Université de Montréal

Résumés automatiques d'articles scientifiques basé sur les citations

par
Bruno Malenfant

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Rapport pour la partie orale
de l'examen pré-doctoral

décembre, 2014

Université de Montréal
Faculté des études supérieures

Cet examen pré-doctoral intitulé :

Résumés automatiques d'articles scientifiques basé sur les citations

présenté par :

Bruno Malenfant

a été évalué par un jury composé des personnes suivantes :

Marc Feeley	président-rapporteur
Guy Lapalme	directeur de recherche
Philippe Langlais	membre du jury

Examen accepté le :

Table des matières

Chapitre 1

Introduction

Mon projet de doctorat a pour objectif de construire et 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 [15]. 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 l'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.

L'analyse des citations pour le résumé d'articles scientifiques est assez récente et très d'actualité. Nous contribuerons à ce domaine dans le contexte de notre thèse de doctorat.

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.[24].

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.[18]

Des articles du même domaine répetent 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.

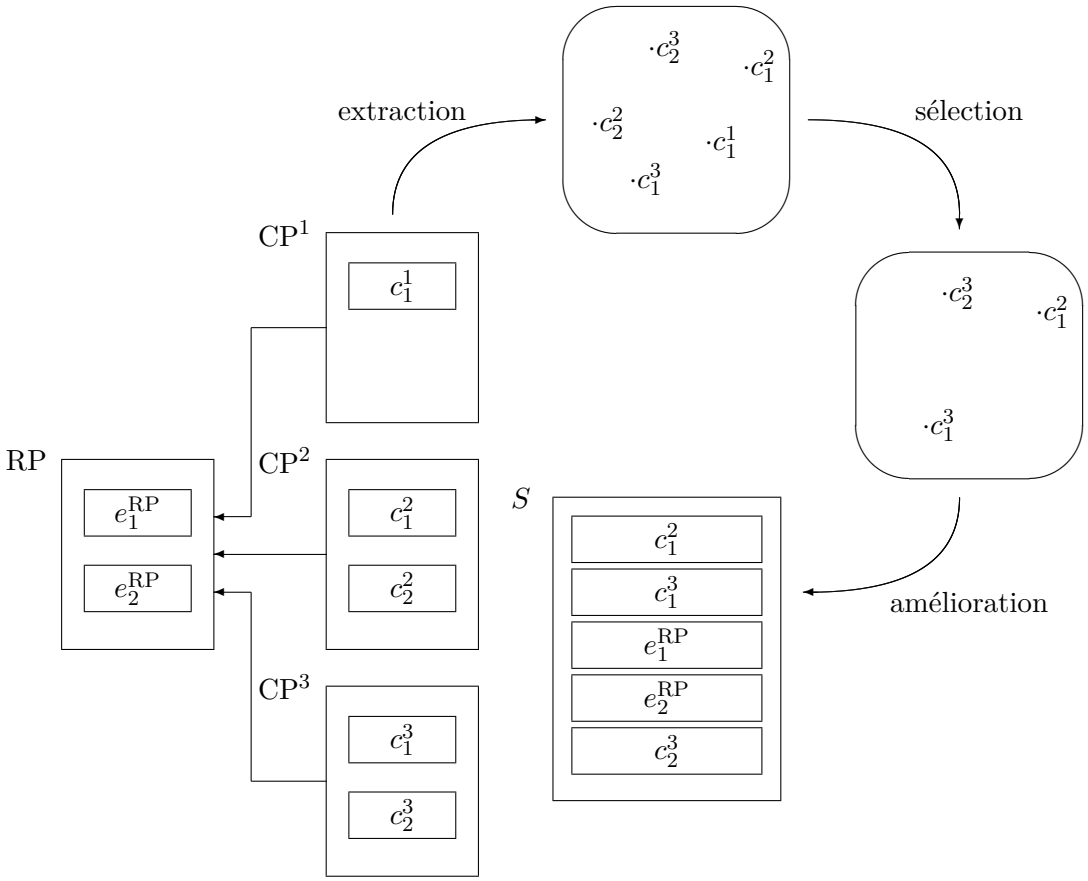
Plusieurs solutions informatiques sont utilisées pour aider les chercheurs. Les techniques de résumé automatique d'article scientifique simple ou multiple permettent de déterminer le sujet de l'article ou de plusieurs articles. Le résumé sur plusieurs articles n'est pas facile, comme les deux extraits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

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 leurs qualités. La compétition TAC 2014 propose de générer des résumés automatique d'article 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) [5]

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 *c*itance à été proposé par Preslav I. Nakov, Ariel S. Schwartz et Marti A. Hearst pour décrire l'ensemble des phrases entourant une citation[21]. Une citance est une partie d'un article scientifique qui référe à un autre article scientifique. Ce sera donc une ou plusieurs phrases expliquant ce qu'un autre article à réalisé dans un domaine lié à l'article citant. Dans son étude, Simone Teufel [28] 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 le TAC 2014 : HYPOTHÈSE, MÉTHODE, RÉSULTATS, IMPLICATION, DISCUSSION, et DONNÉES

[5]. 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.

Dans la suite du document, nous allons présenter l'état de l'art (chapitre 2) dans le domaine du traitement automatique de la langage appliqué aux textes scientifiques. Le chapitre 3 décrira l'approche que nous préconisons. Le chapitre 4 présentera nos premières expérimentations. Le chapitre 5 proposera un échéancier pour la poursuite de notre thèse.



U : une collection de documents.

RP : (reference paper) le document à résumer.

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

S : (summary) le résumé de RP.

e_j^i : 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. ($e_j^i \in \text{CP}^i$).

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 ($c_j^i = \{e_b^a\}$).

FIGURE 1.1 – Chaîne de traitement

Chapitre 2

Revue de littérature

Ce chapitre explore différentes technologies développées dans le domaine de l'analyse de la langue naturelle qui se rapprochent du 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 résumé d'articles multiples. Dans un premier temps, les différentes études sur la structure des articles scientifiques (section 2.1) seront présentées. Ensuite, les méthodes de résumés d'article seront présentées (section 2.2). Suivra une section sur l'extraction de citations et de leurs contextes (section 2.3). Finalement, quelques méthodes pour les résumés d'articles multiples seront décrites (section 2.4).

2.1 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.[24]

Dans son étude, Simone Teufel [27] énumère différents contextes rhétoriques qui peuvent être attribués à une phrase dans un article scientifique. Elle 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 ces observations, elle définit différents attributs 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 attributs OTHER, CONTRAST et BASIS qui vont être les plus importants.

Dans leur article, Simone Teufel et Marc Moens [28] 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[28] 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 contient pas d'information pertinente.

Un ensemble différent de caractéristiques est proposé pour le TAC. Cohen et al. 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. Les facettes qu'ils proposent reflètent un tel choix : HYPOTHÈSE, MÉTHODE, RÉSULTATS, IMPLICATION, DISCUSSION et DONNÉES.

2.2 Résumé d'article

Dans son article de 1999, Karen Spärck Jones[26] donne 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 sections suivantes décrivent les méthodes extractives et abstractives pour la construction automatique de résumé.

