans le texte.
- Références : les références bibliographiques sont extraites et décomposées en champs.
  - Marqueur de citation.
  - Auteurs.
  - Titre.

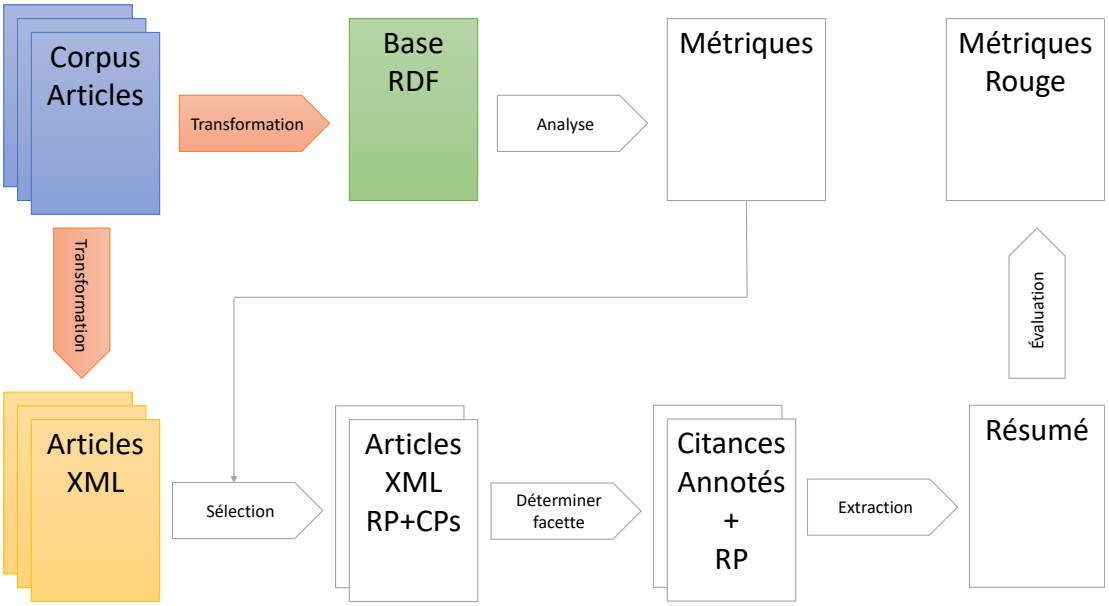

Figure 3.1 – (Détail de la Figure 1.2) Le corpus de base est transformé en deux bases de données : une base RDF pour les méta-informations et une base de fichiers XML pour les articles.

– Revue.
– Année de publication.
– Numéro de page.

L'ACL[1] (Association for Computational Linguistics) est une association qui promeut la recherche et le développement en informatique linguistique. Elle s'occupe des conférences telles que ACL, EACL[2], EMNLP[3], NAACL[4] et IJCNLP[5]. L'ACL Anthology[6] (AA) est une archive numérique d'articles publiés en informatique linguistique.

L'AA contient des publications d'événements de l'ACL et autres(voir le tableau 3.I). Lors de l'écriture de ce document, AA hébergeait 35 230 articles.

L'ACL Anthology Network[7] (AAN) ajoute à cette anthologie des graphes de citation et collaboration entre auteurs. Elle a été créée par Mark Thomas Joseph et est maintenue à l'University of Michigan [31, 32, 34]. L'AAN comprend 21 212 publications de l'ACL Anthology(voir le tableau 3.I). Ces publications sont tirées de 229 actes de conférences, 130 volumes de journaux, et 432 actes de workshops. La distribution des publications par événement est présentée dans le tableau 3.I. Cette distribution est présentée pour le nombre d'actes par événement de l'AA, pour le nombre d'actes par événement présent dans l'AAN et le nombre d'articles par événement qui sont présents dans l'AAN. Il est à remarquer que certains actes sont comptées plusieurs fois, car ils contiennent des événements conjoints. L'ACL Anthology Network (AAN) ajoute certaines méta-informations et des liens de référence entre ces articles. Dans ce chapitre, nous allons présenter les données fournies par l'AAN et les traitements que nous leur appliquons.

3.1 Les données de l'ACL Anthology Network

La dernière version de l'AAN, créée en décembre 2013, contient de l'information sur 21 212 articles de l'ACL. Ces articles sont écrits par 17 792 auteurs. Leur base de

[1] `https://www.aclweb.org/`
[2] EACL : Association for Computational Linguistics - European Chapter
[3] EMNLP : Empirical Methods in Natural Language Processing
[4] NAACL : The North American Chapter of the Association for Computational Linguistics
[5] IJCNLP : International Joint Conference on Natural Language Processing
[6] `http://aclanthology.info/`
[7] `http://clair.eecs.umich.edu/aan/index.php`

Titre	Classe	Nombre d'actes AA	Nombre d'actes AAN	Nombre d'articles AAN
Computational Linguistics Journal	J	136	129	785
Transactions of the Association of the Computational Linguistics	J	3	1	28
ACL Annual Meeting	C	81	71	4054
European Chapter of ACL	C	26	20	831
North American Chapter of ACL	C	48	43	1339
Lexical and Computational Semantics and Semantic Evaluation	C	12	5	431
Applied Natural Language Processing Conference	C	10	9	334
Conference on Empirical Methods in Natural Language Processing	C	20	18	1490
Conference on Computational Natural Language Learning	C	24	24	683
SIGs	C/W	264	209	5222
Int'l Committee on Computational Linguistics (ICCL) Conf.	C	58	51	3361
Human Language Technology Conf.	C	60	55	2365
Int'l Joint Conf. on Natural Language Processing	C	29	20	714
International Conference on Language Resources and Evaluation	C	8	1	620
Pacific Asia Conference on Language, Information and Computation	C	21	0	0
Rocling Computation Linguistics Conference and Journal	C	86	0	0
Theoretical Issues In Natural Language Processing	C	4	4	119
Australasian Language Technology Association Workshop	C	12	0	0
International Conference Recent Advances in Natural Language Processing	C	6	0	0
JEP/TALN/RECITAL	C	23	4	33
Message Understanding Conf.	C	5	6	155
NIST's TIPSTER Text Program	C	3	3	111
autre	W	591	431	—

Tableau 3.I – Le nombre d'actes contenus dans les publications de l'AA et l'AAN et le type d'acte : (J)ournaux (C)onférence ou (W)orkshop au 1 juillet 2015.

données contient un réseau représentant les 110 976 citations d'article de l'ANN vers d'autres articles de l'AAN et un autre réseau représentant les 142 450 collaborations entre auteurs. L'article de Dragomir R. Radev et al [33] décrit une version antérieure de l'AAN contenant 84 237 citations d'articles de l'ANN vers d'autres articles internes à l'AAN sur un total de 155 858 citations, il y avait donc 71 621 citations vers des articles non compris dans l'AAN. Nous pouvons donc approximer que 54% des citations sont incluses dans l'AAN. Cette information a été accumulée sur plusieurs années. La base de données est composée de deux fichiers de format texte décrivant les articles et de 21 212 fichiers contenant le texte des articles. Le premier fichier de description contient la méta-information sur les articles et le deuxième contient un réseau de citations.

3.1.1 Méta-informations sur les articles

La figure 3.2 contient un exemple de méta-information pour un des articles.

```
id = {A00-1002}
author = {Hajic, Jan; Hric, Jan; Kubon, Vladislav}
title = {Machine Translation of Very Close Languages}
venue = {Applied Natural Language Processing Conference
      and Meeting of the North American Association for
      Computational Linguistics}
year = {2000}
```

Figure 3.2 – Exemple des méta-informations incluses dans l'AAN.

L'ACL Anthology affecte un identificateur (`id`) à chaque article. Cet identificateur est composé d'une lettre identifiant le journal (conférence, workshop) qui est suivie de deux chiffres représentant l'année de publication. Ces trois caractères sont suivis d'un tiret et de quatre chiffres. Le premier indique le volume et les trois suivants identifient l'article dans le volume. Une exception est faite pour les workshops, les deux premiers chiffres indiquent le volume et les deux derniers identifient l'article. Par contre, les auteurs et les journaux ne sont pas liés entre eux. De plus, certaines entrées dans le fichier de méta-information ont un encodage différent pour les accents selon l'année où l'information a été ajoutée. Ces accents sont surtout utilisés pour les noms d'auteurs non anglophones. Les auteurs (`author`) ont été extraits. Ils sont énumérés dans une liste et séparés par des points virgules. Les noms et prénoms sont séparés par une virgule. Puisque l'AAN n'a pas d'identificateur pour les auteurs, ils

n'ont pas fait de désambiguïsation sur le nom des auteurs. Deux auteurs ayant le même nom sont donc considérés comme le même auteur par notre système.

3.1.2 Liens entre articles

Le deuxième fichier contient le graphe des citations. Plus particulièrement, un arc est placé entre deux articles si le premier cite le deuxième. Il a un format très simple où chaque ligne représente une citation. L'exemple de la figure 3.3 indique que l'article `A00-1002` cite deux autres articles de l'anthologie : `C90-3057` et `P98-1080`, sans en indiquer le nombre de citations.

```
A00-1002 ==> C90-3057
A00-1002 ==> P98-1080
```

Figure 3.3 – Exemple de références incluses dans l'AAN.

3.1.3 Texte des articles

Le texte des articles a été construit à partir des pdf, soit par extraction ou par reconnaissance optique pour les articles plus anciens. Ils contiennent parfois des erreurs de reconnaissance, et l'information sur la mise en page est perdue, voici quelques types de problèmes rencontrés :

- Certains caractères ne sont pas reconnus : *uch* plutôt que *such*
- Ajout d'espace à l'intérieur d'un mot : *overal l*.
- Les mots coupés à la ligne par un tiret ne sont pas reconstruits.
- La méta-information est incluse dans le texte sans séparateur.
- Les titres de sections ne sont plus en évidence.
- Des numéros de pages sont insérés dans le texte.
- Les figures, graphe et dessin sont retirés, mais leurs légendes peuvent apparaitre au milieu d'un paragraphe.
- Le texte des tableaux est extrait et placé sans les séparateurs graphiques (cases) du tableau.

3.2 Transformation appliquée aux données

Ces scripts vont uniformiser l'information vers deux formats : XML et RDF. Ainsi, dans les scripts de traitement nous pouvons utiliser des lecteurs communs pour en extraire l'information. Nous pouvons aussi faire des requêtes en utilisant `XPath` pour le XML et `SPARQL` pour le RDF.

3.2.1 Génération de données en format RDF

Notre expérience à la compétition ESWC-14 nous a montré qu'il était pratique d'interroger les méta-informations sous forme de triplets RDF à l'aide de requêtes SPARQL. Notre système construit donc un fichier contenant les triplets RDF en format Turtle. Pour construire ces triplets, nous avons utilisé les vocabulaires Dublin Core et Friend Of A Friend. Plus particulièrement nous avons utilisé les relations suivantes :

- Dublin Core : `dc:title`, `dc:isPartOf`, `dc:date`, `dc:creator` et `dc:references`.
- Friend Of A Friend : `foaf:familyName` et `foaf:givenName`.

Nous avons uniformisé l'encodage du fichier en UTF-8 avant de le transformer en RDF. Notre script de transformation va devoir construire un identificateur unique pour chaque auteur, en évitant de créer des doublons. Un dictionnaire des auteurs et des journaux est construit lors de la lecture afin de leur assigner un identificateur unique. Les années de publications et les titres de journaux sont regroupés et identifiés.

- Les identificateurs uniques pour les auteurs sont construits en remplaçant les espaces entre les prénoms et noms de famille par un souligné. Cela nous permet de résoudre un problème simple : il arrive dans la base de données que les noms ne soient pas divisés correctement. Par exemple, les noms `Das Gracas Volpe Nunes, Maria` et `Nunes, Maria Das Gracas Volpe` donnent l'identificateur `maria_das_gracas_volpe_nunes`. Les lettres sont transformées en minuscule. Cela ne permet pas d'identifier un auteur dont le nom est écrit d'une autre façon, par exemple `Eric Villemonte De La Clergerie` et `Eric De La Clergerie`.

Aussi, si deux auteurs ont le même nom, ils seront considérés comme un seul auteur.

- Pour les publications, nous utilisons les mêmes identificateurs que l'ACL Anthology. Ils utilisent l'identificateur de l'article en enlevant le numéro de l'article. Par exemple, l'article *"Machine Translation Of Very Close Languages"* est identifié par `A00-1002` et est le deuxième article de la publication *"Applied Natural Language Processing Conference And Meeting Of The North American Association For Computational Linguistics"*, donc la publication aura le numéro `A00-1`.

La relation `dc:isPartOf` est utilisée pour lier un article à sa publication. Les auteurs sont placés dans une séquence `rdf` liée par la relation `dc:creator`. La relation `dc:reference` indique les citations. Lorsque traités, nous obtenons les triplets de la figure 3.4. Notre sérialisation en Turtle a produit plus de 280 000 triplets[8]. Une affiche sur ce projet a été présentée à WebNLG 2015[9].

[8] Les triplets sont rendus disponibles sur le site : `http://www-ens.iro.umontreal.ca/~malenfab/acl-metadata.ttl`

[9] `http://www.loria.fr/~gardent/WebNLG2015/malenfant-abstract.pdf`

```
acl:A00-1002
    dc:title    "Machine Translation Of Very Close
   Languages"^^xsd:string ;
    dc:isPartOf acl:A00-1 ;
    dc:creator [ a rdf:Seq ;
        rdf:_1 acl:jan_hajic ;
        rdf:_2 acl:jan_hric ;
        rdf:_3 acl:vladislav_kubon ] ;
    dc:references acl:C90-3057 ,
                  acl:P98-1080 .
acl:A00-1
    dc:title "Applied Natural Language Processing
   Conference And Meeting Of The North American
   Association For Computational
   Linguistics"^^xsd:string ;
    dc:date  "2000"^^xsd:gYear .
acl:jan_hajic
    foaf:familyName "Hajic"^^xsd:string ;
    foaf:givenName  "Jan"^^xsd:string .
acl:jan_hric
    foaf:familyName "Hric"^^xsd:string ;
    foaf:givenName  "Jan"^^xsd:string .
acl:vladislav_kubon
    foaf:familyName "Kubon"^^xsd:string ;
    foaf:givenName  "Vladislav"^^xsd:string .
```

Figure 3.4 – Exemple de 17 triplets correspondant à l'article de la figure 3.2 et 3.3 sous format TTL.

3.2.2 Génération de textes en format XML

Il reste à produire une version XML des fichiers textes fournis pour les articles. Une version XML permet d'utiliser des librairies standards pour la lecture et la consultation des fichiers, facilitant l'écriture du code. Notre objectif est de séparer les sections du texte. Entre autres, nous voulons isoler le résumé signalétique (abstract) et la bibliographie. La plupart des articles débutent le résumé signalétique avec le mot *abstract*, ce qui facilite la détection du début du résumé. Pour en trouver la fin, nous essayons de trouver une ligne qui contient le mot *Introduction* souvent précédé du chiffre *1*.

Pour les sections suivantes, nous cherchons des lignes commençant par un nombre, en tenant compte de leur ordre. Les mots *reference* ou *bibliography* sont utilisés pour détecter le début de la bibliographie. La fin du texte marque la fin de la bibliographie. Finalement, les sections sont divisées en phrases. Pour cela, les lignes d'une section sont placées sur une seule ligne. Elles sont séparées d'un espace, sauf si le dernier mot se terminait par un tiret. Dans ce cas le tiret sera simplement supprimé. Un séparateur de phrase de la librairie `nltk` en `Python` est ensuite utilisé. Les abréviations suivantes ont été fournies au séparateur de phrase : {`al, Fig, etc, cf, i, e`}.

Les résultats obtenus sont placés dans des fichiers XML. Nous avons utilisé le format JATS[10] (Journal Article Tag Suite) qui décrit des éléments XML pour les articles scientifiques. Ce format permet de placer les méta-informations. Nous utilisons la base RDF que nous avons construite pour extraire cette information et la placer dans la structure.

3.3 Données des compétitions TAC 2014, CL-2014 et CL-2016

Les compétitions TAC 2014, CL-2014 et CL-2016 ont distribué des ensembles de données annotées (voir le tableau 3.II). L'ensemble d'entrainement de TAC2014 est composé de 20 sujets (topic) en biologie. Chaque sujet comprend un article de référence et 10 articles le citant. Cela représente 313 citations annotées. Chaque citation a été annotée par quatre annotateurs. Ces données vont nous permettre d'entrainer notre système (voir le chapitre 5). La compétition CL-2014 comprend 10 sujets de l'ACL. Chaque sujet est composé d'un article et d'une dizaine d'articles le

[10] `http://jats.nlm.nih.gov/`

citant. La version CL-2016 comprend les 30 sujets contenant en moyenne 17 articles. Pour ces compétitions, chaque citation a été annotée par un seul annotateur.

Pour chaque compétition, un résumé de l'article a été construit par chaque annotateur. Les fichiers d'annotations contiennent plusieurs informations pour chaque annotation.

- Un numéro d'identification du sujet.
- Un numéro d'identification de la citation.
- L'article de référence.
- L'article citant.
- L'emplacement du marqueur de citation.
- Le marqueur de citation.
- L'emplacement de la citation.
- La citation.
- Les emplacements des phrases de référence.
- Les phrases de référence : les phrases de l'article cité qui correspondent au sujet de la citation.
- La facette attribuée aux phrases de référence.
- L'identification de l'annotateur.

	TAC 2014	CL-2014	CL-2016			**Total**
			Train	Dev	Test	
Nombre d'articles	200	84	94	165	257	800
Nombres de citances	313	141	134	219	354	1161
Nombres d'annotateurs	4	1	1	1	1	

Tableau 3.II – Données des compétitions.

Chacun des champs est terminé par une barre verticale '|'. Voir la figure 3.5 pour un exemple d'annotation. Afin d'unifier les fichiers pour notre système, nous avons transformé chaque fichier d'annotation en format XML. La figure 3.6 présente un exemple d'une annotation en XML.

Topic ID : D1409_TRAIN |
Citance Number : 6 |
Reference Article : Sherr.txt |
Citing Article : Moon.txt |
Citation Marker Offset : 4903-4905 |
Citation Marker : 23 |
Citation Offset : 4467-4906 |
Citation Text : E2F1 stabilizes p53 by inducing ... pathway [23] |
Reference Offset : ['11278-11752', '19891-20494', '20891-21172'] |
Reference Text : There is now compelling evidence ... pathways |
Discourse Facet : Discussion_Citation |
Annotator : B, |

Figure 3.5 – Exemple d'annotation pour une citance.

Notre première tâche a été de nous assurer que nous obtenions les mêmes valeurs pour les déplacements (offset) des textes que ceux figurant dans les annotations. Certains fichiers contenaient un *Byte Order Mark* (BOM), ce qui décalait les caractères. Les fichiers ont été transformés en format UTF-8 sans BOM. Le champ de référence pouvait contenir jusqu'à trois extraits différents. La figure 3.7 contient la description RNC (RELAX NG Schema - Compact) de la version XML des annotations.

Nous voulions construire un corpus facile d'utilisation pour nos opérations. Nous avons utilisé le corpus de L'ACL Anthology Network qui est une partie du corpus de l'ACL Anthology. Ce corpus comprend des actes de conférences, workshops et journaux. Après avoir modifié le texte en UTF-8, nous avons construit un graphe RDF représentant la méta-information des 21 212 articles et identifiant les auteurs et actes. Nous avons aussi extrait les sections des textes des articles et séparé leurs phrases. Chaque article a été transformé en format JATS/XML. La construction de ces fichiers XML et RDF va faciliter l'accès à l'information, que ce soit à l'aide de requêtes SPARQL ou l'utilisation de librairie.

Il serait intéressant de pouvoir mieux corriger les textes des articles et pouvoir y ajouter de l'information. Par exemple, repérer les citations et les textes des citances

```
<citation-annotation annotator="B" citance-number="6"
  cp="Moon.txt" facet="Discussion_Citation"
  rp="Sherr.txt">
 <marker>
  <extrait end-offset="4905" start-offset="4903">
  23
  </extrait>
 </marker>
 <citance>
  <extrait end-offset="4906" start-offset="4467">
  E2F1 stabilizes p53 by inducing the expression of
  ...
  suppressor pathway [23]
  </extrait>
 </citance>
 <reference>
  <extrait end-offset="11752" start-offset="11278">
  There is now compelling evidence that particular
  ...
  retinoblastoma, RB loss occurs in many tumor types
  </extrait>
  <extrait end-offset="20494" start-offset="19891">
  The ability of deregulated E2F to induce ARF
  ...
  rrest and, later, to apoptosis ( Lomazzi et al., 2002)
  </extrait>
  <extrait end-offset="21172" start-offset="20891">
  in mouse tumor models lacking functional RB or
  ...
  pathways
  </extrait>
 </reference>
</citation-annotation>
```

Figure 3.6 – Version XML d'une annotation pour une citance.

```
datatypes xsd = "http://www.w3.org/2001/XMLSchema-datatypes"

start = annotation-list

annotation-list = element annotation-list {
        element citation-annotation {
            attribute annotator { xsd:string },
            attribute citance-number { xsd:int },
            attribute cp { xsd:string },
            attribute facet { xsd:string },
            attribute rp { xsd:string },
            element marker { extrait } *,
            element citance { extrait } *,
            element reference { extrait } *
        }*
    }

extrait = element extrait {
        attribute end-offset { xsd:int },
        attribute start-offset { xsd:int },
        text
    }
```

Figure 3.7 – Schéma RNC décrivant les fichiers XML utilisés pour contenir les annotations.

et découper la section des références ajouterait de l'information, car le graphe des citations de l'AAN ne contient que les citations à l'intérieur des 21 212 articles.

Dans le chapitre suivant, nous allons montrer comment nous avons extrait de l'information des liens entre les documents en utilisant SPARQL et la librairie `rdflib` en `Python` sur la base de données RDF.

CHAPITRE 4

UTILISATION DES LIENS ENTRE DOCUMENTS

Nous cherchons à construire le résumé à partir de l'information que d'autres chercheurs ont retenue de l'article de référence. Plus particulièrement, le texte des citations vers l'article de référence sera utilisé pour constituer un résumé de l'impact qu'a eu l'article sur la communauté scientifique. Le résumé d'un article sera donc construit à partir de l'analyse de plusieurs autres qui le citent. Notre première étape sera d'analyser le graphe des citations construit au chapitre précédent (voir la figure 4.1).

L'AAN fournit des scripts `perl` pour l'analyse de ces graphes. Nous voulons que `Citatum` soit indépendant de ces scripts, ainsi nous pourrons importer d'autres corpus qui ne seront pas nécessairement dans le même format que ceux de l'AAN. Cela nous permet de réutiliser nos scripts sur tous les corpus que nous importerons dans notre base de données `RDF`. Notre objectif est de reproduire les résultats qu'ils ont obtenus en utilisant notre représentation `RDF`. Les scripts de l'AAN permettent d'effectuer les tâches suivantes :

1. Construction de graphes (section 4.1)
   (a) Extraction de graphes :
      - Graphe des citations entre articles. (code 4.1)
      - Graphe des citations entre auteurs. (code 4.2)
      - Graphe des collaborations entre auteurs. (code 4.3)

   (b) Filtre :
      - Se limiter aux articles publiés avant une certaine date.
      - Suppression des autocitations. (code 4.4)

   (c) Réduction de graphes :
      - Limiter le graphe à un sous-ensemble des noeuds. Le choix des noeuds est aléatoire.
      - Limiter le graphe à un sous-ensemble des arcs. Le choix des arcs est aléatoire.

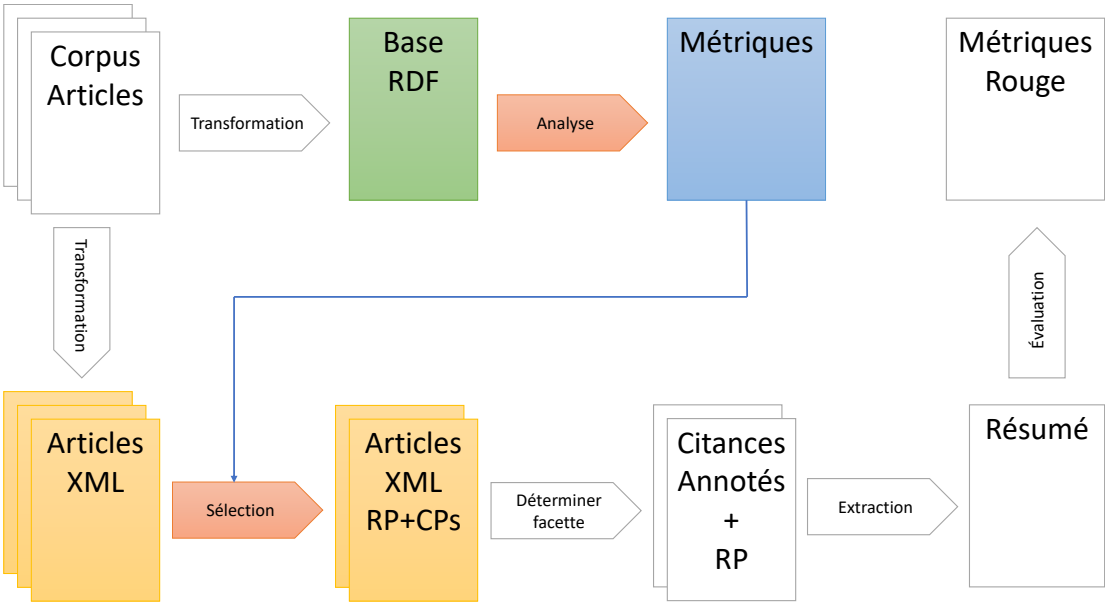

Figure 4.1 – (Détail de la Figure 1.2) Des métriques sont calculées dans la base RDF afin d'assister la sélection des articles à résumer et des articles les citants.

- Utiliser l'algorithme `ForestFire` pour extraire un sous-graphe.
- Extraire la plus grande composante connexe du graphe.

2. Calcul de métriques (section 4.2)

   - Degré des noeuds :
     - Déterminer le nombre de citations par auteur et par article.
     - Vérifier si la distribution des degrés suit une loi de puissance.
   - Associativité : trouver le niveau de densité (*clustering*) des noeuds du graphe.
   - Calcul des plus courts chemins :
     - Trouver le diamètre du graphe
     - Faire la moyenne des plus courts chemins.
     - Calculer la moyenne harmonique des distances géodésiques.
     - Calculer le rapprochement (*closeness*) entre les noeuds.
     - Calculer la centralité (*centrality* et *betweenness*) des noeuds.

- Calculer l'index `H` pour chaque auteur.
- Calcul de la métrique `LexRank`.

Nous allons reproduire le comportement de certains de ces scripts en utilisant notre version `RDF` de l'information 3.2.1. Pour cela, nous allons utiliser des requêtes `SPARQL` et des scripts `python`. L'information extraite par ces scripts va nous être utile pour la construction de nos résumés. Plus particulièrement pour la sélection des articles qui composeront un résumé. Il est à noter que les résultats obtenus par nos scripts sont légèrement différents de ceux obtenus par l'AAN. L'AAN a fait ces calculs sur la plus grande composante connexe du graphe, alors que nous avons utilisé le graphe complet pour nos calculs.

4.1 Construction de graphes

La construction de graphe se fait en trois étapes. Le graphe de base est extrait, filtré et réduit.

4.1.1 Extraction de graphes

Il y a trois requêtes de base. Une première requête `SPARQL` (code 4.1) extrait le graphe des citations entre articles. Les citations sont identifiées par le prédicat `dc:references` (ligne 3).

```
1 SELECT ?aCitant ?aCite
  WHERE {
      ?aCitant dc:references ?aCite .
4 }
```

Code 4.1 – Graphe des citations entre articles

La deuxième (code 4.2) détermine le graphe des citations entre auteurs. Cette requête parcourt la liste des auteurs, indiqués par le prédicat `dc:creator` (ligne 5 et 8) de deux articles lorsqu'il y a un lien de citation entre eux (ligne 3). Les filtres (ligne 6 et 9) permettent de ne pas tenir compte des indicateurs de séquence `RDF`. Les doubles sont éliminés (ligne 1).

```
SELECT DISTINCT ?nCitant ?nCite
WHERE {
    ?aCitant dc:references ?aCite ;
             dc:creator
                 [ ?b1 ?nCitant ] .
    FILTER ( ?nCitant != rdf:Seq )
    ?aCite dc:creator
        [ ?b2 ?nCite ] .
    FILTER ( ?nCite != rdf:Seq )
}
```

Code 4.2 – Graphe des citations entre auteurs

Le code 4.3 calcule le graphe des collaborations entre auteurs. Les lignes 3 à 5 construisent le produit croisé de la liste d'auteurs d'un article avec lui-même. Les filtres (ligne 7 à 9) éliminent les indicateurs de séquence et les collaborations d'un auteurs avec lui-même. Les doubles sont éliminés (ligne 1).

```
SELECT DISTINCT ?nCitant1 ?nCitant2
WHERE {
    ?aCitant dc:creator [
        ?b1 ?nCitant1 ;
        ?b2 ?nCitant2 ] .
    FILTER ( ?nCitant1 != ?nCitant2 )
    FILTER ( ?nCitant1 != rdf:Seq )
    FILTER ( ?nCitant2 != rdf:Seq )
}
```

Code 4.3 – Graphe des collaborations entre les auteurs

Ces trois requêtes permettent d'extraire les couples qui serviront d'arc dans le graphe qui sera analysé.

4.1.2 Filtre

Il est possible d'extraire les graphes de citation en éliminant les autocitations (un auteur citant un de ses articles). Cette fonctionnalité n'est pas présentement utilisée par `Citatum`, mais elle pourrait servir si nous voulions nous assurer que notre résumé de l'impact d'un article ne soit pas influencé par l'opinion de l'auteur sur ses propres articles. Pour cela, une requête (code 4.4) permet de trouver les références qui sont

des autocitations. Elle est utilisée avant les trois requêtes présentées plus haut et trouve des couples qui seront enlevés au graphe avant la requête principale. Les listes d'auteurs (ligne 3 et 5) d'un article en citant un autre (ligne 4) sont parcourues afin de trouver un auteur commun (ligne 6).

```
1 SELECT ?aCitant ?aCite
  WHERE {
      ?aCitant dc:creator [ ?b1 ?nCitant ] .
4     ?aCitant dc:references ?aCite .
      ?aCite dc:creator [ ?b2 ?nCite ] .
      FILTER ( ?nCitant = ?nCite )
      FILTER ( ?nCitant != rdf:Seq )
8 }
```

Code 4.4 – Extraction des autocitations

4.1.3 Réduction de graphes

Lorsque le graphe de base a été extrait, il est possible de le réduire pour des raisons de performance. La taille actuelle du graphe de l'AAN ne nécessite pas une telle réduction.

Nous utilisons des scripts `Python` pour extraire un sous-graphe aléatoirement. La première technique consiste à choisir aléatoirement un sous-ensemble ($S' \subset S$) de noeuds du graphe de base ($G =< S, A >$). Ensuite nous trouvons tous les arcs de l'ensemble des arcs (A) du graphe qui relient des noeuds de l'ensemble S' ($A' = \{(v_1, v_2) | v_1 \in S', v_2 \in S', (v_1, v_2) \in A\}$). Puisque le graphe des citations contient peu d'arcs par rapport au nombre de noeuds (environ 12 arcs par noeud comparé à 18 000 noeuds), cette technique donne un graphe qui contient très peu d'arcs.

La deuxième technique consiste à choisir aléatoirement un certain nombre d'arcs $A' \subset A$. Dans ce cas, l'ensemble des noeuds choisis devient l'ensemble des noeuds aux extrémités des arcs de A' ($S' = \{v_1, v_2 | (v_1, v_2) \in A'\}$).

4.2 Calcul de métriques

Les métriques que nous calculons sont basées sur différentes valeurs présentes dans un graphe : le degré des noeuds, l'associativité et la longueur des plus courts

chemins entre les noeuds.