2.2.1 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.

H. P. Edmundson[10] 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 la redondance (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.

Julian Kupiec, Jan O. Pedersen et Francine Chen utilisent des principes similaires pour construire un extracteur avec apprentissage automatisé[17]. Leur système calcule la probabilité qu'une phrase appartienne à un résumé dépendant 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 pré identifiés.
— Pour les dix premiers paragraphes et les cinq derniers, chaque phrase reçoit un attribut indiquant si elles sont 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'être 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[6] ont proposé une méthode utilisant les chaînes de Markov pour décider si une phrase devrait être 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 s'il est dans le document.

Filatova et Hatzivassiloglou[12] 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 maximum 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 qui contiennent le plus grand poids et qui ne sont pas encore dans la solution. Parmi celles-ci, choisir celle qui a 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 il faut retourner à l'étape 2.

2.2.2 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 abstractif vont construire de nouvelles phrases à partir de l'information extraite. Cela implique une manipulation syntaxique (section 2.2.3) ou sémantique du texte choisi. Dans le contexte de cette recherche, nous allons nous concentrer sur les manipulations syntaxiques, les méthodes sémantiques demandent un ensemble différent de manipulation.

2.2.3 Manipulation syntaxique

Kevin Knight propose une manipulation afin de réduire la longueur des phrases d'un texte[16]. Pour cela, 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 logiciel construit des arbres représentant les phrases longues. Ensuite, plusieurs petits arbres qui pourraient représenter la phrase d'origine sont construits. Le logiciel essaie de reconstruire 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, Inderjeet Mani, Barbara Gates et Eric Bloedorn[19] 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[7]. 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 à 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.

2.3 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.

Stephen Wan et.al.[29] 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.

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 ...).

Dominique Besagni, Abdel Belaïd, et Nelly Benet présentent dans leurs articles de 2003[2] une méthode basée sur la reconnaissance des *p*arties d'un discours. Ils commencent avec des articles numérisés et sauvegardés en XML et vont identifier différents mots du texte selon les caractères les composant (lettre, chiffre, case ...). Ensuite, ils utilisent des patrons de reconnaissance pour identifier les différentes sections d'une référence. Lorsque certaines sections ont été identifiées, le système peut déduire les sections restantes.

Vahed Qazvinian et Dragomir R.Radev vont considérer un système où chaque phrase est représentée par une variable aléatoire indiquant si la phrase appartient à une citance. Ils vont utiliser des champs aléatoire de Markov comme modèle de graphe pour représenter les liens entre les variables aléatoires. Leurs graphes va lier les variables entre elles lorsque les phrases du texte sont voisines. Un algorithme de propagation des croyances (*Belief Propagation*) à travers le graphe va tenter de faire converger les valeurs (pointages). Les phrases ayant un pointage de 1 seront choisies pour faire partie de la citance.

Extraction des références/citations simultanément

Afin d'augmenter la précision de l'extraction des citations et des références dans un article scientifique, Brett Powley et Robert 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[22]. 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 logiciel 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.

2.3.1 Site internet avec des systèmes d'extraction de citations

CiteSeer utilise des logiciels d'explorations (*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 (TF $\cdot$ 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 [31].

Enfin pour pouvoir suggérer des documents similaires à celui choisi par un utilisateur, CiteSeer utilise une dernière métrique : CC$\cdot$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 smilarité 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.

2.4 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 multiples. Le résumé d'articles multiples contient 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[24].

Jaime G. Carbonell et Jade 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. Cette technique permet d'éliminer la redondance.

Michael White et Claire Cardie proposent une procédure de recherche locale pour choisir un ensemble de phrases en optimisant la somme des rangs de chaque phrase[30]. 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 Dragomir R. Radev, Hongyan Jing, Magorzata Sty et Daniel Tam[25] 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 $\cdot$ IDF. L'outil CIDR calcule un centroïde à partir du premier document (CP^1) de l'ensemble de documents (U). Ensuite, les valeurs TF $\cdot$ 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 inférieur à un certain seuil alors le document est ajouté à l'ensemble. A 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 :

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

   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).

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

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

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

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 :

$$\begin{aligned} \text{score}'(e_a^i) &= \text{score}(e_a^i) - w_r R(e_a^i) \\ w_r &= \max_{e_j^i \in \text{CP}^i} (\text{score}(e_j^i)) \\ R(e_a^i) &= 2 \times \frac{|\text{words}(e_a^i) \cap \text{words}(e_b^i)|}{|\text{words}(e_a^i)| + |\text{words}(e_b^i)|} \end{aligned}$$

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.

Günes Erkan utilise des principes similaires[11]. 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.

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

où $\text{tf}(t_w, e_j^i)$ 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 (degré) et de leurs qualités (centralité).

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

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

2.4.1 Analyse des citations

Vahed Qazvinian, et all.[23] 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 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 Microsaft Academic Search)(section 2.3). 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 2.2.3).

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 2.4). Il est aussi proposé d'utiliser des règles afin d'améliorer un résumé(section 2.2.3). Le chapitre suivant va présenter notre proposition de recherche qui s'inspire de ces travaux.

Chapitre 3

Sujet de recherche proposé

Nous proposons de développer un système pour résumer l'impact d'un article scientifique (RP) en utilisant les citations comme base. Des documents (CP^i) citant cet article sont d'abord téléchargés pour être analysés. Les citations (c^i_j) de l'article de départ sont extraites avec le texte environnant. Un premier résumé est construit à l'aide de ces citations. Finalement, le résumé sera amélioré à l'aide de l'article de base.

Nous allons maintenant illustrer notre méthodologie à l'aide d'un exemple que nous avons fait manuellement. Dans notre exemple, nous allons utiliser comme RP un article d'Eugenio Moggi et Amr Sabry, *Monadic encapsulation of effects : a revised approach (extended version)* [20].

3.1 Extraction des articles

3.1.1 Extraction des articles, citations et références.

Pour construire la liste des citations, nous pouvons utiliser des moteurs de recherche comme celui de Microsoft (academic search) ou du Pennsylvania State University (CiteSeerX). Il serait aussi possible d'utiliser ACL Anthology : `http://aclweb.org/anthology`, archive produite par *The Association for Computational Linguistics*. Ensuite, nous devons distinguer parmi les articles trouvés ceux qui sont accessibles et en anglais.