4.2.1 Degré des noeuds

Un premier ensemble de métriques se calcule simplement avec le degré des noeuds (nombre d'arcs entrant et sortant). Ces calculs nous donnent directement le nombre de citations par auteur ou article. La moyenne des degrés obtenue est de 12.22 (même valeur que l'AAN). Soit la fonction $f(k)$ de distribution des degrés dans un graphe contenant n noeuds. Nous posons $P(k) = \frac{f(k)}{\sum_{i=1}^{n} f(k)}$ la probabilité qu'un noeud soit de degré k et $C(k) = \sum_{i=k}^{n} P(k)$ La probabilité qu'un noeud soit de degré d'au moins k. Il a été remarqué que pour un phénomène naturel, $C(k)$ donne souvent une loi de puissance. Nous pouvons vérifier si le graphe de citation répond à cette affirmation. Cette fonction est illustrée dans la figure 4.2. Pour faire notre calcul, nous devons prendre le logarithme des valeurs pour chaque axes. Pour un degré i, nous posons $x_i = \log(i)$ et $y_i = \log(f(i))$. Ces valeurs sont présentées à la figure 4.3

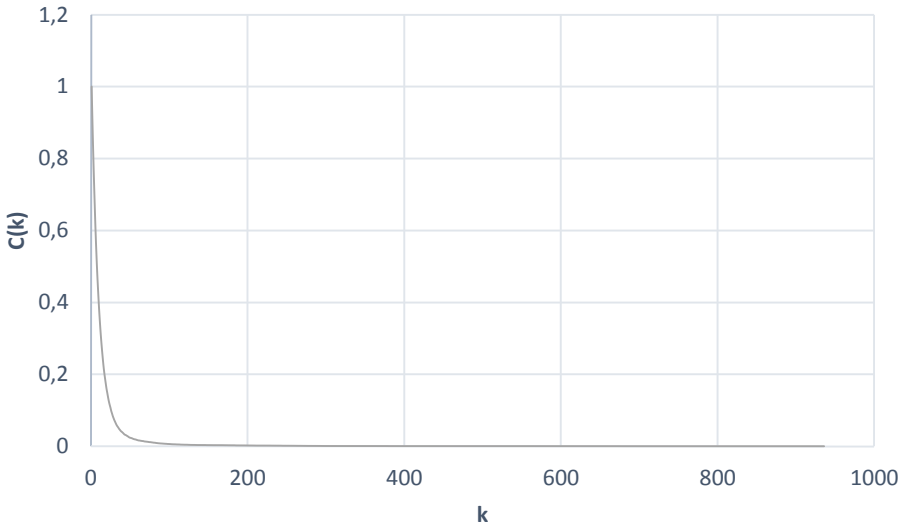

Figure 4.2 – Probabilité cumulative d'avoir un noeud de degré supérieur ou égal à k

Il est possible de calculer la pente de cette fonction avec son écart-type. Habituellement, si la pente obtenue est supérieure à 2 alors la distribution est considérée étant une loi de puissance.

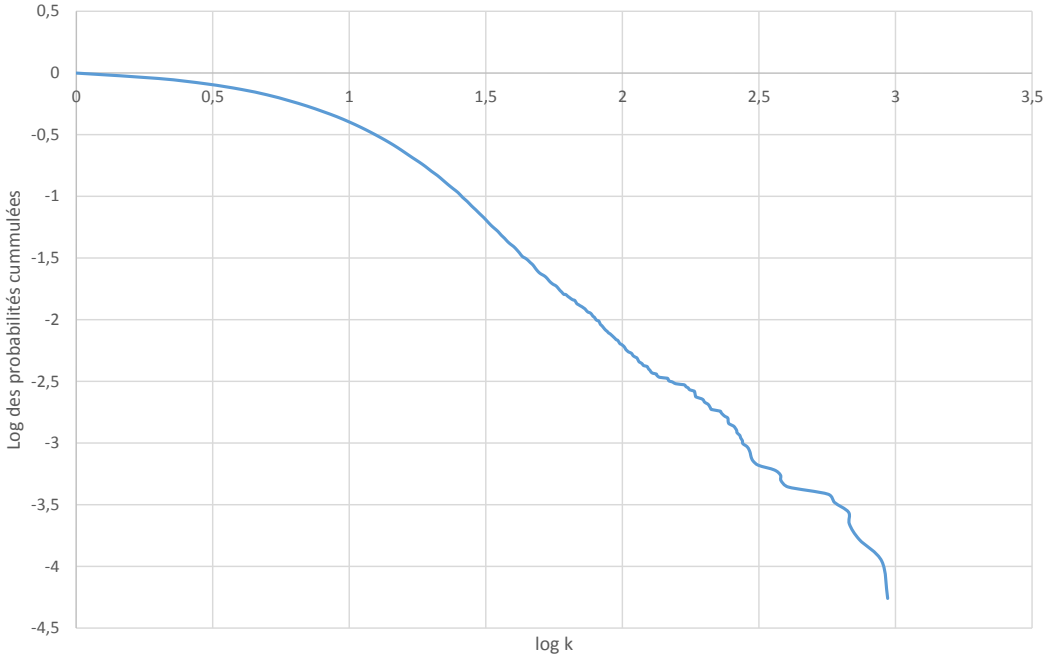

Figure 4.3 – Log des distributions

$$\alpha = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x^2 - (\sum x)^2}$$
$$r^2 = \frac{\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{(\sum x^2 - (\sum x)^2) \times (\sum y^2 - (\sum y)^2)}}$$

Nous avons obtenu une pente $\alpha = -1.1$ avec un écart de $r^2 = 1.00$.

4.2.2 Associativité

Un autre ensemble important de métriques calculent la densité ou l'associativité d'un graphe. Ces métriques permettent, entre autres, de calculer si un graphe est un petit monde (*small world*). Un petit monde est un graphe où la moyenne des plus courts chemins entre les noeuds est très petite comparée au nombre de noeuds

dans le graphe, ce qui implique un niveau d'associativité élevé entre les noeuds. Cela représenterait un sous-ensemble d'articles ayant beaucoup de citations entre eux. Ce qui indiquerait qu'ils décrivent un sujet commun et il serait intéressant de les résumer ensemble. Nous allons premièrement présenter des calculs communs à ces métriques et ensuite nous allons présenter deux métriques d'associativité (densité) : celle de Watts-Strogatz [40] et celle de Newman [24].

Maintenant que nous avons les degrés de chaque noeud du graphe, il est possible de calculer le nombre de triplets (R) pour lesquels un noeud (v) est central. Un triplet est une suite de trois noeuds voisins (liés par des arcs). Par exemple, si l'article A cite l'article B et que l'article B cite l'article C (écrit plus simplement : $A \rightarrow B \rightarrow C$). Dans ce cas, nous disons que le noeud B est central à ce triplet. Cette valeur $R(v)$ nous permet de calculer la densité (*clustering*) d'un noeud et la densité du graphe. Une région dense indiquerait un sujet (un ensemble d'articles ayant le même sujet), et les régions moins denses seraient les frontières entre ces sujets.

Nous commençons avec le calcul suivant pour chaque noeud (v) du graphe :

$$R(v) \quad = \quad \frac{\mathrm{Deg}(v)(\mathrm{Deg}(v)-1)}{2}$$

Ensuite, nous devons trouver le nombre de triangles $T(v)$ pour lesquels le noeud (v) est central. Un triangle est un triplet transitif, c.-à-d. s'il y a un arc de A vers B et un autre arc de B vers C ($A \rightarrow B \rightarrow C$) alors il doit y avoir un arc de A vers C ($A \rightarrow C$). Dans ce contexte, B est désigné le noeud central de ce triplet transitif. Il suffit de vérifier s'il existe un arc entre chaque paire de sommets voisins du noeud central. Le coefficient de densité de Watts-Strogatz pour un noeud est le ratio $\frac{T(v)}{R(v)}$. Nous avons calculé que la moyenne des coefficients obtenue pour le graphe est de 0.18102, ce qui est similaire à la valeur de 0.1803 obtenue par l'AAN.

Une autre métrique de densité pour un graphe est celle de Newman.

$$C = \frac{3 \times \text{nbr. triangles dans le graphe}}{\text{nbr. triplets dans le graphe}}$$

Puisqu'un triangle passe par trois sommets du graphe, si nous faisons la somme des triangles pour chaque noeud nous aurons trois fois le nombre de triangles dans le graphe. Il suffit de faire le calcul suivant pour obtenir le coefficient de densité de Newman.

$$C = \frac{\sum T(v)}{\sum R(v)}$$

Nous avons calculé un coefficient de densité de Newman de 0.0642, alors que l'AAN a obtenu 0.0631.

4.2.3 Calcul des plus courts chemins

Pour les calculs suivants, nous calculons la matrice des plus courts chemins. Comme le graphe contient plus de 18 000 noeuds et qu'il y a peu d'arcs par noeud, nous avons utilisé un algorithme de plus court chemin applicable dans la représentation de listes d'adjacences. Dans ce cas, l'algorithme de Floyd-Warshall est appliqué à chaque noeud du graphe. Cela utilise moins de mémoire, car le graphe des citations contient peu d'arcs par rapport au nombre de noeuds.

```
def performFlowAnalysis( self, a_flowAnalyser ):
a_flowAnalyser.preprocessing( self )

for node in self.nodes :
a_flowAnalyser.init( node )

while not all( [ a_flowAnalyser.asConverge( node )
for node in self.nodes ] ):
for node in self.nodes:
a_flowAnalyser.exit( node )
for node in self.nodes:
a_flowAnalyser.entry( node )
```

Code 4.5 – Propagation des flux

L'algorithme a été appliqué dans le contexte d'un algorithme de propagation des flux dans un graphe (code 4.5). Ces algorithmes transportent des valeurs d'un noeud à l'autre jusqu'à ce que les valeurs se stabilisent. Ils sont composés de deux fonctions :

`entry` et `exit`. La fonction `entry` calcule la valeur affectée à un noeud à partir des valeurs émises par les noeuds voisins. La fonction `exit` calcule la valeur qui sera émise par un noeud selon la valeur qu'il contient. Une valeur de départ est affectée à chaque noeud (fonction `init`). Cette algorithme reçoit en argument une instance de classe (`a_flowAnalyser`) contenant la définition des fonctions `entry, exit` et `init`.

Dans le cas du plus court chemin, nous avons utilisé les fonctions présentées dans le code 4.6

```
class ShortestPath:
def __init__( self ):
pass

def preprocessing( self, a_graph ):
pass

def init( self, a_node ):
a_node.entry = { a_node.id : 0 }
a_node.exit  = {}
a_node.hasBeenModified = True

def entry( self, a_node ):
for node in a_node.incoming:
for ident, length in node.exit.items():
if ( ident not in a_node.entry or
a_node.entry[ ident ] > length ):
a_node.entry[ ident ] = length
a_node.hasBeenModified = True

def exit( self, a_node ):
a_node.hasBeenModified = False

for ident, length in a_node.entry.items():
a_node.exit[ ident ] = length + 1

def asConverge( self, a_node ):
return not a_node.hasBeenModified
```

Code 4.6 – Plus court chemin

Lorsque les plus courts chemins sont disponibles, cela nous permet de calculer le diamètre du graphe, qui est le plus long des plus courts chemins. Le graphe des citations a un diamètre de 21.

Nous pouvons aussi calculer la moyenne des plus courts chemins. L'AAN a utilisé deux méthodes pour calculer la moyenne des longueurs des plus courts chemins. Soit L_{ij} la longueur du plus court chemin entre les noeuds i et j, et N_{ij} le nombre de paires de noeuds qui peuvent être joints, la moyenne est calculée comme suit :

$$\text{avr} = \frac{\sum_{i=1}^{n} \left(\frac{\sum_{j=1}^{n} L_{ij}}{n_i} \right)}{n}$$

où n_i est le nombre de noeuds que nous pouvons joindre à partir du noeud i. Cela nous donne une moyenne de 5.8 alors que l'AAN a obtenu 5.82. Il est normal que nous ayons une valeur inférieure, puisqu'ils font leurs calculs sur la plus grande composante connexe du réseau. Il y a donc des chemins des plus petites composantes qui ne sont pas comptés. Nous pouvons aussi calculer la moyenne à l'aide de la formule de Ferrer.

$$\text{avr}_F = \frac{\sum_{i=1}^{n} (\sum_{j=1}^{n} L_{ij})}{N_{ij}}$$

Cela nous a donné une moyenne de 5.02 alors que l'AAN obtient 5.52. La moyenne de Ferrer compte, pour chaque noeud, une moyenne des chemins partant de ce noeud. Ensuite la moyenne de ces moyennes est faite.

Une autre métrique liée à la longueur des chemins est la moyenne harmonique des distances géodésiques (*harmonic mean geodesic distance*).

$$H = \frac{\frac{n(n-1)}{2}}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{L_{i,j}}}$$

Nous avons obtenu une valeur de 61.35, alors que l'AAN obtient 56.03. Les derniers coefficients calculés par l'AAN sont les coefficients de centralité.

4.2.4 Calcul de la métrique PageRank

La métrique qui nous intéresse est le `PageRank` [2] qui est calculée pour repérer les articles qu'il serait intéressant de résumer. Nous avons utilisé l'algorithme de propagation des flux pour faire ce calcul. Nous avons utilisé deux versions de l'algorithme `PageRank`. Premièrement, la version originale :

$$\begin{aligned} \text{Entry}(n) &= \frac{1-d}{|V|} + d(\sum_{m \in \text{In}(n)} \text{Exit}(m)) \\ \text{Exit}(n) &= \frac{\text{Entry}(n)}{|\text{Out}(n)|} \end{aligned}$$

Où V est l'ensemble des noeuds du graphe et d est un coefficient (*damping factor*) qui indique la probabilité qu'un lecteur décide de lire l'article cité. Cela nous donne le code 4.7.

Cette technique suppose que chaque papier a la même valeur à l'origine. Nous avons essayé une autre technique où nous avons affecté une importance différente aux articles en fonction de leurs origines : journal, proceeding ou workshop. Il existe une version de `PageRank` pour associer un poids aux arcs du graphe [43]. Nous avons adapté cette version pour affecter un poids W_n aux noeuds du graphe, les journaux ont un poids de 3, les proceedings de 2 et les workshops de 1. La deuxième version utilise les équations suivantes :

$$\begin{aligned} \text{Entry}(n) &= \frac{1-d}{|V|} + dW_n(\sum_{m \in \text{In}(n)} \text{Exit}(m)) \\ \text{Exit}(n) &= \frac{\text{Entry}(n)}{\sum_{m \in \text{Out}(()n)} W_m} \end{aligned}$$

```
class PageRank:
def __init__( self, a_dampingFactor_float,
a_threshold_float ):
self.dampingFactor = a_dampingFactor_float
self.threshold = a_threshold_float

def preprocessing( self, a_graph ):
self.k = ( 1.0 - self.dampingFactor )
/ a_graph.vertexSize()

def init( self, a_node ):
a_node.entry = self.k
a_node.exit  = 0.0
a_node.oldValue = self.k + 2.0 * self.threshold

def entry( self, a_node ):
a_node.oldValue = a_node.entry

a_node.entry = self.k + self.dampingFactor
* sum( [ node.exit for node in a_node.incoming ] )

def exit( self, a_node ):
outgoingNumber = len( a_node.outgoing )
if outgoingNumber == 0:
a_node.exit = 0.0
else:
a_node.exit = a_node.entry / outgoingNumber

def asConverge( self, a_node ):
return
abs( a_node.oldValue - a_node.entry ) < self.threshold
```

Code 4.7 – PageRank

Le code 4.8 contient ces équations. Les deux versions ont donné des résultats similaires. Le nombre de citations est plus grand que le poids de chaque noeud, certains articles ont jusqu'à 900 citations. Cela explique nos résultats. Cela nous a permis d'identifier les articles les plus importants selon `PageRank` (Tableau 4.I) et nous a permis de trouver les articles les plus importants qui les citent(Tableau 4.II).

```
class WeightedPageRank:
    def __init__( self, a_dampingFactor_float,
                  a_threshold_float, a_weight_fct ):
        self.dampingFactor = a_dampingFactor_float
        self.threshold = a_threshold_float
        self.weight_fct = a_weight_fct

    def preprocessing( self, a_graph ):
        self.k = ( 1.0 - self.dampingFactor )
           / a_graph.vertexSize()

 def init( self, a_node ):
        a_node.entry = self.k
        a_node.exit  = 0.0
        a_node.oldValue = self.k + 2.0 * self.threshold
        a_node.weight = self.weight_fct( a_node.info )

    def entry( self, a_node ):
        a_node.oldValue = a_node.entry

        incomingSum = self.dampingFactor
                * a_node.weight
       * sum( [ node.exit for node in a_node.incoming ] )
        if incomingSum == 0.0 :
            a_node.entry = self.k * a_node.weight
        else :
            a_node.entry = self.k + incomingSum

    def exit( self, a_node ):
        outgoingSum =
   sum( [ node.weight for node in a_node.outgoing ] )
        if outgoingSum == 0:
            a_node.exit = 0.0
        else:
            a_node.exit = a_node.entry / outgoingSum

    def asConverge( self, a_node ):
        return
      abs( a_node.oldValue - a_node.entry ) < self.threshold
```

Code 4.8 – PageRank pondéré

Pagerank	**Paper**	**# cit.**
0.0122	A Stochastic Parts Program and Noun Phrase Parser for ...	236
0.0104	Finding Clauses in Unrestricted Text by Finitary and ...	5
0.0070	A Stochastic Approach to Parsing	9
0.0059	A Statistical Approach to Machine Translation	196
0.0045	The Contribution Of Parsing To Prosodic Phrasing In An ...	4
0.0040	Building A Large Annotated Corpus Of English : The Penn ...	928
0.0039	The Mathematics Of Statistical Machine Translation : Parameter ...	729
0.0031	Attention, Intentions, and the Structure of Discourse	354
0.0023	Deterministic Parsing Of Syntactic Non-Fluencies	37
0.0022	A Statistical Approach To Language Translation	30

Tableau 4.I – Les dix articles ayant le meilleur PageRank dans l'AAN avec le nombre de citations.