Academic Search de Microsoft nous donne 26 citations pour ce RP dont 7 sont des doublons [1]. Voici ces citations avec le rôle attribué selon la classification de Simone Teufel et Marc Moens [28] :

1. L'annexe A contient les références pour les articles utilisés dans l'exemple.

id	**texte**	**rôle**
c_1^1	*Besides supporting generic results that can be instantiated to particular effects at little or no cost, monads allow for a clear delineation of the scope of effects [*].*	OTHER
c_1^2	*Moggi and Sabry [*] used operational techniques to prove the safety of the full ST monad with typed references.*	OTHER
c_1^3	*E.g. monads have appeared in an abstract modelling of the Java semantics and in region analysis [*], and they form the basis of functional-imperative programming in Haskell; indeed Haskell's do-notation is essentially the metalanguage of effects.*	OTHER
c_1^4	*Haskell's ST monad and its STRef values guarantees such memory safety [*] by using the same type variable s to tag both the type of memory references (STRef s a) and the type of computations using these references (ST s b). Not only are ST and STRef both abstract type constructors, but runST, the function for running ST computations, has a rank-2 type that universally quantifies over the type variable s and prevents it from 'leaking'.*	OTHER
c_1^5	*Moggi and Sabry [*] prove syntactic type soundness for encapsulated lazy state.*	OTHER
c_1^6	*The parameter r is a region label [*] and does not concern us here.*	CONTRAST
c_1^7	*This is a challenging problem; despite much work on monadic encapsulation (and on region-based memory menagement) since the introduction of runST in Haskell, some of it incorrect and most of it rather complex, previous work mostly addresses simple syntatic type soundness, rather than equations [*], though the region calculus has been given a relation-based semantics and studied using bisimulation.*	CONTRAST
c_1^8	*Secondly, side effects are cleanly encapsulated and equipped with a clearly delineated scope[*].*	OTHER
c_1^9	*It turns out that this problem is a simple instance of the monadic encapsulation problem[*].*	BASIS

c_1^{10}	*As described in the previous section, the design of FRGN takes inspiration from the work on monadic state[*].*	BASIS
c_2^{10}	*The second line of research on which we draw is the work done in monadic encapsulation of effects[*].*	BASIS
c_3^{10}	*The majority of this work has focused on effects arising from reading and writing mutable state, which we reviewed in Section 2. While recent work [*] has considered more general combinations of effects and monads, no work has examined the combination of regions and monads.*	CONTRAST
c_1^{11}	*The linguistic framework of Haskell is Moggi's monadic metalanguage, ml, which has served as the de facto standard for monadic languages[*].*	OTHER
c_2^{11}	*The idea of indexing state transformers is used in subsequent monadic languages ; it also inspires the higher-order type of a similar construct run in a monadic language of Moggi and Sabry[*].*	BASIS
c_1^{12}	*This style of presentation is directly inspired by the distinction between pure and monadic evaluation in [*].*	BASIS
c_1^{13}	*Intuitively, Code is the type of process code abstracted over the local type P and the local operations in O. Its definition mimics that given in [*] for monadic code.*	BASIS
c_1^{14}	*More recently, a type soundness result for a language with lazy state operations has been proved by Moggi and Sabry [*].*	OTHER
c_1^{15}	*The language FRGN is an extension of System F (also referred to as the polymorphic -calculus), adding monadic types and operations and taking inspiration from the work on monadic state [*].*	BASIS
c_2^{15}	*While recent work [*] has considered more general combinations of effects and monads, no work has examined the combination of regions and monads.*	CONTRAST

Ces citations ont été extraites des documents suivants :

3.2 Extraction des citations

Les sites *CiteSeerX* et *Academic Search* donnent la phrase contenant la citation de l'article. Il pourrait être intéressant d'ajouter quelques phrases avant et après. Par exemple, nous pouvons prendre les phrases à partir de la phrase contenant la citation jusqu'à la phrase contenant la citation suivante ou jusqu'à la fin du paragraphe. Il existe plusieurs techniques qui permettent d'extraire les citations d'un article. Celle présentée par Brett Powley et Robert Dale[22] donne

des résultats intéressants (section 2.3).

3.3 Construction du résumé

Le résumé sera construit en deux étapes : un résumé de base est construit à l'aide des citations et ensuite il est amélioré en utilisant l'article de base.

3.3.1 Production du résumé de départ

La construction du résumé de départ sera faite à l'aide des citations extraites. Il y a déjà peu de phrases au total, le résumé consistera donc à choisir des phrases dont le sens est différent plutôt que de choisir des phrases plus importantes. Nous pouvons utiliser un algorithme qui fusionne l'information similaire telle que proposée par Regina Barzilay, Kathleen R. Mckeown et Michael Elhadad[1]. Ils vont construire l'intersection entre plusieurs phrases similaires pour trouver ce qu'elles ont en commun.

Par exemple, si nous prenons les trois phrases suivantes :

id	**texte**
c_1^1	*Besides supporting generic results that can be instantiated to particular effects at little or no cost, monads allow for a clear delineation of the scope of effects [*].*
c_1^2	*Moggi and Sabry [*] used operational techniques to prove the safety of the full ST monad with typed references.*
c_1^5	*Moggi and Sabry [*] prove syntactic type soundness for encapsulated lazy state.*

Nous remarquons que les phrases c_1^1 et c_1^5 sont impliquées par c_1^2 et donc ne sont pas nécessaires au résumé. Il est à remarquer que l'automatisation d'une telle détection serait difficile et ne sera pas effectuée dans ce projet. Pour le résumé de l'article, nous obtenons les phrases suivantes pour la catégorie BASIS :

id	**texte**
c_2^{10}	*The second line of research on which we draw is the work done in monadic encapsulation of effects [*].*
c_2^{11}	*The idea of indexing state transformers is used in subsequent monadic languages ; it also inspires the higher-order type of a similar construct run in a monadic language of Moggi and Sabry [*].*

Les phrases suivantes ont été retenues pour la catégorie CONTRAST :

id	texte
c_1^7	*This is a challenging problem ; despite much work on monadic encapsulation (and on region-based memory menagement) since the introduction of runST in Haskell, some of it incorrect and most of it rather complex, previous work mostly addresses simple syntatic type soundness, rather than equations[*], though the region calculus has been given a relation-based semantics and studied using bisimulation.*
c_2^{15}	*While recent work [*] has considered more general combinations of effects and monads, no work has examined the combination of regions and monads.*

Et finalement, pour OTHER nous avons :

id	texte
c_1^2	*Moggi and Sabry [*] used operational techniques to prove the safety of the full ST monad with typed references.*
c_1^4	*Haskell's ST monad and its STRef values guarantees such memory safety [*] by using the same type variable s to tag both the type of memory references (STRef s a) and the type of computations using these references (ST s b). Not only are ST and STRef both abstract type constructors, but runST, the function for running ST computations, has a rank-2 type that universally quantifies over the type variable s and prevents it from 'leaking'.*
c_1^{11}	*The linguistic framework of Haskell is Moggi's monadic metalanguage, ml, which has served as the de facto standard for monadic languages [*].*

Nous remarquons que les extraits retenus pour BASIS seraient plus utiles pour faire un lien vers un résumé de l'article d'où ils sont extraits.

3.3.2 Amélioration du résumé

Lorsque le résumé de départ est terminé, il ne reste plus qu'à l'améliorer. Dans leur article, Inderjeet Mani, Barbara Gates et Eric Bloedorn[19], proposent une technique pour faire cette amélioration (section 2.4). Ils utilisent des règles qui transforment le résumé de base pour éliminer la répétition, compléter l'information et combiner des phrases.