Pagerank	**Paper**
0.0104	Finding Clauses in Unrestricted Text by Finitary and Stochastic Methods
0.0040	Building a Large Annotated Corpus of English : The Penn Treebank
0.0018	Word Association Norms, Mutual Information, and Lexicography
0.0014	A Program For Aligning Sentences In Bilingual Corpora
0.0012	A Practical Part-Of-Speech Tagger
0.0011	Deducing Linguistic Structure From The Statistics Of Large Corpora
0.0011	Parsing, Word Associations And Typical Predicate-Argument Relations
0.0011	Word Association Norms, Mutual Information, And Lexicography
0.0011	A New Statistical Parser Based On Bigram Lexical Dependencies
0.0010	Identifying Word Correspondences In Parallel Texts

Tableau 4.II – Les dix articles avec le plus haut PageRank citant le premier article du tableau 4.I : *A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text*

Nous avons montré comment nous pouvons utiliser notre base RDF. Elle nous permet d'extraire les graphes de citation et de collaboration à l'aide de requêtes SPARQL. Nous avons aussi extrait plusieurs métriques : détection d'une loi de puissance, plus court chemin, densité et centralité. Entre autres nous avons calculé PageRank à l'aide d'un algorithme de propagation des flux. Cette mesure va nous permettre de détecter l'importance qu'ont eue certains articles. Dans le chapitre suivant nous allons regarder les différents contextes rhétoriques (facettes) des phrases à l'intérieur d'un article et comment les déduire.

CHAPITRE 5

DÉTERMINATION DES FACETTES DES CITATIONS

Une facette est un attribut affecté à une phrase indiquant son contexte rhétorique. Par contexte rhétorique d'une phrase, nous entendons sa fonction communicationnelle dans le contexte de l'article scientifique [37]. Les phrases extraites d'un texte perdent leur contexte, elles ne sont plus entourées des autres phrases. Le rôle d'une facette est de pallier ce manque en ajoutant de l'information sur le contexte rhétorique de la phrase. La facette de type RESULTS sera attribuée à cette phrase :

> Culotta et al (2004) extended this work to estimate kernel functions between augmented dependency trees and achieved 63.2 F-measure in relation detection and 45.8 F-measure in relation detection and classification on the 5 ACE relation types.

Nous voulons automatiser l'attribution d'une facette aux phrases d'un article, y compris les citances (voir la Figure 5.1). Ces facettes vont nous permettre de qualifier les liens (citations) entre articles et de construire des résumés spécifiques. Les facettes seront aussi utilisées pour regrouper les phrases d'un article et nous permettre de trouver une section de texte à laquelle une citation se rapporte. Ainsi, il serait possible pour un chercheur consultant des citations de retrouver les parties correspondantes dans l'article cité.

La section 5.1 va décrire le choix de l'ensemble de facettes utilisé par notre système. Ensuite, à la section 5.2 nous présenterons la méthodologie utilisée pour entrainer notre système pour la détermination automatique des facettes. Finalement, la section 5.3 décrira comment notre système retrouve les phrases référées par une citation.

5.1 Ensembles de facettes

Nous avons présenté la proposition de Simone Teufel pour les contextes rhétoriques d'une phrase dans la section 2.2 Parmi les sept facettes proposées, seulement trois

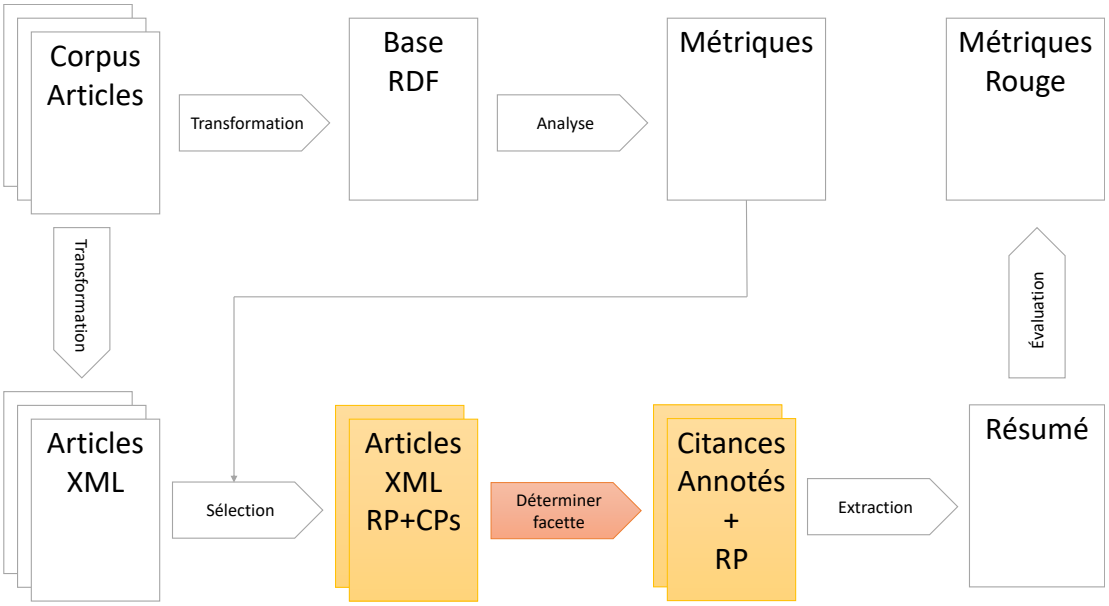

Figure 5.1 – (Détail de la Figure 1.2) Les citances sont extraites et analysées pour en déterminer les facettes.

se rapportent aux citations : BASIS, CONTRAST et OTHER. Les autres se rapportent aux phrases de l'article : AIM, TEXTUAL, OWN, BACKGROUND. Un autre système de facettes est le CiTO (Citation Typing Ontology). C'est un module ontologique qui fait partie du SPAR (Semantic Publishing and Referencing Ontologies). Il distingue 41 facettes possibles pour une citation (voir la figure 5.2). Ces facettes sont très précises. Par exemple, USESDATAFROM, EXTENDS, UPDATES, USESMETHODIN, ... sont des spécialisations de la facette BASIS de Simone Teufel.

En général, les utilisateurs de ce standard n'utilisent qu'un petit sous-ensemble des facettes. Pour les compétitions TAC 2014, CL-2014 et CL-2016 seulement cinq facettes ont été choisies : HYPOTHESIS, METHOD, RESULTS, IMPLICATION et DISCUSSION. Ces facettes se rapportent aux phrases de l'article, dans la compétition elles doivent être affectées à des extraits de l'article. Dans nos expériences, nous les affecterons aussi à des citances pour indiquer le rôle de la phrase auquel la citance fait référence dans l'article cité. Par exemple, la phrase mentionnée au début du chapitre est référée par la citance suivante :

> dependency kernel Zhou et al.

citesAsRecommendedReading	citesAsRelated	compiles
citesAsPotentialSolution	disagreesWith	confirms
citesAsMetadataDocument	speculatesOn	corrects
citesAsSourceDocument	usesDataFrom	disputes
containsAssertionFrom	usesMethodIn	parodies
includesQuotationFrom	plagiarizes	retracts
obtainsBackgroundFrom	agreesWith	supports
citesForInformation	critiques	credits
includesExcerptFrom	describes	derides
usesConclusionsFrom	discusses	extends
obtainsSupportFrom	documents	reviews
citesAsDataSource	repliesTo	updates
citesAsAuthority	ridicules	refutes
citesAsEvidence	qualifies	

Figure 5.2 – Les 41 facettes du CiTO.

Puisque la phrase référée est affectée à la facette results (voir plus haut), nous affecterons la même facette à cette citance. Pour les compétitions CL-2014 et CL-2016, la facette discussion a été remplacée par la facette aim. Nous avons utilisé ces facettes, puisque des données préannotées étaient disponibles pour ces trois compétitions comme nous l'avons évoqué au chapitre 3.

Comme nous voulons construire un système indépendant des domaines, applicable sur des articles autant en biologie qu'en mathématiques, nous avons décidé de ne pas tenir compte du vocabulaire appartenant à un domaine particulier. Nous nous sommes donc appuyé sur le lexique scientifique transdisciplinaire (LST), construit par Patrick Drouin [7, 8]. Ce lexique contient 1627 mots en anglais qui peuvent se retrouver dans un article scientifique, peu importe le domaine. La figure 5.3 présente un échantillon de ce lexique. Ainsi, `Citatum` devrait pouvoir fonctionner pour tous les domaines. Nous allons dénoter ce lexique L et ses mots par $w \in L$.

5.2 Entrainement pour la reconnaissance de facettes

Nous avons extrait des annotations les phrases de citance et de références avec leur facette pour entrainer notre système. Chaque phrase est divisée en mots à l'aide du segmenteur de la librairie `NLTK`. Seul les mots appartenant au LST sont retenus.

ability(nom)	background(nom)	calculate(verbe)	data(nom)
able(adj)	backward(adv)	calculation(nom)	date(nom)
about(adv)	barrier(nom)	call(verbe)	date(verbe)
above(adj)	base(nom)	call(nom)	day(nom)
above(adv)	base(verbe)	called(adj)	daylight(nom)
absence(nom)	basic(adj)	capability(nom)	deal(verbe)
absent(adj)	basis(nom)	capable(adj)	death(nom)
absolute(adj)	bearing(nom)	capacity(nom)	decision(nom)
absolutely(adv)	before(adv)	care(nom)	decrease(nom)
abstract(adj)	beforehand(adv)	carefully(adv)	decrease(verbe)

Figure 5.3 – Exemples de mots du Lexitrans.

Notre base de données pour l'entrainement et l'évaluation de la reconnaissance de facettes sera composée de paires constituées d'une liste de mots et d'une facette : $D :: [([\texttt{Mot}], \texttt{Facette})]$. Nous avons choisi une méthode d'apprentissage simple pour nos premières expérimentations. Nous construisons une table qui, pour chaque facette, va contenir le nombre d'occurrences de chaque mot. Notre système construit un profil (histogramme) (h_f) pour chaque facette.

$$h_f(w, D) = \sum [\mathrm{cnt}(w, W) | (W, f) \in D] \tag{5.1}$$
$$\mathrm{cnt}(w, W) = |[w_i \in W, w_i = w]| \tag{5.2}$$

Pour déterminer la facette d'une phrase, il suffit d'extraire la liste (P) des mots de la phrase qui sont aussi présents dans le LST. Un pointage s_f est calculé pour chaque facette f. Ce pointage est la somme des valeurs dans l'histogramme de f pour chaque mots w de la phrase. La facette avec le meilleur pointage est choisie.

$$s_f(P, D) = \sum_{w \in P} h_f(w, D) \tag{5.3}$$

Nous construisons deux systèmes de classification, un pour les citances et un pour les phrases référées. Ces deux éléments de discours sont assez différents pour demander leur propre classificateur. Chaque classificateur utilise une méthode de validation

croisée. L'ensemble des données est divisé en 4, 5, 6, 7, 8, 9 et 10 sous-ensembles et entrainé sur chaque combinaison. La meilleure combinaison est choisie comme résultat. Le tableau 5.I donne les mots les plus utilisés pour chaque facette. Nous avons remarqué que certains mots nuisent à l'entrainement. Par exemple : *human*, *response*, *development*, *number*. Nous obtenons de meilleurs résultats lorsqu'ils ne sont pas présents.

Afin de trouver la bonne combinaison de mots à contribution positive, nous avons utilisé un algorithme génétique pour déterminer un sous-ensemble du Lexitrans qui soit plus efficace à classifier les phrases. L'algorithme génère aléatoirement des sous-ensembles du Lexitrans comme population initiale ($L_i \subseteq L$). Trois transformations sont utilisées afin de construire la prochaine génération.

1. Combinaison : prendre aléatoirement deux membres de la population et unir un certain nombre de leurs mots : $L'_i = L'_j \cup L'_k$ où $L'_j \subseteq L_j \wedge L'_k \subseteq L_k$.
2. Supprimer un mot d'un membre de la population : $L'_i = L_i - \{w\}$ où $w \in L_i$.
3. Ajouter un mot à un membre de la population : $L'_i = L_i + \{w\}$ où $w \in (L - L_i)$.

Les membres ayant les meilleurs résultats lors de l'entrainement sont choisis pour passer à la génération suivante. Pour nos expériences, l'algorithme génétique construit 25 générations. Chaque génération commence avec 1 000 listes de mots et 9 000 listes de mots sont ajoutées. Nous utilisons les données présentées à la figure 3.II. Plus particulièrement, les ensembles 'Train' et 'Dev' de la compétition CL-2016 sont utilisés. Nous avons aussi testé sans l'algorithme génétique et sans l'utilisation du LST.

DISCUSSION (93 mots)	RÉSULTATS (76 mots)	IMPLICATION (45 mots)	MÉTHODE (34 mots)	HYPOTHÈSE (20 mots)
show	data	approach	data	will
evidence	similar	identification	approach	similar
data	show	data	determine	show
similar	effect	evidence	following	exclusively
effect	further	defined	similar	develop
even	significantly	will	region	uniform
crucial	contrast	show	further	defined

Tableau 5.I – Les sept mots les plus communs pour chaque facette après l'entrainement.

Les résultats sont présentés dans la table 5.II pour l'entrainement sur les citances et dans la table 5.III pour l'entrainement sur les phrases de l'article.

	Ensembles de test		
Ensembles d'entrainement	**Train**	**Dev**	**Train + Dev**
Train sans LST	47%	61%	59%
Train	65%	52%	57%
Train + Dev sans LST	56%	61%	59%
Train + Dev	61%	57%	58%
gen_T	74%	43%	55%

Tableau 5.II – Taux de succès pour l'attribution de facette à une citance.

	Ensembles de test		
Ensembles d'entrainement	**Train**	**Dev**	**Train + Dev**
Train sans LST	60%	60%	60%
Train	74%	46%	57%
Train + Dev sans LST	60%	61%	61%
Train + Dev	70%	59%	64%
gen_T	76%	35%	51%

Tableau 5.III – Taux de succès pour l'attribution de facette aux phrases d'un article.

Nous remarquons que, lorsque nous utilisons l'ensemble d'entrainement **Train** cela nous donne des bons résultats sur l'ensemble lui-même mais, cela donne des résultats inférieurs sur l'ensemble **Dev**. Après l'utilisation de deux ensembles pour l'entrainement, les résultats pour l'ensemble **Dev** augmentent mais ceux de l'ensemble **Train** sont réduits. Pour les citances, l'utilisation de l'algorithme génétique donne de meilleurs résultats seulement sur l'ensemble **Train**. Cela ne nous a pas aidé à obtenir de meilleurs histogrammes. L'utilisation du LST n'a pas diminué le taux de succès. Il reste à vérifier s'il permet d'obtenir des résultats comparables dans d'autres domaines.

5.3 Extraction des phrases référées

Une des tâches proposées dans les compétitions TAC 2014, CL-2014 et CL-2016 est d'associer un extrait d'un article avec chaque citation. Cet extrait serait le texte référé

par la citation. Notre hypothèse est que le texte référé et la citation devraient partager la même facette. Cette hypothèse nous permet de réduire l'espace de recherche pour le texte référé.

Pour une citance c_j^i ayant la facette f, notre système commence par extraire l'ensemble Q_f des phrases de l'article référé RP ayant la même facette f que la citance c_j^i. Ensuite, nous cherchons la phrase de Q_f qui est le plus similaire à c_j^i. Nous utilisons la métrique de similarité de Mihalcea, Corley et Strapparava [22]. Nous avons choisi cette métrique car elle associe une similarité entre chaque mot. Ainsi, même si une phrase ne réutilise pas les mêmes mots, des points lui sont données pour des mots ayant un sens voisin. Aussi, nous utilisons déjà cette métrique pour nos résumés automatiques, ce qui nous permet de réutiliser son code. Les détails de ce calcul sont donnés à la section 6.2.

Cette technique nous donne une métrique F1 de 0.095 sur l'ensemble **Train** et 0.052 pour l'ensemble **Dev** (table 5.IV). Nous avons réduit l'espace de recherche en utilisant les citances attribuées. Puisque l'identification des facettes n'est pas parfaite, il est possible que la phrase recherchée ait été enlevée de l'ensemble de recherche. Sur les neuf équipes qui ont participé à la compétition CL2016, `Citatum` s'est classé en sixième place pour l'identification de phrases référées. La meilleure équipe à obtenue une métrique F1 de 0.12 [15].

	Train	**Dev**
F1	0.095	0.052

Tableau 5.IV – Métrique F1 pour la recherche de phrases référées.

Nous gardons pour des travaux futurs (CL-2017, CL-2018) l'amélioration des algorithmes de cette section de notre système. Bien que l'extraction automatique de phrase référées soit importante pour les compétitions, notre système de résumé automatique ne l'utilise pas.