Cette technique peut ajouter des phrases au résumé, par exemple pour RP les phrases suivantes pourraient être sélectionnées :

id	**texte**
e_1^{RP}	*We formalize the intended implementations as big-step operational semantics (which are referred to as dynamic semantics), then we prove type safety for three systems.*
e_2^{RP}	*We extend the pure untyped λ-calculus with a run-construct. Intuitively, when an interpreter for the λcalculus has to evaluate run e, it calls a monadic interpreter which evaluates e applied to an internal implementation of the monadic operations, and then evaluates the term returned by the monadic interpreter. The term e in run e should be considered abstract code, since it abstracts from the implementation of the monadic operations.*

Nous remarquons que dans ce cas-ci se sont des phrases décrivant le fonctionnement qui ont été ajoutées. Une autre idée est de construire un résumé de l'article de façon automatique et de l'améliorer à l'aide des citations. Après quelques essais, nous avons remarqué que le résultat final semble identique.

Par contre, il est avantageux de construire un résumé à partir des citations car cela nous permet d'avoir une idée des sujets importants de l'article. Cette information est très utile pour terminer la construction du résumé.

Voici le résumé final après l'ajout de ces phrases.

id	texte
c_1^2	*Moggi and Sabry [*] used operational techniques to prove the safety of the full ST monad with typed references.*
e_1^{RP}	*We formalize the intended implementations as big-step operational semantics (which are referred to as dynamic semantics), then we prove type safety for three systems.*
c_1^4	*Haskell's ST monad and its STRef values guarantees such memory safety [*] by using the same type variable s to tag both the type of memory references (STRef s a) and the type of computations using these references (ST s b). Not only are ST and STRef both abstract type constructors, but runST, the function for running ST computations, has a rank-2 type that universally quantifies over the type variable s and prevents it from 'leaking'.*
c_1^{11}	*The linguistic framework of Haskell is Moggi's monadic metalanguage, ml, which has served as the de facto standard for monadic languages [*].*
e_2^{RP}	*We extend the pure untyped λ-calculus with a run-construct. Intuitively, when an interpreter for the λcalculus has to evaluate run e, it calls a monadic interpreter which evaluates e applied to an internal implementation of the monadic operations, and then evaluates the term returned by the monadic interpreter. The term e in run e should be considered abstract code, since it abstracts from the implementation of the monadic operations.*
c_2^{15}	*While recent work [*] has considered more general combinations of effects and monads, no work has examined the combination of regions and monads.*
c_1^7	*This is a challenging problem; despite much work on monadic encapsulation (and on region-based memory menagement) since the introduction of runST in Haskell, some of it incorrect and most of it rather complex, previous work mostly addresses simple syntatic type soundness, rather than equations[*], though the region calculus has been given a relation-based semantics and studied using bisimulation.*

Les phrases BASIS qui n'ajoutent pas d'information au résumé ont été supprimé, par contre elles sont utiles pour identifier les articles qui font suites à celui-ci. Regardons maintenant comment une de ces citations peut être utilisée pour construire une revue de littérature.

3.3.3 Résumé d'un sujet : plusieurs articles

Pour construire un résumé plus complet sur un sujet nous devons ajouter l'information provenant des articles qui poursuivent sur le même sujet. Les citations de type BASIS représentent bien un tel lien. Par exemple les citations c_1^{10} et c_2^{10} sont de type BASIS.

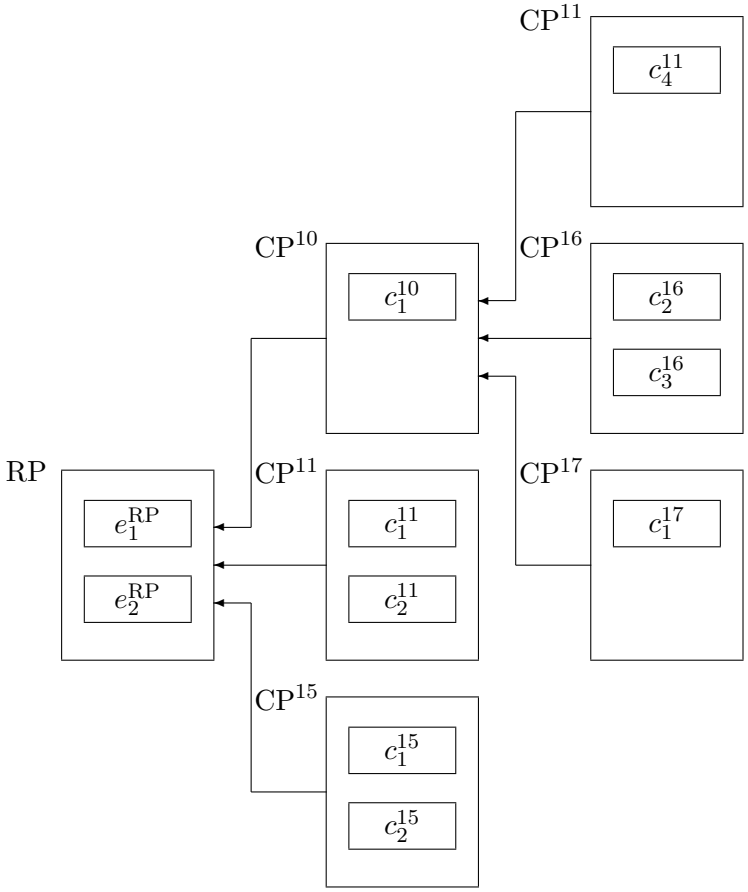


FIGURE 3.1 – Arbre de citations

c_1^{10}	*As described in the previous section, the design of FRGN takes inspiration from the work on monadic state [*].*
c_2^{10}	*The second line of research on which we draw is the work done in monadic encapsulation of effects [*].*

Ces citations expliquent la relation entre les deux articles et comment la recherche de CP^{10} est liée à la recherche présenté par RP. Nous pouvons appliquer à nouveau notre méthode de résumé pour CP^{10}.

id	**texte**
c_{12}^6	*As Fluet and Morrisett [*] put it, 'we consider a region to be a subtype of all the regions that it outlives.'.*
c_{14}^6	*Cyclone and Fluet and Morrisett's monadic regions [*] extend the region calculus with region subtyping : a resource allocated in an older region is available as if it were located in any younger region.*
e_1^{10}	*In this paper, we consider a monad family, called RGN, which does provide the necessary power to encode region calculi and back this claim by giving a translation from a core-Cyclone calculus to a monadic version of System F which we call FRGN.*
c_3^{16}	*Our previous work [*] gave an operational semantics for FRGN and proved the type soundness of FRGN. Since the introduction in the context of programming languages by Moggi and their popularization by Wadler, monads have become an established technique for reasoning and programming [*].*
c_1^{17}	*It would also be interesting to see whether other clever programming tricks, such as phantom types [*], could reduce this overhead.*
c_4^{11}	*For this reason, Fluet [*] uses a construct newRGN for creating new regions (in addition to another construct runRGN similar to runST) in his translation of a variant of the region calculus into an extension of System F with monadic types.*
c_3^{11}	*The idea of indexing state transformers is used in subsequent monadic languages [*]; it also inspires the higher-order type of a similar construct run in a monadic language of Moggi and Sabry.*
c_9^6	*Alas, as extensively discussed by Fluet and Morrisett [*], this solution lacks region polymorphism : within a child computation, we would like to use (even create) safe handles in any ancestor region, without nailing down the exact lineage of each handle.*
c_2^{16}	*And while the soundness of Cyclone's lexical regions and type-and-effects system has been established [*], a model that justifies the soundness of the new features has eluded our grasp, due to sheer complexity.*

Chapitre 4

Premières expérimentations

Au cours de la dernière année, nous avons participé à deux compétitions portant sur l'extraction de citations dans des articles scientifiques afin d'en déterminer les impacts. Ces initiatives démontrent l'intéret grandissant de la communauté scientifique pour ce domaine de recherche.

4.1 ESWC-14

La compétition ESWC-14 : *Semantic Publishing – Assessing the Quality of Scientific Output*[1] visait à produire et exploiter des données sémantiques sur les publications scientifiques. Cette compétition faisait partie de la onzième *Extended Semantic Web Conference*. Elle était composée de trois tâches : déterminer la qualité d'un workshop, déterminer la qualité d'un article et soumettre un outil innovateur pour chercher et représenter de l'information. Nous avons participé à la deuxième tâche. L'objectif était d'enrichir les articles avec de l'information sémantique permettant d'en évaluer la qualité.

Pour l'entrainement, nous avions à notre disposition 150 articles du domaine de la biologie extraits de 15 journaux. L'évaluation utilisait 400 articles, incluant les 150 articles pour l'entrainement. Ces fichiers étaient en format JATS, *Journal Article Tag Suite*, un standard XML pour encoder des articles et developpé par le *National Center for Biotechnology Information*. C'est un dérivé du NLM, *Archiving and Interchange Tag Suite*, aussi développé par le NCBI. Certains articles utilisaient aussi TaxPub, une extension officielle de JATS qui ajoute des balises taxonomiques et est développé par Terence Catapano[4].

Nous devions premièrement construire une base de données à l'aide de la méta information des articles et ensuite préparer des requêtes permettant de consulter ces informations.

1. http ://2014.eswc-conferences.org/semantic-publishing-challenge

```
<article>
    <front>
    ...
        <article-meta>
5           <article-id pub-id-type="doi">10.3897/zookeys.90.1121
            </article-id>
            <title-group>
                <article-title>New earthworm species of the genus
                  <italic>Amynthas</italic> Kinberg, 1867 from Thailand
10                (Clitellata, Oligochaeta, Megascolecidae)
                </article-title>
            </title-group>
            <contrib-group>
                <contrib contrib-type="author" xlink:type="simple">
15                  <name name-style="western">
                        <surname>Tongkerd</surname>
                        <given-names>Piyoros</given-names>
                    </name>
                </contrib>
20          </contrib-group>
            ...
        </article-meta>
    </front>
    <body>
25  ...
        <sec sec-type="Introduction"><title>Introduction</title>
        <p>
        ...
           (<italic>sensu</italic>
30         <xref ref-type="bibr" rid="B3">Blakemore 2000</xref>)
           ...
        </p>
        ...
        </sec>
35      ...
    </body>
    <back>
    ...
            <ref id="B3"><mixed-citation xlink:type="simple">
40          <person-group>
            <name name-style="western">
            <surname>Blakemore</surname>
            <given-names>RJ</given-names>
            </name>
45          </person-group>
            ...
    </back>
</article>
```

FIGURE 4.1 – Un extrait de l'article *New earthworm species of the genus Amynthas Kinberg, 1867 from Thailand (Clitellata, Oligochaeta, Megascolecidae)* par Bantaowong Ueangfa et al. (2011) en format JATS. Nous avons la méta-information (lignes 4 – 22), une citation 'B3' (ligne 30) et une référence bibliographique (lignes 39 – 45).

```
iri:B1P10740S4508A3160  rel:refere-a  iri:CA3160-B3
iri:CA3160-B3  rel:rid     B3 ;
               rel:acount  1 ;
               rel:cited   iri:A5205 ;
               rel:citing  iri:A3160 .
iri:A5205  rel:est-ecrit-par  iri:N680-2 .
iri:N680-2  rel:auteur-surname                "Blakemore" ;
            rel:auteur-first-name-initials    "RJ" .
iri:S4508A3160  rel:ref-count   24 ;
                rel:in-article  iri:A3160 ;
                rel:titre-sec   Introduction .
iri:A3160  rel:est-ecrit-par  iri:N7525-1 ,
                              iri:N7266-1 ,
                              iri:N5637-1 ,
                              iri:N396-1 ,
                              iri:N3425-10 ;
           rel:issue          0 ;
           rel:doi            10.3897/zookeys.90.1121 ;
           rel:article-titre  New earthworm species of the genus
                              Amynthas Kinberg, 1867 from Thailand
                              (Clitellata, Oligochaeta, Megascolecidae) .
iri:N7525-1  rel:auteur-surname              "Tongkerd" ;
             rel:auteur-first-name-initials  "P" ;
             rel:auteur-first-name           "Piyoros" .
```

FIGURE 4.2 – Exemple de 3 tuples TTL représentant la citation de la figure 4.1, avec la méta-information (lignes 12 – 24), une citation 'B3' (ligne 1)ttl :citf et une référence bibliographique (ligne 9)ttl :bibf.

4.1.1 Construction de la base de données

Nous devions premièrement construire une base de données. Les informations sont liées en utilisant des URI. Le RDF (Resource Description Framework) est un langage d'usage générique pour représenter l'information sur internet, il utilise une forme sujet – prédicat – objet, et est donc composé de triples décrivant des relations. Pour notre projet nous avons utilisé le format TTL (TurTLe), qui est une syntaxe textuelle pour le RDF.

Les articles XML contiennent plusieurs balises imbriquées. Pour placer toute l'information de plusieurs articles dans une seule base de données, nous devons générer des identificateurs uniques et nous en servir pour étiqueter l'information importante des articles. Cela demande de remonter les balises de l'arbre XML vers la racine, étape par étape. Dans notre cas, six scripts de transformations successives permettent de remonter les balises et de les étiqueter au moment où elles sont accessibles. Ces scripts ont été écris en format XSLT, permettant de décrire des transformations sur les arbres XML. Les scripts ont extrait 138 146 triples à partir des 150 articles d'entrainement. Le balisage intensif des articles originaux simplifie grandement

l'extraction de données. Il serait bien que l'usage d'un tel standard de balisage soit plus répandu. Cela permettrait d'extraire l'information sans avoir à construire des logiciels ne réussissant pas toujours à la retrouver.