Dans ce chapitre, nous avons décrit comment notre système utilise un histogramme pour apprendre et déterminer la facette associée à une citance ou à une phrase d'un article. Nous avons aussi expliqué l'utilisation que nous avons faite du Lexique Scientifique Transdisciplinaire afin que notre système soit applicable à plusieurs domaines. Un algorithme génétique à été utilisé afin d'isoler les mots qui représentaient une plus grande contribution à l'apprentissage.

Ensuite, ces résultats ont été utilisés pour réduire l'espace de recherche pour

trouver la phrase référée par une citance. Les techniques proposées par les participants de `scisumm 2016` pour résoudre ce problème utilisent des méthodes numériques : apprentissage et métriques. Malgré la diversité des méthodes proposées, la meilleure équipe obtient un score F1 de 0.12, ce qui n'est pas très élevé. Notre méthode emploie une similarité qui tient compte des synonymes sans pour autant obtenir un meilleur pointage (F1 = 0.052). Pour de futures recherches sur cette tâche, il serait préférable de tester des techniques non numériques. Une première proposition serait d'utiliser les manipulations syntaxiques, tel celles proposées par Knight [17] (voir la Section2.3.3). Cela nous permettrait de comparer les structures syntaxiques des phrases. Les manipulations de Knight réduisent la taille des phrases vers une structure minimale. Il serait intéressant de vérifier si la comparaison des structures simplifiées, ayant moins de décoration verbale, donnerait un meilleur résultat que la comparaison des phrases originales. Une deuxième proposition serait de s'orienter vers des méthodes plus abstractives. À la section 2.3.2, nous avons expliqué la technique d'extraction de Genest et Lapalme [12]. Ils transforment le texte d'une phrase en un graphe représentant ses éléments syntaxiques, ce qui leur permet d'utiliser des règles logiques pour effectuer des déductions retrouvant l'information voulue. Il serait peut-être possible de modifier ces règles afin de repérer les phrases référées par une citation. Dans le prochain chapitre, nous présenterons notre technique pour l'automatisation des résumés.

CHAPITRE 6

CONSTRUCTION D'UN RÉSUMÉ

À partir d'un article du ANN, `Citatum` utilise la base de données `RDF` (voir le chapitre 3) afin de trouver une liste d'articles le citant. De ces articles, il extrait l'ensemble des phrases contenant les citations (figure 6.2). L'étape suivante est de choisir les phrases qui vont composer le résumé. Pour cela, nous devrons identifier les citances et choisir celles qui apportent le plus d'informations pertinentes (figure 6.1). Nous utilisons la méthode de *Maximal Marginal Relevance* (MMR) proposée par Jaime G. Carbonell et Jade Goldstein [3] (voir la section 2.5). Pour cette technique, une phrase de départ est choisie pour représenter le sujet du résumé (phrase requête). Nous prenons le titre de l'article comme phrase requête. Ensuite, nous cherchons à ajouter des phrases ayant un sujet similaire à la phrase requête et qui ajouteront le plus d'information nouvelle au résumé. Nous voulons donc une phrase ayant un sujet similaire à la phrase requête, mais traitant d'un sujet le plus différent possible d'une autre phrase déjà générée.

Cette technique utilise une métrique de similarité pour calculer l'apport d'une phrase. Jaime G. Carbonell et Jade Goldstein ont obtenu de meilleurs résultats avec la métrique de Mihalcea, Corley et Strapparava [22]. Aussi, nous présenterons les coefficients de ces équations que nous avons ajustés afin d'obtenir de meilleurs résultats. Finalement, les résumés seront présentés en HTML afin d'en faciliter la lecture et l'évaluation.

Nous allons premièrement expliquer comment les mots des textes ont été traités (section 6.1). Deuxièment, nous présenterons les métriques que nous avons utilisées (section 6.2). Les deux sections suivantes (6.3 et 6.4) vont expliquer comment les citations ont été étendues pour trouver les citances et comment le MMR à été appliqué pour construire les résumés. Finalement, les résultats obtenus sont présentés au chapitre 7.

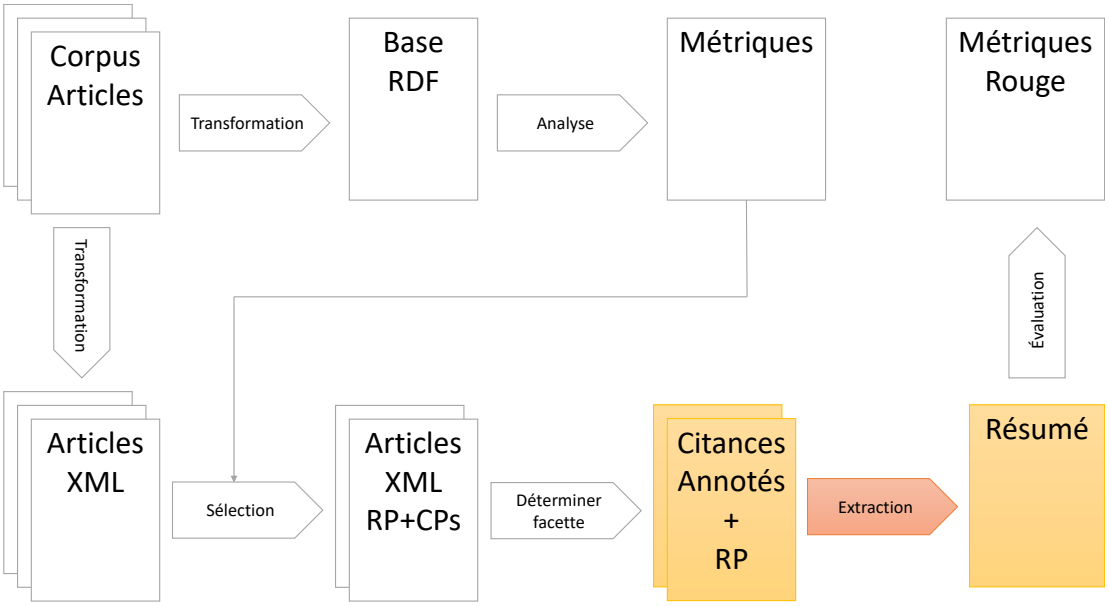

Figure 6.1 – (Détail de la Figure 1.2) Le chemin d'information de `Citatum` : le résumé automatique à partir des citances et d'extraits de l'article de référence.

6.1 Identification des mots

Nous voulons permettre à notre système de comparer les abréviations, fréquentes dans les textes scientifiques, avec les mots originaux, par exemple 'MMR'. Pour cela nous voulons que les abréviations soient remplacées par les mots qu'elles représentent (MMR $\mapsto$ Maximal Marginal Relevance). En général, lorqu'une abréviation est mentionnée pour la première fois dans un texte, elle est placée entre parenthèses après les mots qu'elle remplace. Notre système cherche des lettres entre parenthèses dans le texte. Si les mots avant les parenthèses commencent par les mêmes lettres, alors le système considère avoir identifié une abréviation. Cela nous permet de construire un dictionnaire des abréviations pour ensuite les remplacer dans tout le texte.

La métrique de similarité utilise quatre groupes de mots : nom, adverbe, adjectif et verbe. Cela va impliquer l'utilisation d'un étiqueteur des parties du discours (*POS tagging*) pour identifier les étiquettes grammaticales des mots. Le *POS tag* de `NLTK` est utilisé dont l'implémentation par défaut utilise un perceptron pré-entrainé. Nous utilisons le séparateur de mots de la librairie `NLTK` en `Python` afin de séparer les mots. Ensuite, les lemmes des mots sont extraits et le groupe-synonyme (les `synsets` de

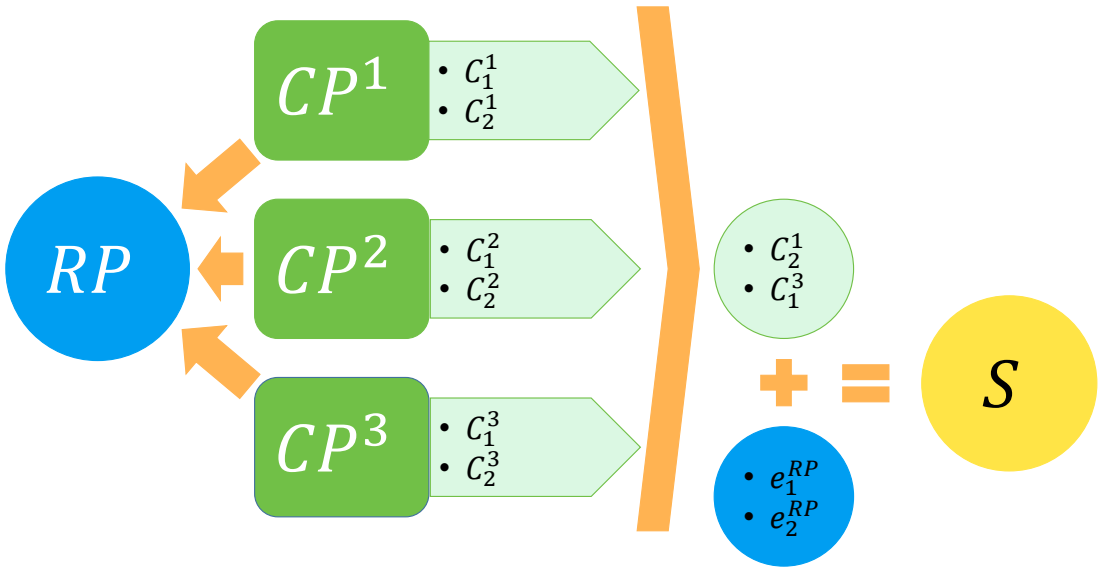

Figure 6.2 – Chaîne de traitement, reprise de la figure 1.1

`wordnet`) des mots est identifié.

Les phrases ne contenant pas de verbe sont éliminées pour enlever une partie du bruit introduit par la reconnaissance des caractères (OCR). Par exemple, des mots/nombres qui étaient dans des cellules d'un tableau dans l'article original. Le texte est maintenant prêt pour l'extraction des citances.

6.2 Métrique de similarité

Autant pour l'extraction des citances que pour la construction du résumé, nous utilisons une métrique de comparaison pour les phrases. Les métriques de ce type comparent deux phrases (P_1 et P_2) et calculent leur niveau de similarité. Nous utilisons la métrique de Mihalcea, Corley et Strapparava [22] (désigné par sim_{mcs}). Cette métrique essaie de jumeler les mots d'une phrase avec des mots équivalents dans une autre phrase (équation 6.3). Pour cela, une métrique secondaire est utilisée pour mesurer l'équivalence entre deux mots (sim_{wup}). Pour chaque mot d'une phrase, le mot de l'autre phrase ayant la plus grande similarité est identifié (voir la figure 6.3)

.

Lorsque nous avons trouvé tous les mots similaires, il reste à faire la moyenne des valeurs de similarité que nous pondérons avec l'importance du mot (équation 6.2). L'indice IDF est utilisé comme mesure d'importance d'un mot. Nous avons calculé l'indice IDF sur le total des articles du corpus de l'AAN. Ce calcul est fait dans les deux sens, c.a.d, de la première phrase vers la deuxième et ensuite de la deuxième vers la première. La moyenne des deux résultats est retournée (équation 6.1).

$$\text{sim}_{mcs}(P_1, P_2) = \frac{1}{2}(hsim(P_1, P_2) + hsim(P_2, P_1)) \quad (6.1)$$

$$hsim(S, D) = \frac{\sum_{w_i^S \in S} maxsim(w_i^S, D) \times \text{idf}_{w_i^S}}{\sum_{w_i^S \in S} \text{idf}_{w_i^S}} \quad (6.2)$$

$$maxsim(w_i^S, D) = \max_{w_j^D \in D} \text{sim}_{wup}(w_i^S, w_j^D) \quad (6.3)$$

La similarité entre les mots n'est calculée qu'entre des mots du même groupe (nom, adjectif, adverbe et verbe). Sinon, ils sont considérés comme différents (sim = 0). Cela permet de facilement interchanger les métriques de comparaison puisque certaines d'entre elles ne permettent pas la comparaison entre mots appartenant à différents groupes syntaxiques. Nous avons utilisé la métrique de Zhibiao Wu et Martha Palmer [42] pour comparer les mots (désigné par sim_{wup}). Cette métrique donne de bons résultats [22] et est disponible dans la librairie `nltk`.

Par exemple, la table 6.I montre deux phrases pour lesquelles nous voulons calculer la similarité

Ces phrases sont premièrement transformées en jetons couplés avec les étiquettes grammaticales. Pour le calcul des métriques, nous ne considérons que les suivantes : n (nom), v(verbe), a(adjectif) et r(adverbe). Ensuite, pour chaque mot d'une phrase, nous cherchons le mot de la même étiquette de l'autre phrase le plus similaire. Un exemple des résultats est dans la table 6.II. Cela nous donne une valeur de similarité de 0.31.

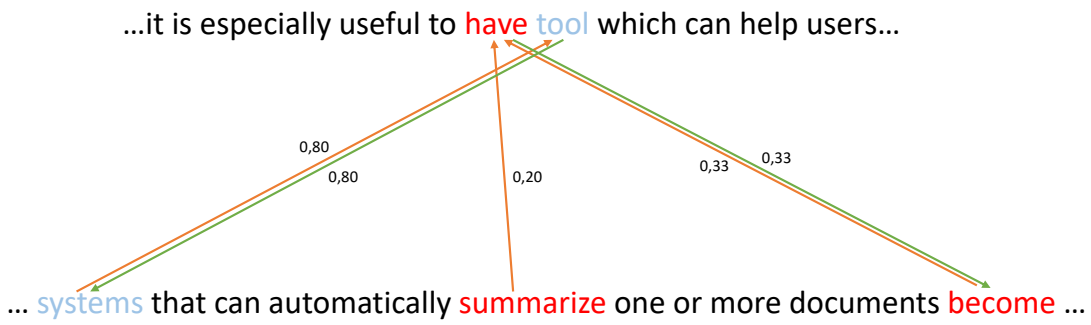

Figure 6.3 – Le calcul de similarité entre deux phrases détermine pour chaque mot d'une phrase, le mot le plus similaire dans l'autre phrase.

	As the amount of on-line information increases, systems that can automatically summarize one or more documents become increasingly desirable.	With the mushrooming of the quantity of on-line text information, triggered in part by the growth of the World Wide Web, it is especially useful to have tools which can help users digest information content.
n	amount, information, increases, systems, documents.	quantity, text, information, part, growth, World, Wide, Web, tools, users, information, content.
a	on-line, more, desirable.	on-line, useful, digest.
r	automatically, increasingly.	especially.
v	summarize, become.	mushrooming, triggered, is, have, help.

Tableau 6.I – Deux phrases et les mots qui les composent classés dans quatre groupes : nom (n), adjectif (a), adverbe (r) et verbe (v).

mot	**synset**(w_i^S)	$\text{idf}_{w_i^S}$	w_j^D	$\text{sim}_{wup}(w_i^S, w_j^D)$
amount	sum.n.01	0.74	part.n.01	0.60
information	information.n.01	0.17	information.n.01	1.00
increase	addition.n.03	1.07	measure.n.02	0.75
systems	system.n.01	0.14	tool.n.01	0.80
documents	document.n.01	0.81	text.n.01	0.77
summarize	sum_up.v.01	1.99	be.v.01 have.v.01	0.20
become	become.v.01	1.04	be.v.01 have.v.01	0.33

mot	**synset**(w_i^S)	$\text{idf}_{w_i^S}$	w_j^D	$\text{sim}_{wup}(w_i^S, w_j^D)$
quantity	measure.n.02	2.50	addition.n.03	0.75
text	text.n.01	0.34	document.n.01	0.77
information	information.n.01	0.17	information.n.01	1.00
part	part.n.01	0.36	sum.n.01	0.60
growth	growth.n.01	3.10	system.n.01	0.33
world	universe.n.01	1.46	system.n.01	0.62
web	web.n.01	1.43	system.n.01	0.55
tools	tool.n.01	1.06	system.n.01	0.80
users	user.n.01	1.06	system.n.01	0.53
content	content.n.01	1.16	information.n.01 addition.n.03	0.40
mushrooming	mushroom.v.01	9.23	become.v.01	0.25
triggered	trip.v.04	2.54	become.v.01	0.22
is	be.v.01	0.02	become.v.01	0.33
have	have.v.01	0.04	become.v.01	0.33
help	help.v.01	1.06	become.v.01	0.29

Tableau 6.II – Mots les plus similaires entre deux phrases, le tableau du haut présente les résultats lorsque $S = P_1, D = P_2$ et le tableau du bas présente les résultats lorsque $S = P_2, D = P_1$ (voir l'équation 6.1).