4.1.2 Requêtes

Nous devions ensuite construire des requêtes pour effectuer les recherches suivantes :

1. Identifier les articles cités par l'article X.
2. Identifier les articles publiés dans des journaux, cités par l'article X.
3. Identifier les auteurs cités par l'auteur X.
4. Identifier les articles cités par l'article X et écris par le même auteur.
5. Identifier les articles cités plusieurs fois par l'article X.
6. Identifier les articles cités plusieurs fois dans un même paragraphe d'un article X.
7. Identifier l'agence subventionnant la recherche de l'article X.
8. Identifier la section *'literature review'* de l'article X.
9. Identifier les articles desquels l'article X prétend utiliser les méthodologies ou la théorie.
10. Identifier les articles desquels l'article X prétend étendre les résultats.

La compétition demandait que ces requêtes soient écrites en SPARQL (SPARQL Protocol and RDF Query Language). Ce langage de requête est construit pour permettre d'interroger autant des bases de données que des données de l'internet. Nous avons construit les requêtes fonctionnelles pour les six premières questions et la huitième. Les trois dernières questions demandaient l'analyse du texte autre que les balises, nous n'avons pas eu le temps de faire cette analyse. La version SPARQL de la première requête est présentée à la figure 4.3, les autres requêtes se retrouvent dans l'annexe B.

Nous avons rencontré un autre problème lors de la réception des articles pour l'évaluation ; nos scripts n'ont pas réussi à transformer l'information nouvellement reçue et nous n'avons donc pas soumis nos résultats. Malgré ce problème, cette expérience nous a appris l'importance du balisage des textes scientifiques afin de faciliter l'extraction d'informations. La compétition TAC-2014 nous a confirmé cette impression.

4.2 TAC-2014

Depuis 2001, le National Institute of Standards and Technology (NIST) organise des compétitions pour évaluer les nouvelles technologies en traitement de la langue naturelle. Au début,

```
SELECT ?referenceiri ?doi ?pmid ?title
WHERE
{
        FILTER ( ?X = "New␣earthworm␣...␣Megascolecidae)" ) .
                ?IRIciting rel:article-titre ?X .
                ?IRIreference rel:citing     ?IRIciting .
                ?IRIreference rel:cited      ?referenceiri .
                ?referenceiri  rel:article-titre ?title .
                OPTIONAL { ?referenceiri  rel:doi           ?doi } .
                OPTIONAL { ?referenceiri  rel:pmid          ?pmid }
}
```

FIGURE 4.3 – Requête pour la question 1 : Identifier les articles cités par l'article X.

sous le nom de *Document Understanding Conferences* (DUC), changeant pour le *Text Analysis Conference* en 2008, ces compétitions sont devenus une référence en résumé automatique. Le laboratoire Rali a participé à toutes ces compétitions, se classant souvent parmi les meilleures équipes.

La catégorie *Biomedical Summarization* de l'édition 2014 du TAC [2] est basée sur l'hypothèse que les citances (chapitre 1.1) peuvent être utilisées pour mesurer l'impact d'un article. Cette catégorie est nouvelle de cette année. Cette tâche demandait d'identifier une facette parmi HYPOTHESIS, METHOD, RESULTS, IMPLICATION, et DISCUSSION pour chaque citance. De plus, nous devions retrouver le texte de l'article cité qui se rapporte le mieux à la citance. Finalement, un résumé de l'article cité devait être construit à partir de cette information.

À la fin du mois de mai 2014, un premier ensemble d'entrainement fut distribué aux participants. Cet ensemble contenait 4 sujets composés chacun d'un article de référence RP et de dix articles CP^i contenant au moins une citation vers l'article de référence. Pour chaque article, nous avions une version pdf et une version texte. Contrairement aux articles du ESWC, ceux-ci n'étaient pas balisés en XML, mais n'étaient que du texte libre extraits automatiquement de PDF.

Un deuxième ensemble d'entrainement, distribué au milieu du mois d'août, ajouta 16 sujets, nous donnant 20 sujets (220 articles) pour entrainer notre système. Chaque sujet venait avec des fichiers d'annotations. Ces annotations, au nombre de quatre et faites par des humains, indiquaient les citations, leurs facettes et leurs sections de références. Aussi, quatre résumés rédigés manuellement étaient fournis avec chaque sujet.

L'évaluation des systèmes a été effectuée sur un corpus de 30 sujets reçu le 17 septembre. Nous avons soumis nos résultats le 29 septembre. Un article décrivant notre système est fourni

2. http ://www.nist.gov/tac/2014/BiomedSumm/

dans l'annexe C de ce document.

4.2.1 Techniques utilisées

Lors de la réception des données, nous avons transformé les articles en format XML. Cette transformation s'avéra difficile. L'identification des citations et des éléments bibliographiques contenaient plusieurs cas différents, comme tout traitement d'information produite par l'humain. Nous avons finalement constaté que cette transformation n'aurait pas été nécessaire pour cette compétition car les citations avaient été identifiées dans les données. Par contre, nous trouvons utile d'avoir un outil qui l'effectue pour nos expérimentations futures.

Notre hypothèse pour identifier la facette d'une phrase est que les mots se rapportant au domaine spécifique de la biologie ne donnaient pas d'informations pertinentes. Pour distinguer les mots n'appartenant pas au domaine de la biologie, nous avons utilisé le *Lexique scientifique transdisciplinaire* (LST) de Patrick Drouin [8, 9]. Le LST est un ensemble de mots trouvés dans les écrits scientifiques et qui ne se rapportent pas à un domaine particulier.

Notre algorithme détermine la distribution des mots pour chaque facette en utilisant un histogramme. Ce calcul nous donne la somme des mots présents pour chaque facette. Pour trouver la facette d'une citance, nous additionnons le pointage de chaque mot pour chaque facette. La facette ayant le plus haut pointage est choisie.

Les pointages en entrainement pour l'identification d'une facette étaient comptés en trouvant le nombre d'annotateurs qui avaient la même facette et en divisant par le nombre d'annotateurs. Une moyenne de toutes les valeurs obtenues donne le pointage final. Notre système a atteint 47.2% sur un maximum possible de 66.7% pour identifier les facettes des citances. Ce maximum est calculé en prenant la facette la plus fréquement mentionnée pour chaque citation.

Afin de trouver la section de texte du RP référée par une citance, nous voulions associer une facette à chacune des phrases du RP. Un deuxième système a été entrainé pour cette tâche. Afin d'améliorer ce deuxième système, nous voulions réduire l'ensemble de mots pris dans le LST. Un algorithme génétique a été utilisé pour faire cette sélection. Cette approche nous a permis de passer d'un score de 50% à 57.7%.

Lorsque la facette de chaque phrase est connue, nous prenons les phrases avec la même facette que la citance et cherchons parmi celles-ci laquelle se rapporte à la citance. Pour trouver la bonne phrase, nous les représentons sous forme d'un sac de mots (bag of words). Une mesure de similarité est calculée en prenant la norme de l'intersection des sacs de mots avec un ensemble de mots choisis. En nous basant sur 2 caractéristiques, nous avons construit 15 ensembles de mots possibles pour cette comparaison. Nous avons pris les mots du résumé contextuel et/ou les mots de la citance et pour chaque cas, ceux appartennant au LST et/ou ceux ne lui appartenant

pas. Les pointages pour cette tâche sont calculés avec une métrique F1. Dans tous les cas, nous avons obtenu une métrique entre 0.03 et 0.04. D'autres tests ont été faits en ajoutant des points pour les phrases contenant le mot 'we', nous donnant une F1 entre 0.037 et 0.049. Ces pointages sont plus bas que les scores d'entrainement publiés par les autres équipes, se situant entre 0.16 et 0.20 avec un score de 0.31 pour l'équipe de Diego Mollá. Nous n'avons pas observé de différence marquante entre l'utilisation du LST ou non.