6.3 Extraction des citances

Les citations étant déjà identifiées dans nos textes (voir le chapitre 3), nous devons maintenant trouver les phrases environnantes qui composent la citance. Pour cela, nous voulons comparer chaque phrase avec la phrase suivante. Nous posons comme hypothèse qu'un groupe de phrases similaires représente le même sujet, dans notre cas, la citation. La métrique de similarité (6.1) est utilisée entre chaque phrase consécutive. Nous calculons la moyenne des similarités entre phrase consécutive pour un texte. Ensuite, pour constituer nos citances, nous prenons les phrases précédant et suivant la citance ayant une similarité supérieure à la moyenne.

6.4 Construction des résumés

Afin de construire le résumé, nous constituons deux ensembles de phrases, un composé des citances et l'autre composé des phrases de l'article à résumer. Les phrases du le résumé seront extraites de ces deux ensembles (voir la figure 6.4).

La similarité de Mihalcea, Corley et Strapparava est calculée entre chaque phrase de l'ensemble. Nous utilisons cette métrique, car c'est celle proposée dans l'article sur le MMR [3]. Elle a l'avantage de tenir compte des synonymes des mots et de leurs importances (IDF). Les phrases sont choisies afin de couvrir le plus de sujets différents possible. Elles sont choisies une à la fois, ainsi lorsqu'une nouvelle phrase est ajoutée, nous pouvons consulter les phrases déjà contenues dans le résumé pour évider la redondance. C'est le résultat espéré de l'algorithme MMR. L'algorithme utilise une phrase de départ (requête) afin de trouver les phrases suivantes et applique l'équation 6.4 pour trouver les suivantes (voir la figure 6.5). Nous avons utilisé le

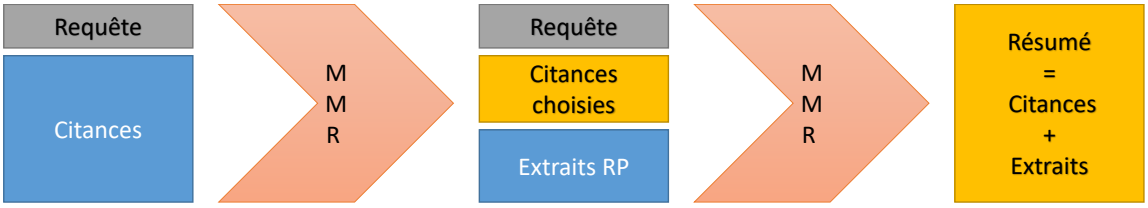

Figure 6.4 – L'algorithme MMR sera utilisé deux fois. Dans un premier temps, il va choisir des citances pour construire un résumé de base et ensuite il va compléter le résumé à l'aide des d'extrait de l'article de référence (RP).

titre de l'article comme phrase requête, telle que proposée par Jaime G. Carbonell et Jade Goldstein [3]. La phrase requête n'apparaîtra pas dans le résumé final. Dans un premier temps, les phrases de l'ensemble de citance sont ajoutées à l'article en utilisant la phrase requête. Lorsque nous avons suffisamment de phrases ajoutées des citances nous passons à l'ensemble des phrases extraites du RP.

Plus précisément, lorsque nous cherchons une phrase à ajouter au résumé, chaque phrase potentielle est comparée à la phrase requête. Nous cherchons une phrase la plus similaire possible à la requête (terme additionné dans l'équation 6.4). Par contre, nous comparons aussi chaque phrase potentielle avec les phrases déjà dans le résumé afin de trouver une phrase la moins similaire possible aux phrases déjà incluses (le terme soustrait dans l'équation 6.4). Nous soustrayons la similarité aux phrases du résumé à la similarité avec la requête, construisant une métrique de sélection unique pour chaque phrase potentielle. Cette combinaison linéaire est pondérée par un coefficient λ. La constante λ est un coefficient à ajuster afin d'obtenir le résumé voulu. Une valeur élevée indique que nous favorisons la similarité à la requête, alors qu'une valeur faible indique que nous défavorisons la similarité aux phrases du résumé. Pour nos tests, nous considérons que le résumé est assez long lorsqu'il contient 250 mots.

$$\text{MMR} \stackrel{\text{def}}{=} \underset{\text{CP}^i \in R \setminus V}{\arg\max} \left[\lambda(\text{Sim}(\text{CP}^i, Q) - (1-\lambda) \max_{\text{CP}^j \in V} \text{Sim}(\text{CP}^i, \text{CP}^j)) \right] \quad (6.4)$$

Nous avons trois coefficients à déterminer. Notre procédé utilise l'algorithme MMR à deux reprises. Pour chaque instance nous devons déterminer le coefficient λ qu'il est préférable d'utiliser. Nous devons aussi déterminer le ratio ρ d'information que chacune de ces phases va extraire.

1. ρ : le pourcentage de mot extrait de l'ensemble des phrases des citances lors de la première phase d'extraction.. Il y aura donc $(1-\rho)$ mots extraits de l'ensemble des phrases de l'article à résumer lors de la deuxième phase d'extraction.. .

2. λ_c : le λ utilisé pour extraire les citances lors de la première utilisation du MMR (voir la figure 6.4).

3. λ_r : le λ utilisé pour extraire les phrases de l'article lors de la deuxième utilisation du MMR (voir la figure 6.4).

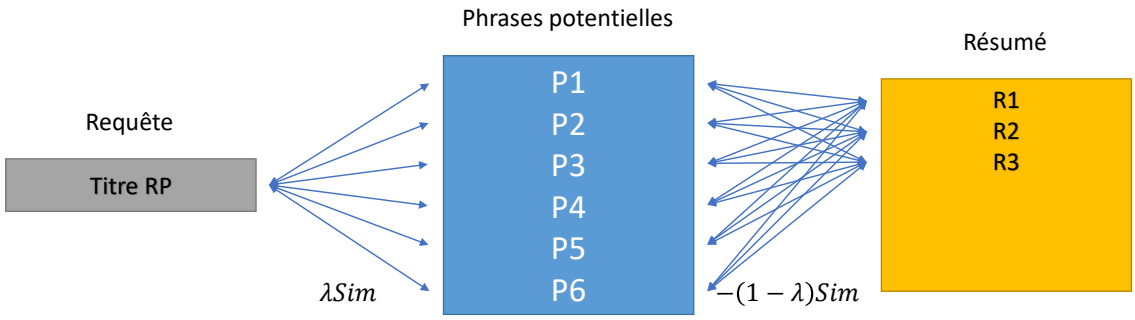

Figure 6.5 – Étape de sélection d'une phrase ajoutée au résumé. Les phrases potentielles sont comparées à la requête afin de trouver une phrase similaire. Aussi, les phrases potentielles sont comparées aux phrases déjà présentes dans le résumé pour éliminer la redondance.

La section 7.3 présentera notre technique d'attribution des valeurs pour ces coefficients.

Dans ce chapitre, nous avons présenté notre technique pour construire un résumé d'un article scientifique. Notre système remplace les abréviations par les mots qu'elles représentent et trouve les ensembles synonymes des noms, adverbes, adjectifs et verbes. Seules les phrases contenant des verbes sont conservées pour la construction des résumés. Nous utilisons ces résultats pour calculer la métrique de Mihalcea, Corley et Strapparava [22]. Cette métrique est utilisée par l'algorithme MMR afin d'extraire des phrases de deux ensembles, soit les citances et les phrases de l'article résumé. Cela nous permet de construire un résumé en limitant les phrases répétitives. Dans le chapitre suivant, nous présenterons les résultats expérimentaux obtenus.

CHAPITRE 7

ÉVALUATION

Nous allons maintenant présenter les résultats obtenus pour les compétitions BiomedSumm 2014 et CL-2016. Pour la compétition BiomedSumm (voir la section 7.1), notre système ne construisait pas encore les résumés. Seuls les résultats sur les facettes sont présentés. Par contre, pour la compétition CL-2016 (voir la section 7.2), les résumés ont été évalués à l'aide de l'ensemble de métriques `ROUGE`. Nous avons utilisé les résultats de cette évaluation pour ajuster les coefficients de nos équations (voir la figure 7.1). La section 7.3 décrit notre interface présentant les résumés construits par `Citatum`.

7.1 Résultat pour la compétition `BiomedSumm 2014`

Nous avions utilisé nos techniques de classement lors de la compétition `BiomedSumm 2014` (TAC 2014). Chaque citances était annoté par quatre personnes. Ceci nous donne possiblement quatre réponses différentes pour chaque facette lors de l'entrainement. Les annotateurs humains arrivent à un taux d'accord de 66% (voir le tableau 7.I). Ce taux représente le pointage maximum qu'un annotateur peut atteindre avec les meilleures réponses possibles. Cela indique qu'en moyenne il y a entre 2 et 3 annotateurs qui donnent la même réponse. Notre système a atteint 47.2%, indiquant qu'en moyenne deux annotateurs sont en accord avec les résultats du système. Les résultats de CL-2016 compare avec un seul annotateur, ce qui rend difficile la comparaison des résultats de cette compétition[1].

7.2 Résultat pour le corpus `scisumm 2016`

`ROUGE` (Recall-Oriented Understudy for Gisting Evaluation) est un ensemble de métriques pour évaluer les logiciels produisant des résumés [19]. Ces métriques mesurent la similarité entre deux résumés. L'ensemble de métriques `ROUGE` comprend

[1]La conférence TAC 2014 n'a jamais publié les résultats pour la tâche BiomedSumm 2014, il n'est donc pas possible de comparer nos résultats avec ceux des autres équipes.

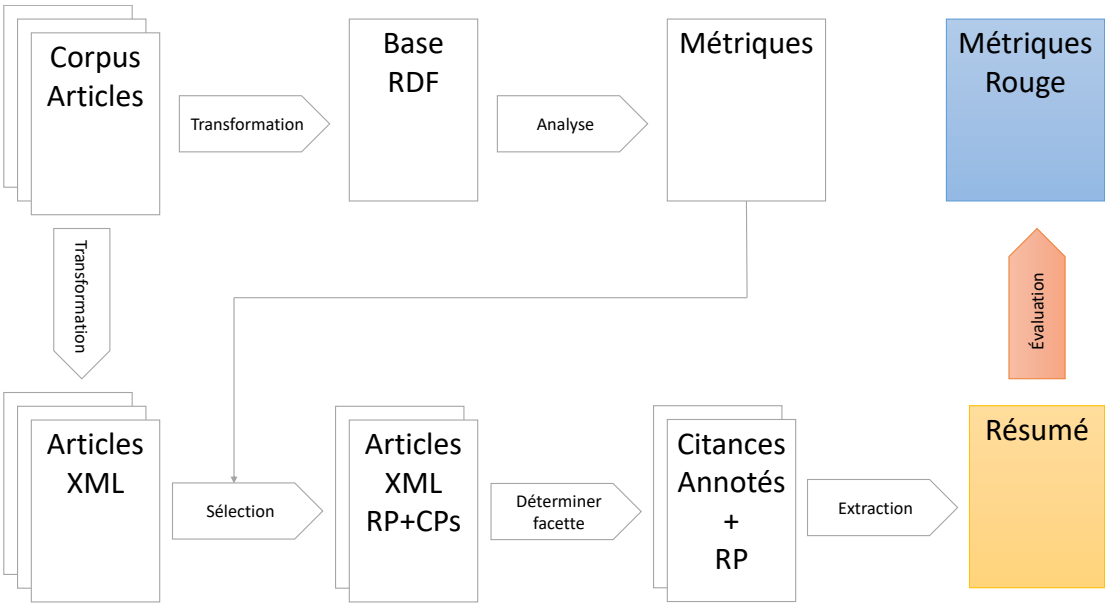

Figure 7.1 – (Détail de la Figure 1.2) Évaluation des résultats de notre système.

	TAC 2014
Nombre d'articles	200
Nombres de citance	313
Nombres d'annotateurs	4
Classification des citances	47.2%
Classification des textes référés	57.7%
Annotateurs humains	66.6%

Tableau 7.I – Résultat des classificateurs.

quatre métriques : `ROUGE-N`, `ROUGE-L`, `ROUGE-W` et `ROUGE-S`. Cette métrique est expliquée plus en détail à l'appendice I. Nous avons utilisé la métrique `ROUGE-N` pour évaluer nos résumés. La commande suivante à été utilisée :

```
ROUGE-1.5.5.pl -a -d -e ../data -l 250 -m -n 4 -x
```

Elle effectue le calcul sur les 250 premiers mots lemmatisés du résumé avec un n de 1, 2, 3 et 4. Nous ne conservons que les résultats pour $n = 2$ et $n = 4$.

Le système `Citatum` permet de contrôler trois coefficients pour la construction de nos résumés : le ratio k de phrases extraites des citations et de l'article de référence, une constante λ_c pour le calcul MMR pour les citances et une constante λ_r pour le calcul MMR pour les extraits de l'article de référence.

Nous avons fait des expériences avec différentes valeurs pour ces coefficients. Dans une première expérience, nous avons utilisé les valeurs $[0, 0.2, 0.4, 0.6, 0.8, 1]$ pour chacun des coefficients. Les résultats sont comparés avec les résumés fournis par les organisateurs de la compétition CL-2016. Les résultats sont présentés dans le tableau 7.II. Les métriques de `ROUGE-4` présentées varient entre 0.003 et 0.124. Chaque sous-tableau contient les résultats pour différentes valeurs de λ_c utilisées pour l'extraction des citations, les colonnes contiennent les résultats pour différentes valeurs de λ_r utilisées pour l'extraction des phrases de référence et les lignes contiennent les résultats pour différents ratios entre les phrases extraites des citances et des références. Un ratio de 0.0 indique que seulement des phrases de référence ont été extraites, alors qu'un ratio de 1.0 indique que seulement des citations ont été utilisées pour la construction du résumé. Cela a permis d'isoler une zone plus performante pour les résumés.

- k : 0.00 à 0.40
- λ_c : 0.20 à 0.60
- λ_r : 0.60 à 1.00

Moyenne de Rouge-4	λ_r						
λ_c | ratio cit-ref	0,00	0,20	0,40	0,60	0,80	1,00	Total général
0,00	**0,028**	**0,015**	**0,029**	**0,030**	**0,045**	**0,048**	**0,033**
0,00	0,124	0,035	0,022	0,036	0,074	0,078	0,061
0,20	0,010	0,015	0,046	0,037	0,071	0,074	0,043
0,40	0,010	0,013	0,053	0,035	0,063	0,064	0,040
0,60	0,014	0,011	0,034	0,037	0,031	0,035	0,027
0,80	0,008	0,012	0,016	0,030	0,028	0,032	0,021
1,00	0,003	0,003	0,003	0,003	0,003	0,003	0,003
0,20	**0,040**	**0,031**	**0,023**	**0,038**	**0,054**	**0,054**	**0,040**
0,00	0,124	0,035	0,022	0,036	0,074	0,078	0,061
0,20	0,029	0,056	0,022	0,054	0,081	0,081	0,054
0,40	0,025	0,030	0,022	0,050	0,072	0,071	0,045
0,60	0,025	0,024	0,027	0,041	0,043	0,042	0,034
0,80	0,021	0,026	0,029	0,034	0,040	0,039	0,031
1,00	0,015	0,015	0,015	0,015	0,015	0,015	0,015
0,40	**0,039**	**0,032**	**0,022**	**0,039**	**0,055**	**0,055**	**0,041**
0,00	0,124	0,035	0,022	0,036	0,074	0,078	0,061
0,20	0,030	0,059	0,025	0,054	0,081	0,082	0,055
0,40	0,022	0,028	0,024	0,050	0,073	0,073	0,045
0,60	0,022	0,027	0,024	0,044	0,045	0,044	0,034
0,80	0,022	0,026	0,023	0,037	0,043	0,040	0,032
1,00	0,016	0,016	0,016	0,016	0,016	0,016	0,016
0,60	**0,038**	**0,025**	**0,024**	**0,038**	**0,056**	**0,056**	**0,040**
0,00	0,124	0,035	0,022	0,036	0,074	0,078	0,061
0,20	0,029	0,032	0,027	0,052	0,081	0,081	0,050
0,40	0,020	0,025	0,027	0,049	0,072	0,074	0,044
0,60	0,017	0,020	0,024	0,046	0,046	0,045	0,033
0,80	0,018	0,021	0,025	0,027	0,044	0,037	0,029
1,00	0,019	0,019	0,019	0,019	0,019	0,019	0,019
0,80	**0,040**	**0,027**	**0,027**	**0,041**	**0,057**	**0,058**	**0,042**
0,00	0,124	0,035	0,022	0,036	0,074	0,078	0,061
0,20	0,029	0,033	0,029	0,053	0,081	0,081	0,051
0,40	0,027	0,033	0,033	0,052	0,074	0,076	0,049
0,60	0,019	0,021	0,030	0,048	0,047	0,046	0,035
0,80	0,019	0,022	0,026	0,036	0,047	0,045	0,033
1,00	0,020	0,020	0,020	0,020	0,020	0,020	0,020
1,00	**0,041**	**0,028**	**0,026**	**0,041**	**0,059**	**0,059**	**0,042**
0,00	0,124	0,035	0,022	0,036	0,074	0,078	0,061
0,20	0,033	0,036	0,031	0,053	0,083	0,083	0,053
0,40	0,026	0,032	0,030	0,052	0,075	0,075	0,048
0,60	0,019	0,021	0,027	0,048	0,048	0,047	0,035
0,80	0,021	0,024	0,028	0,037	0,049	0,047	0,034
1,00	0,022	0,022	0,022	0,022	0,022	0,022	0,022
Total général	**0,038**	**0,026**	**0,025**	**0,038**	**0,054**	**0,055**	**0,039**

Tableau 7.II – Métriques `ROUGE-4` pour les résumés sur présentés dans le cadre de la compétition CL-2016, Chaque sous-tableau contient les résultats pour différentes valeurs de λ_c, les colonnes contiennent les résultats pour différentes valeurs de λ_r et les lignes contiennent les résultats pour différents ratios entre les phrases extraites des citances et des références.

Moyenne de Rouge-4	λ_r						
λ_c | ratio cit-ref	0,60	0,68	0,76	0,84	0,92	1,00	Total général
0,20							
0,00	0,036	0,061	0,075	0,075	0,073	0,078	0,066
0,08	0,060	0,069	0,080	0,083	0,084	0,083	0,077
0,16	0,054	0,063	0,079	0,082	0,083	0,082	0,074
0,24	0,053	0,062	0,075	0,079	0,082	0,081	0,072
0,32	0,053	0,062	0,069	0,074	0,083	0,082	0,071
0,40	0,050	0,052	0,069	0,073	0,078	0,071	0,066
0,28							
0,00	0,036	0,061	0,075	0,075	0,073	0,078	0,066
0,08	0,060	0,069	0,080	0,083	0,084	0,083	0,077
0,16	0,054	0,063	0,079	0,082	0,083	0,082	0,074
0,24	0,052	0,060	0,079	0,083	0,083	0,082	0,073
0,32	0,052	0,059	0,072	0,077	0,083	0,077	0,070
0,40	0,049	0,060	0,070	0,073	0,082	0,072	0,068
0,36							
0,00	0,036	0,061	0,075	0,075	0,073	0,078	0,066
0,08	0,060	0,069	0,080	0,083	0,084	0,083	0,077
0,16	0,053	0,061	0,079	0,081	0,083	0,081	0,073
0,24	0,053	0,060	0,078	0,082	0,082	0,081	0,073
0,32	0,052	0,060	0,077	0,082	0,083	0,082	0,073
0,40	0,049	0,060	0,069	0,073	0,082	0,071	0,067
0,43							
0,00	0,036	0,061	0,075	0,075	0,073	0,078	0,066
0,08	0,060	0,069	0,080	0,083	0,084	0,083	0,077
0,16	0,052	0,061	0,079	0,080	0,082	0,081	0,073
0,24	0,053	0,061	0,078	0,083	0,083	0,082	0,073
0,32	0,052	0,061	0,078	0,083	0,084	0,083	0,074
0,40	0,050	0,051	0,070	0,074	0,074	0,073	0,065
0,52							
0,00	0,036	0,061	0,075	0,075	0,073	0,078	0,066
0,08	0,060	0,069	0,080	0,083	0,084	0,083	0,077
0,16	0,053	0,060	0,078	0,080	0,082	0,081	0,072
0,24	0,052	0,061	0,078	0,083	0,083	0,082	0,073
0,32	0,052	0,061	0,078	0,083	0,084	0,083	0,074
0,40	0,048	0,052	0,054	0,057	0,057	0,056	0,054
0,59							
0,00	0,036	0,061	0,075	0,075	0,073	0,078	0,066
0,08	0,060	0,069	0,080	0,083	0,084	0,083	0,077
0,16	0,052	0,061	0,081	0,080	0,082	0,081	0,073
0,24	0,051	0,061	0,080	0,082	0,082	0,081	0,073
0,32	0,052	0,061	0,080	0,081	0,082	0,081	0,073
0,40	0,049	0,052	0,071	0,075	0,075	0,074	0,066
Total général	**0,050**	**0,061**	**0,076**	**0,079**	**0,080**	**0,079**	**0,071**

Tableau 7.III – Métrique `ROUGE-4` avec coefficient ajustés, Chaque sous-tableau contient les résultats pour différentes valeurs de λ_c, les colonnes contiennent les résultats pour différentes valeurs de λ_r et les lignes contiennent les résultats pour différents ratios entre les phrases extraites des citances et des références.

Nous avons repris les tests avec des valeurs comprises dans ces intervalles plus petits. Les résultats de ces deuxièmes tests apparaissent au tableau 7.III. Nous obtenons de meilleurs résultats pour les valeurs suivantes :

- k : 0.08
- λ_c : 0.52
- λ_r : 1.00

Le ratio k indique que les meilleurs résumés sont composés à 8% de mots provenant des citations et le reste de l'article lui même. Ce qui semble normal, puisque le résumé de référence est construit par un humain à partir de l'article sans tenir compte des citances. Les phrases du résumé de référence seront donc plus similaires à celle de l'article. Le coefficient λ_r de 1.00 indique qu'il est préférable de favoriser les phrases qui sont similaires à la phrase requête (le titre de l'article) même si elles ne contiennent pas d'information nouvelle. La valeur du coefficient λ_c a peu d'importance puisque seulement 8% des phrases du résumé final s'y rapportent. Cela nous a permit d'obtenir un `ROUGE-4` de 0.084. Lors de la compétition `scisumm 2016`, les autres équipes ont obtenues des valeurs entre 0.035 et 0.117. Cela nous classe troisième sur les cinq équipes.

7.3 Présentation des résultats

Nous avons créé une page `HTML` pour faciliter la consultation des résultats de nos analyses : les citances, les articles et les résumés construits (Voir les figures 7.2 et 7.3). Cette page contient un menu permettant la sélection des sujets (topics). Lorsqu'un sujet est choisi, la page est divisée en deux, à gauche nous avons accès aux articles et à droite, nous avons les résumés construits par notre système. Des menus permettent de choisir les articles et le résumé à afficher. Les textes sont affichés à raison d'une phrase par ligne. Les citances sont mises en évidence par une couleur de fond jaune.

Chaque phrase du résumé contient un lien vers la phrase originale. Chaque article contient un lien vers le fichier `pdf` maintenu par le site de l'ACL Anthology. Leur site offre une table de liens normalisés pour les références externes vers leurs articles. Il suffit de préfixer le numéro d'identification de l'article (par exemple P05-1004)

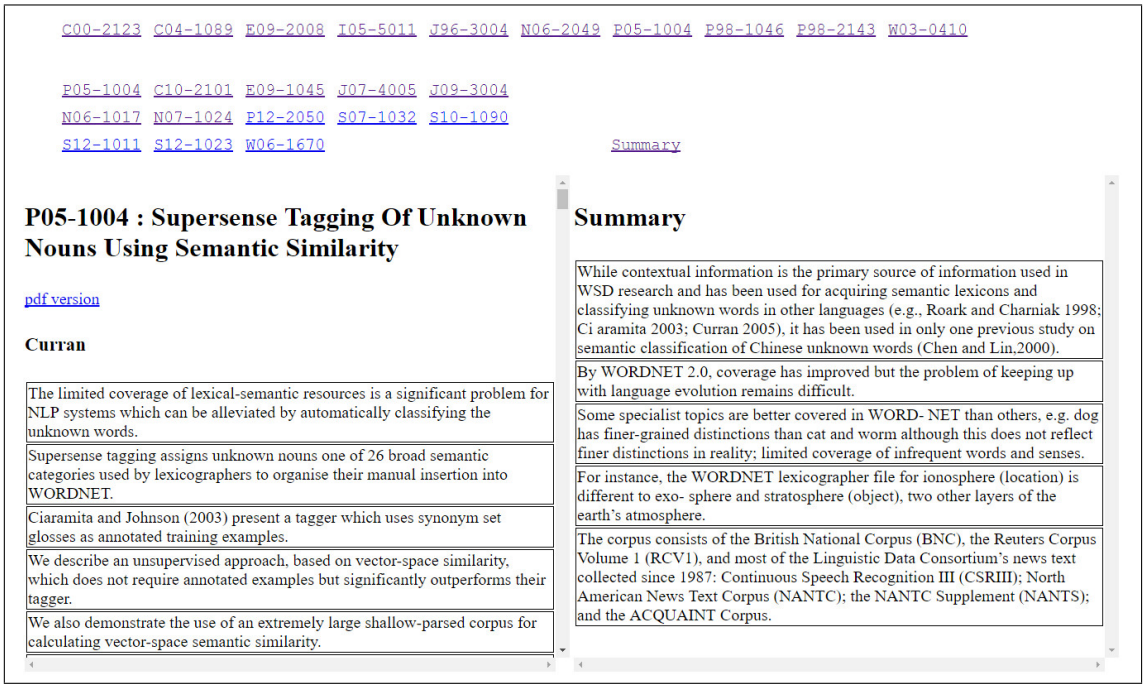

Figure 7.2 – Capture d'écran de notre interface HTML. L'image montre le RP et son résumé. Le menu du haut contient la liste des différents sujets. Le menu du dessous contient la liste des articles inclus dans ce sujet.

par l'adresse `http://aclweb.org/anthology/`. Notre document `HTML` a été généré à l'aide de la librairie `markup` de `python`.

Dans ce chapitre, nous avons présenté les résultats de l'évaluation du système `Citatum` pour la détermination des facettes et la construction automatique de résumé. Notre système atteint un taux d'accord de 47.5% avec les annotateurs pour la détermination des facettes sur un maximum possible de 66.6%. Le maximum possible nous amène à poser des questions importantes sur les objectifs d'une telle tâche :

- Les cinq facettes sont-elles exclusives ?
- S'il est difficile pour des chercheurs d'attribuer la même facette à une citation, est-ce valide (voir utile) de construire un système automatique pour faire cette annotation ?

Nous croyons que l'attribution d'une facette à une citation et aux phrases d'un

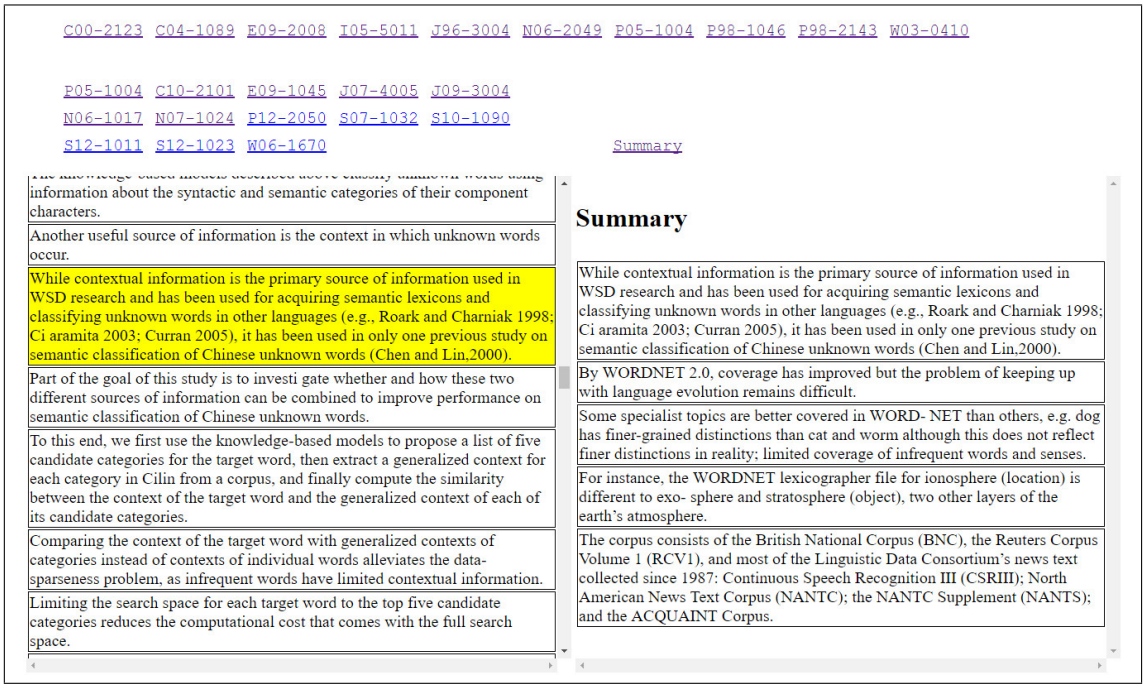

Figure 7.3 – Capture d'écran de notre interface HTML. L'image montre un CP dans lequel nous voyons la citance qui a été choisie pour la première phrase du résumé.

article reste une tâche importante afin de garder de l'information sur le contexte rhétorique. Il serait important de déterminer un ensemble de facettes exclusives entre elles et peut-être même d'avoir des facettes dont la determination seraient plus objectives que subjectives.

Finalement, La métrique `ROUGE` à été utiliée afin de comparer les résumés de `Citatum` aux résumés attendus. Cette comparaison est intervenue dans deux contextes : ajuster les coefficients de notre système et évaluer les résultats du système. Cela nous a permis de déterminer qu'un résumé de 250 mots devrait contenir 8% de phrases provenants des citations. Aussi, pour les phrases provenant du texte de l'article à résumer, il est préférable de donner priorité aux phrases similaires au titre de l'article sans tenir compte de leur similitude avec les autres phrases du résumé. Avec ces coefficients, notre système c'est classé troisième sur cinq à la compétition `scisumm 2016`. Finalement, nous avons présenté notre interface `HTML` généré par `Citatum`. Cette interface permet la navigation des résumés générés avec les articles RP et CPs qui ont été utilisés pour la construction du résumé.

CHAPITRE 8

CONCLUSION

Nous voulions que notre système (`Citatum`) assiste les chercheurs dans leur tâche, plus particulièrement la lecture d'articles scientifiques, que ce soit pour les comparer, pour identifier de nouveaux problèmes, pour situer son travail dans la littérature courante ou pour définir des propositions de recherche [16]. Un chercheur doit donc rapidement repérer la contribution d'un article scientifique. Ce repérage va demander la lecture et compréhension de plusieurs articles similaires. Comme nous l'avons vu au chapitre 2, le résumé automatique d'articles scientifiques existait déjà. Nous voulions modifier ces techniques pour prendre en compte l'impact qu'un article a eu sur la communauté scientifique. Qazvinian, et al. [28] montrent qu'il est possible d'extraire cette information du texte des citations d'un article. Nous voulons que `Citatum` effectue le traitement complet des articles scientifiques, c.-à-d. qu'à partir du texte des articles, qu'il analyse les méta-données afin de choisir les articles à résumer, qu'il extraie les citations, qu'il les annote et que finalement il construise un résumé de l'article. À notre connaissance, `Citatum` est le seul système effectuant toutes ces étapes.

Le corpus que nous avons utilisé contient des articles du domaine du traitement des langues naturelles. Il serait intéressant d'essayer notre système sur des corpus provenant d'autres domaines. Pour un domaine donné, il faudrait vérifier que les citations y jouent un rôle similaire. Si ce n'est pas le cas, alors il faudrait déterminer la signification qu'aurrait le résumé construit à partir des citations de ce domaine.

`Citatum`, notre système, présente les résultats finaux sous forme d'une page `HTML`. Entre autres, cette page contient les citances et les résumés construits par le système. Ces résumés sont construits à partir des citances extraites des articles citant et des phrases de l'article de référence. Il existe déjà des systèmes construisant les résumés à partir des phrases d'un article alors que, Qazvinian, et al. [28] ont construit des résumés à partir des citations. L'originalité de notre système est de combiner ces deux sources d'informations pour la construction du résumé. Les citations nous donnent de l'information sur l'impact qu'à eu l'article et les extraits de l'article complètent cette information. À notre connaissance, `Citatum` est le seul système utilisant les deux sources pour la construction de résumé. Nous avons utilisé l'équation de MMR pour

choisir les phrases qui contenaient le plus d'information différente afin de composer le résumé. L'évaluation de ces résumés à l'aide de la métrique `ROUGE-4` comparant avec les résumés produits par des humains nous place en troisième place sur cinq participants à la compétition CL-2016.