4.2.2 Résultats

Nous avons assisté à la conférence TAC 2014, mais les organisateurs avaient pris beaucoup de retard et les résultats de la compétition ne sont pas encore disponibles au moment de l'écriture de ce document. Neuf équipes ont participées à la compétition BiomedSumm. Les résultats semblent assez bons sur l'ensemble d'entrainement pour l'identification des facettes. Il ne semble pas que le NIST reprendra cette compétition dans le cadre du TAC, mais une autre compétition, le CLsummarization, a été organisé pour une première fois cette année. Cette tâche a une structure similaire à celle du TAC, mais sur un corpus de l'*Association of Computational Linguistics*.

Chapitre 5

Échéancier provisoire

Voici un échéancier approximatif des différentes étapes de ce projet :

1. janvier à avril 2013 : Réussite de la vérification du niveau de connaissance portant sur le contenu des cours IFT2015 et IFT2125(Prédoc 1). Réussite du cours IFT6281 : gestion de documents.
2. mai 2013 à avril 2014 : Préparation du document de présentation du sujet de doctorat.
3. septembre à décembre 2013 : Réussite au cours IFT6390, apprentissage machine : fondements.
4. novembre 2013 : Réussite du niveau de synthèse(Prédoc 2).
5. janvier à avril 2014 : Préparation à ESWC 2014.
6. mai à septembre 2014 : Préparation à TAC-2014.
7. septembre 2014 : Soumission des résultats TAC-2014.
8. novembre 2014 : Participation à la conférence TAC-2014.
9. décembre 2014 : Présentation du sujet de thèse.
10. janvier, février 2015 : Construction d'un logiciel pour extraire les citations.
11. mars 2015 : Trouver et utiliser un logiciel qui construit le résumé de départ.
12. avril à mai 2015 : Construire un logiciel qui effectue l'amélioration du résumé de départ.
13. juin à juillet 2015 : Tester le système et compiler les résultats.
14. août à décembre 2015 : Terminer l'écriture de la thèse.

Le projet devrait donc être complété pour la fin de l'année 2015.

Chapitre 6

Conclusion

Nous avons présenté le problème actuel en science, celui de se tenir à jour, malgré une quantité grandissante de publications. Il existe plusieurs techniques de résumé, tel que présenté dans ce document. Nous avons énuméré des techniques pour identifier le type de discours des phrases d'un article et comment construire des résumés extractifs. Aussi, nous avons vus des techniques de résumé multiples et d'extraction de méta-information.

Les articles scientifiques demandent un traitement spécial : pouvoir mesurer leurs impacts. Les compétitions auquels nous avons participées (ESWC14 et TAC2014) traitent toutes deux de ce sujet. Cela nous donne un indice sur l'importance que la communauté scientifique porte vers le résumé automatique des articles scientifiques. Notre participation à ces compétitions nous à permis de développer des techniques intéressantes pour extraire de l'information des articles. Nous avons décrits ces techniques et les avons présentés dans le contexte du TAC 2014. Nous croyons pourvoir ajouter aux techniques existantes lors de notre doctorat.

Bibliographie

[1] Regina Barzilay, Kathleen R. Mckeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In *Meeting of the Association for Computational Linguistics - ACL*, pages 550–557, 1999.

[2] Dominique Besagni, Abdel Belaïd, and Nelly Benet. A segmentation method for bibliographic references by contextual tagging of fields. In *ICDAR '03 Proceedings of the Seventh International Conference on Document Analysis and Recognition*, volume 1, pages 384–388, 2003.

[3] Jaime G. Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Research and Development in Information Retrieval - SIGIR*, pages 335–336, 1998.

[4] Terence Catapano. Taxpub : An extension of the nlm/ncbi journal publishing dtd for taxonomic descriptions. In *Journal Article Tag Suite Conference (JATS-Con) Proceedings 2010*. National Center for Biotechnology Information (US), 2010.

[5] Kevin Bretonnel Cohen, Hoa Trang Dang, Anita de Waard, Prabha Yadav, and Lucy Vanderwende. Tac 2014 biomedical summarization track, may 2014.

[6] John M. Conroy and Dianne P. O'leary. Text summarization via hidden markov models. In *Research and Development in Information Retrieval - SIGIR*, pages 406–407, 2001.

[7] Hercules Dalianis and Eduard H. Hovy. Aggregation in natural language generation. In *European Workshop on Natural Language Generation - EWNLG*, pages 88–105, 1993.

[8] Patrick Drouin. Extracting a bilingual transdisciplinary scientific lexicon. In *Proceedings of eLexicography in the 21st century : New challenges, new applications*, volume 7, pages 43–54. Presses universitaires de Louvain, Louvain–a-Neuve, 2010.

[9] Patrick Drouin. From a bilingual transdisciplinary scientific lexicon to bilingual transdisciplinary scientific colloations. In *Proceedings of the 14th EURALEX International Congress*, pages 296–305. Fryske Akademy, Leeuwarden/Ljouwert, Pays-Bas, 2010.

[10] H. P. Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2) :264–285, apr 1969.

[11] Günes Erkan and Dragomir R. Radev. Lexrank : Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research - JAIR*, 22 :457–479, 2004.

[12] Elena Filatova and Vasileios Hatzivassiloglou. Event-based extractive summarization. In *ACL-04*, pages 104Ű–111, 2004.

[13] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. Citeseer : an automatic citation indexing system. In *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, and Kathleen R. McKeown. Simfinder : A flexible clustering tool for summarization, 2001.

[15] Kokil Jaidka, Christopher S.G. Khoo, Jin-Cheon Na, and Wee Kim Wee. Deconstructing human literatur e reviews Ű a framework for multi-document summarization. 2013.

[16] Kevin Knight and Daniel Marcu. Statistics-based summarization - step one : Sentence compression. In *National Conference on Artificial Intelligence - AAAI*, pages 703–710, 2000.

[17] Julian Kupiec, Jan O. Pedersen, and Francine Chen. A trainable document summarizer. In *Research and Development in Information Retrieval - SIGIR*, pages 68–73, 1995.

[18] Inderjeet Mani, Barbara Gates, and E. B. Eric Bloedorn. Multi-document summarization by graph search and matching. *Computing Research Repository - CORR*, cmp-lg/971 :622–628, 1997.

[19] Inderjeet Mani, Barbara Gates, and Eric Bloedorn. Improving summaries by revising them. In *Meeting of the Association for Computational Linguistics - ACL*, 1999.

[20] Eugenio Moggi and Amr Sabry. Monadic encapsulation of effects : a revised approach (extended version). *Journal of Functional Programming - JFP*, 11(6) :591–627, 2001.