Nous avons aussi montré comment notre système extrait les facettes des phrases et des citances afin d'identifier leurs rôles rhétoriques. Pour l'instant ces facettes ne sont pas utilisées pour construire nos résumés. Éventuellement, cette information sera ajoutée pour construire des résumés avec des objectifs plus spécifiques. Par exemple, pour un article, nous pourrions construire un résumé décrivant les objectifs de l'article et un autre décrivant les résultats obtenus. Tel que mentionné au Chapitre 2, un ensemble d'articles sur un sujet particulier représente un discours entre plusieurs auteurs. C'est une suite d'arguments dont la continuité est représentée par les citations et références entre articles. Nous croyons qu'il serait possible d'utiliser les contextes rhétoriques des citations afin de construire un résumé représentant cette suite d'arguments. Par exemple, en utilisant des contextes similaires à ceux proposés par Simone Teufel (OTHER, CONTRAST et BASIS [37]), nous pourrions indiquer dans le résumé ce qu'un article vient contredire, appuyer, améliorer ou utiliser d'un autre article. Cela nous donnerait des résumés non pas d'un seul article, mais de plusieurs articles représentant l'évolution d'un sujet dans le temps. Il serait intéressant de faire plus d'expériences avec le Lexitrans. Pour cela, il nous faudrait un corpus multi-disciplinaire annoté pour mesurer les performances sur plusieurs domaines.

Notre système de résumé utilisait le corpus de l'ACL. Nous avons augmenté ce corpus en y joignant les données de l'AAN. L'information de ces corpus à été transformée en `XML` pour les articles et en `RDF` pour la méta-information afin de la rendre plus accessible. Nous avons automatisé l'analyse de cette information afin d'extraire des groupes d'articles à résumer. Entre autres, nous avons utilisé un algorithme de type PageRank pour mesurer l'importance d'un article.

Notre technique pour trouver les phrases référées par les citances n'a malheureusement pas produit de très bons résultats. Une discussion et comparaison avec les autres participants ayant utilisé des métriques de similarité pour cette tâche nous indique que les métriques de similarité semblent inappropriées, car nous avons constaté que les résultats pour ces techniques plafonnent avec une métrique f1 de 0.07.

Notre système, à notre connaissance, est le premier à construire des résumés contenant à la fois des phrases extraites du texte et des citances. Les résultats sont présentés dans une interface contenant à la fois les citances et les documents cités ce

qui en facilite l'exploration et l'utilisation dans un contexte de revue de littérature ou de veille stratégique.

BIBLIOGRAPHIE

[1] Dominique Besagni, Abdel Belaïd et Nelly Benet. A Segmentation Method for Bibliographic References by Contextual Tagging of Fields. Dans *ICDAR '03 Proceedings of the Seventh International Conference on Document Analysis and Recognition*, volume 1, pages 384–388, 2003.

[2] Sergey Brin et Lawrence Page. The Anatomy of a Large-scale Hypertextual Web Search Engine. *Seventh International World-Wide Web Conference (WWW 1998)*, 30(1–7):107–117, 1998.

[3] Jaime G. Carbonell et Jade Goldstein. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. Dans *Research and Development in Information Retrieval - SIGIR*, pages 335–336, 1998.

[4] Kevin Bretonnel Cohen, Hoa Trang Dang, Anita de Waard, Prabha Yadav et Lucy Vanderwende. TAC 2014 Biomedical Summarization Track, may 2014. URL `http://www.nist.gov/tac/2014/BiomedSumm/`.

[5] John M. Conroy et Dianne P. O'leary. Text Summarization via Hidden Markov Models. Dans *Research and Development in Information Retrieval - SIGIR*, pages 406–407, 2001.

[6] Hercules Dalianis et Eduard H. Hovy. Aggregation in Natural Language Generation. Dans *European Workshop on Natural Language Generation - EWNLG*, pages 88–105, 1993.

[7] Patrick Drouin. Extracting a Bilingual Transdisciplinary Scientific Lexicon. Dans *Proceedings of eLexicography in the 21st Century : New Challenges, New Applications*, volume 7, pages 43–54. Presses universitaires de Louvain, Louvain–a-Neuve, 2010.

[8] Patrick Drouin. From a Bilingual Transdisciplinary Scientific Lexicon to Bilingual Transdisciplinary Scientific Colloations. Dans *Proceedings of the 14th EURALEX International Congress*, pages 296–305. Fryske Akademy, Leeuwarden/Ljouwert, Pays-Bas, 2010.

[9] H. P. Edmundson. New Methods in Automatic Extracting. *Journal of the ACM (JACM)*, 16(2):264–285, apr 1969.

[10] Günes Erkan et Dragomir R. Radev. LexRank : Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research - JAIR*, 22:457–479, 2004.

[11] Elena Filatova et Vasileios Hatzivassiloglou. Event-Based Extractive Summarization. Dans *ACL-04*, pages 104—-111, 2004.

[12] Pierre-Etienne Genest et Guy Lapalme. Task Knowledge in Abstractive Summarization. page 10 pages, 2015.

[13] C. Lee Giles, Kurt D. Bollacker et Steve Lawrence. CiteSeer :An Automatic Citation Indexing System. Dans *DL '98 Proceedings of the third ACM conference on Digital libraries*, pages 89–98, 1998.

[14] Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-Yen Kan et Kathleen R. McKeown. SIMFINDER : A Flexible Clustering Tool for Summarization, 2001.

[15] Kokil Jaidka, Chandrasekaran, Muthu Kumar, Rustagi, Sajal, Kan et Min-Yen. Overview of the 2nd Computational Linguistics Scientific Document Summarization Shared Task (cl-scisumm 2016). Dans *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)*, 2016.

[16] Kokil Jaidka, Christopher S.G. Khoo, Jin-Cheon Na et Wee Kim Wee. Deconstructing Human Literature Reviews – A Framework for Multi-Document Summarization. Dans *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 125—-135, 2013.

[17] Kevin Knight et Daniel Marcu. Statistics-Based Summarization - Step One : Sentence Compression. Dans *National Conference on Artificial Intelligence - AAAI*, pages 703–710, 2000.

[18] Julian Kupiec, Jan O. Pedersen et Francine Chen. A Trainable Document Summarizer. Dans *Research and Development in Information Retrieval - SIGIR*, pages 68–73, 1995.

[19] Chin-Yew Lin. ROUGE : A Package for Automatic Evaluation of Summaries. Dans *Workshop on Text Summarization Branches Out (WAS 2004)*, 2004.

[20] Inderjeet Mani, Barbara Gates et E. B. Eric Bloedorn. Multi-Document Summarization by Graph Search and Matching. *Computing Research Repository - CORR*, cmp-lg/971:622–628, 1997.

[21] Inderjeet Mani, Barbara Gates et Eric Bloedorn. Improving Summaries by Revising Them. Dans *Meeting of the Association for Computational Linguistics - ACL*, 1999.

[22] Rada Mihalcea, Courtney Corley et Carlo Strapparava. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. *AAAI*, 6:775–780, 7 2006.

[23] Preslav I. Nakov, Ariel S. Schwartz et Marti A. Hearst. Citances : Citation Sentences for Semantic Analysis of Bioscience Text. Dans *In Proceedings of the SIGIR'04 workshop on Search and Discovery in Bioinformatics*, 2004.

[24] Mark EJ Newman, Duncan J Watts et Steven H Strogatz. Random Graph Models of Social Networks. *Proceedings of the National Academy of Sciences*, 99 (suppl 1):2566–2572, 2002.

[25] Daraksha Parveen et Michael Strube. Integrating importance, non-redundancy and coherence in graph-based extractive summarization. Dans *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1298–1304, 2015.

[26] Brett Powley et Robert Dale. Evidence-based Information Extraction for High Accuracy Citation and Author Name Identification. Dans *RIAO '07 Large Scale Semantic Access to Content*, pages 618–632, 2007.

[27] Vahed Qazvinian et Dragomir R Radev. Identifying Non-Explicit Citing Sentences for Citation-Based Summarization. Dans *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 555–564. Association for Computational Linguistics, 2010.

[28] Vahed Qazvinian, Dragomir R. Radev, Saif Mohammad, Bonnie J. Dorr, David M. Zajic, M. Whidby et T. Moon. Generating Extractive Summaries of Scientific Paradigms. *JAIR*, 46:165–201, 2013.

[29] Dragomir R. Radev, Eduard Hovy et Kathleen McKeown. Introduction to the Special Issue on Summarization. *Computational Linguistics - Summarization*, 28(4):399–408, dec 2002.

[30] Dragomir R. Radev, Hongyan Jing, Magorzata Sty et Daniel Tam. Centroid-based Summarization of Multiple Documents. *Information Processing and Management - IPM*, 40(6):919–938, 2004.

[31] Dragomir R. Radev, Mark Thomas Joseph, Bryan Gibson et Pradeep Muthukrishnan. A Bibliometric and Network Analysis of the Field of Computational Linguistics. *Journal of the American Society for Information Science and Technology*, 2009.

[32] Dragomir R. Radev, Pradeep Muthukrishnan et Vahed Qazvinian. The ACL Anthology Network Corpus. Dans *Proceedings, ACL Workshop on Natural Language Processing and Information Retrieval for Digital Libraries*, Singapore, 2009.

[33] Dragomir R Radev, Pradeep Muthukrishnan et Vahed Qazvinian. The ACL Anthology Network Corpus. Dans *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 54–61. Association for Computational Linguistics, 2009.

[34] Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian et Amjad Abu-Jbara. The ACL Anthology Network Corpus. *Language Resources and Evaluation*, pages 1–26, 2013. ISSN 1574-020X. URL `http://dx.doi.org/10.1007/s10579-012-9211-2`.

[35] David Shotton. CiTO, the Citation Typing Ontology. Dans *J Biomed Semantics*, volume 1 de *6*, 2009.

[36] K. Sparck-Jones. Automatic Summarising : Factors and Directions. Dans I. Mani et M. Maybury, éditeurs, *Advances in Automatic Text Summarisation*, Cambridge MA., 1999. MIT Press.

[37] Simone Teufel. *The Structure of Scientific Articles : Applications to Citation Indexing and Summarization*. CSLI Studies in Computational Linguistics. CSLI Publications, 2010.

[38] Simone Teufel et Marc Moens. Summarizing Scientific Articles : Experiments with Relevance and Rhetorical Status. *Computational Linguistics - COLI*, 28(4): 409–445, 2002.

[39] Stephen Wan, Cécile Paris, Michael Muthukrishna et Robert Dale. Designing a Citation-sensitive Research Tool : An Initial Study of Browsing-specific Information Needs. Dans *NLPIR4DL '09 Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 45–53, 2009.

[40] Duncan J Watts et Steven H Strogatz. Collective Dynamics of 'small-world' Networks. *Nature*, 393:440–442, 1998.

[41] Michael White et Claire Cardie. Selecting Sentences for Multidocument Summaries using Randomized Local Search. Dans *Workshop on Automatic Summarization*, pages 9–18, jul 2002.

[42] Zhibiao Wu et Martha Palmer. Verbs Semantics and Lexical Selection. Dans *ACL '94 Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, 1994.

[43] Wenpu Xing et Ali A. Ghorbani. Weighted PageRank Algorithm. Dans *Conference on Communication Networks and Services Research - CNSR*, pages 305–314, 2004.

[44] Peter N. Yianilos et Kirk G. Kanzelberger. The Likeit Intelligent String Comparison Facility. Rapport technique, NEC Research Institute, 1997.

Annexe I

Métrique ROUGE

`ROUGE` (Recall-Oriented Understudy for Gisting Evaluation) est un ensemble de métriques pour évaluer les logiciels produisant des résumés [19]. Ces métriques mesurent la similarité entre deux résumés. L'ensemble de métriques `ROUGE` comprend quatre métriques : `ROUGE-N`, `ROUGE-L`, `ROUGE-W` et `ROUGE-S`

I.1 ROUGE-N

Cette métrique compare un résumé obtenu par le logiciel(résultat obtenu S) avec k résumés utilisés comme point de référence (résultats attendus S_i). Pour une valeur n donnée, les n-grammes de S et S_i sont construit. Ensuite un rappel est calculé entre les n-grammes de S et les S_i

$$\text{ROUGE}-\text{N} \quad = \quad \frac{\sum_{i=1}^{k} \sum_{g \in \text{ngramme}_{S_i}} \text{count}_{\text{match}}(g)}{\sum_{i=1}^{k} \sum_{g \in \text{ngramme}_{S_i}} \text{count}(g)} \tag{I.1}$$

La fonction count(g) indique le nombre d'occurrence de g dans S_i et $\text{count}_{\text{match}}(g)$ calcule le nombre d'occurence de g dans S_i et dans S et ensuite nous donne le minimum entre les deux valeurs. Il est à remarquer que cette métrique favorise les extraits d'un résultat qui se trouvent dans plusieurs résumés attendus, indiquant que cet extrait est plus commun.

I.2 ROUGE-L

La deuxième métrique utilise la longueur de la plus longue séquence de mots communs entre le résumé obtenu et un des résumés attendus. La plus longue sous-séquence commune(LSC) entre deux phrases (séquences de mots r et s) est une

séquence de mots t tel que tous les mots de t sont à la fois dans r et s, dans le même ordre La séquence t est choisie de telle sorte qu'elle ait une taille maximale.

Soit un résumé obtenu S contenant v phrases r_i et p mots au total ($p = \sum_{i=1}^{v} |r_i|$) et un résumé attendu S contenant u phrases s_i et totalisant m mots. Il est possible de caluler une mesure F du rappel (ρ) et de la précision (ϕ) au niveau de la plus longue sous-séquence entre les phrases du résumé obtenu et celles du résumé attendu.

$$\kappa_i = \frac{\left|\bigcup_{j=1}^{v} \mathrm{LSC}(s_i, r_j)\right|}{|s_i|} \tag{I.2}$$

$$\rho_L = \frac{\sum_{i=1}^{u} \kappa_i}{m} \tag{I.3}$$

$$\phi_L = \frac{\sum_{i=1}^{u} \kappa_i}{p} \tag{I.4}$$

$$F_L = \frac{(1+\beta^2)\rho_L \phi_L}{\rho_L + \beta^2 \phi_L} \tag{I.5}$$

Le coefficient β permet de placer l'emphase sur la précision ou le rappel. Lorsqu'il y a plus d'un résumé attendu, la métrique est calculée pour chaque résumé attendu et ensuite la valeur maximum est prise.

I.3 ROUGE-W

`ROUGE-L` donne le même pointage peu importe si les mots communs sont consécutifs ou non. Par exemple, si nous cherchons la phrase s composée des mots suivants : $s = v_1, v_2, v_3, v_4, v_5, v_6, v_7$. Les deux phrases suivantes obtiennent le même pointage selon `ROUGE-L` : $r_1 = v_1, v_8, v_3, v_9, v_5, v_{10}, v_7$ et $r_2 = v_1, v_2, v_3, v_8, v_9, v_7, v_{10}$. Or, nous voudrions attribuer un meilleur pointage pour la phrase r_2 puisqu'elle contient une sous-séquence de trois mots continus dans le même ordre que ceux de s.

Une fonction est utilisée pour donner un pointage à une sous-séquence commune selon sa longueur. Il est préférable d'avoir une fonction ayant la caractéristique $f(x+y) > f(x)+f(y)$ et dont l'inverse existe. Par exemple, $f(x) = x^2$. Un algorithme de programmation dynamique est utilisé afin de trouver la meilleur attribution (le meilleur pointage). Dans notre exemple, si nous comparons s et r_1, nous avons quatre

sous-séquences communes de longueur 1 chacune. ce qui nous donne un pointage de : $\text{WLCS}(s,r_1) = f(1) + f(1) + f(1) + f(1) = 4$. Par contre, entre s et r_2 nous obtiendrons : $\text{WLCS}(s,r_2) = f(3) + f(1) = 10$. Le calcul de la métrique final utilise une méthode similaire à `ROUGE-L`.

$$\rho_W = f^{-1}\left(\frac{\text{WLCS}(s,r)}{f(m)}\right) \tag{I.6}$$

$$\phi_W = f^{-1}\left(\frac{\text{WLCS}(s,r)}{f(p)}\right) \tag{I.7}$$

$$F_W = \frac{(1+\beta^2)\rho_W\phi_W}{\rho_W + \beta^2\phi_W} \tag{I.8}$$

I.4 ROUGE-S

La dernière métrique offerte par `ROUGE` vérifie les coocurences (skip-bigram) de mots entre deux phrases. Une paire de mots (v_1, v_2) sont cooccurents dans une phrase s'ils sont tout les deux dans la même phrase et que v_1 apparait dans la phrase avant v_2. La fonction SKIP2(r,s) calcule le nombre de cooccurences d'une phrase attendue r qui apparaissent dans la phrase obtenue s. Une métrique F est calculée à partir de cette fonction.

$$\rho_S = \frac{\text{SKIP2}(r,s)}{C(m,2)} \tag{I.9}$$

$$\phi_S = \frac{\text{SKIP2}(r,s)}{C(p,2)} \tag{I.10}$$

$$C(x,y) = \frac{x!}{(x-y)!y!} \tag{I.11}$$

$$F_S = \frac{(1+\beta^2)\rho_S\phi_S}{\rho_S + \beta^2\phi_S} \tag{I.12}$$