[21] Preslav I. Nakov, Ariel S. Schwartz, and Marti A. Hearst. Citances : Citation sentences for semantic analysis of bioscience text. In *In Proceedings of the SIGIRŠ04 workshop on Search and Discovery in Bioinformatics*, 2004.

[22] Brett Powley and Robert Dale. Evidence-based information extraction for high accuracy citation and author name identification. In *RIAO '07 Large Scale Semantic Access to Content*, pages 618–632, 2007.

[23] Vahed Qazvinian, Dragomir R. Radev, Saif Mohammad, Bonnie J. Dorr, David M. Zajic, M. Whidby, and T. Moon. Generating extractive summaries of scientific paradigms. *JAIR*, 46 :165–201, 2013.

[24] Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown. Introduction to the special issue on summarization. *Computational Linguistics - Summarization*, 28(4) :399–408, dec 2002.

[25] Dragomir R. Radev, Hongyan Jing, Magorzata Sty, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing and Management - IPM*, 40(6) :919–938, 2004.

[26] K. Sparck-jones. Automatic summarizing : factors and directions. In I. Mani and M. Maybury, editors, *Advances in Automatic Text Summarisation*, Cambridge MA., 1999. MIT Press.

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

[28] Simone Teufel and Marc Moens. Summarizing scientific articles : Experiments with relevance and rhetorical status. *Computational Linguistics - COLI*, 28(4) :409–445, 2002.

[29] Stephen Wan, Cécile Paris, Michael Muthukrishna, and Robert Dale. Designing a citation-sensitive research tool : an initial study of browsing-specific information needs. In *NLPIR4DL '09 Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries*, pages 45–53, 2009.

[30] Michael White and Claire Cardie. Selecting sentences for multidocument summaries using randomized local search. In *Workshop on Automatic Summarization*, pages 9–18, jul 2002.

[31] Peter N. Yianilos and Kirk G. Kanzelberger. The likeit intelligent string comparison facility. Technical report, NEC Research Institute, 1997.

Annexe A

Bibliographie des articles servant d'exemple.

id	référence
CP[1]	Goncharov, S. & Schröder, L. Powermonads and Tensors of Unranked Effects, *L*ogic in Computer Science - LICS, **2011**, *a*bs/1101.2, 227-236
CP[2]	Atkey, R. Syntax for Free : Representing Syntax with Binding Using Parametricity, *T*yped Lambda Calculus and Applications - TLCA, **2009**, 35-49
CP[3]	Goncharov, S. ; Schröder, L. & Mossakowski, T. Kleene Monads : Handling Iteration in a Framework of Generic Effects, *C*onference on Algebra and Coalgebra in Computer Science - CALCO, **2009**, 18-33
CP[4]	Kiselyov, O. & Chieh Shan, C. Lightweight monadic regions, *S*igplan Notices - SIGPLAN, **2009**, *4*4, 1-12
CP[5]	Benton, N. ; Kennedy, A. ; Beringer, L. & Hofmann, M. Relational semantics for effect-based program transformations with dynamic allocation, *P*rinciples and Practice of Declarative Programming - PPDP, **2007**, 87-96
CP[6]	Kiselyov, O. & chieh Shan, C. Delimited Continuations in Operating Systems, *C*onference on Modeling and Using Context - CONTEXT, **2007**, 291-302
CP[7]	Benton, N. ; Kennedy, A. ; Hofmann, M. & Beringer, L. Reading, Writing and Relations Towards Extensional Semantics for Effect Analyses, *A*sian Symposium on Programming Languages and Systems - APLAS
CP[8]	Goncharov, S. ; Schröder, L. & Mossakowski, T. Completeness of Global Evaluation Logic, *M*athematical Foundations of Computer Science - MFCS, **2006**, 447-458

CP[9]	Thiemann, P. User-level transactional programming in Haskell, *H*askell Workshop - Haskell, **2006**, 84-95
CP[10]	Fluet, M. & Morrisett, J. G. Monadic regions, *S*igplan Notices - SIGPLAN, **2004**, *3*9, 103-114
CP[11]	Sungwoo Park, R. H. A Modal Language for Effects, **2004**
CP[12]	Moggi, E. & Fagorzi, S. A Monadic Multistage Metalanguage, *F*oundations of Software Science and Computation Structure - FoSSaCS, **2003**, 358-374
CP[13]	Ferrari, G. L.; Moggi, E. & Pugliese, R. Global Types and Network Services, *E*lectronic Notes in Theoretical Computer Science - ENTCS, **2001**, *5*4, 35-48
CP[14]	Benton, N.; Hughes, J. & Moggi, E. Monads and Effects, *A*dvanced Courses - AC, **2000**, 42-122
CP[15]	Fluet, M. Monadic Regions (Extended Abstract)
CP[16]	Fluet, M. Morrisette, G. & Ahmed, A. J. Linear Regions Are All You Need, *E*uropean Symposium on Programming - ESOP, **2006**, 7-21
CP[17]	Tse, S. & Zdancewic, S. Designing a Security-typed Language with Certificate-based Declassification, *E*uropean Symposium on Programming - ESOP, **2005**

Annexe B

Requêtes SPARQL pour la compétition ESCW14

1. Identifier les articles cités par l'article X.

```
SELECT ?referenceiri ?doi ?pmid ?title
WHERE
{
        FILTER ( ?X = "paper title here" ) .
                ?IRIciting rel:article-titre ?X .
                ?IRIreference rel:citing     ?IRIciting .
                ?IRIreference rel:cited      ?referenceiri .
                ?referenceiri  rel:article-titre ?title .
                OPTIONAL { ?referenceiri  rel:doi           ?doi } .
                OPTIONAL { ?referenceiri  rel:pmid          ?pmid }
}
```

2. Identifier les articles publiés dans des journaux, cités par l'article X.

```
SELECT ?referenceiri ?doi ?pmid ?papertitle ?journaltitle
       ?journalvolume ?journalissue
WHERE
{
        FILTER ( ?X = "paper title here" ) .
                ?IRIciting     rel:article-titre  ?X .
                ?IRIref        rel:citing         ?IRIciting .
                ?IRIref        rel:cited          ?referenceiri .
                ?referenceiri  rel:article-titre  ?papertitle .
                ?referenceiri  rel:in-journal     ?IRIjournal .
                ?IRIjournal    rel:journal-titre  ?journaltitle .
```

```
                OPTIONAL { ?referenceiri  rel:volume  ?journalvolume } .
                OPTIONAL { ?referenceiri  rel:issue   ?journalissue } .
                OPTIONAL { ?referenceiri  rel:doi     ?doi } .
                OPTIONAL { ?referenceiri  rel:pmid    ?pmid }
}
```

3. Identifier les auteurs cités par l'auteur X.

```
SELECT ?resourceiri ?lastname ?firstname ?firstnameinitials
WHERE
{
        FILTER ( ?X = "Keklikoglou" ) .
                ?IRIcitingA   rel:auteur-surname              ?X .
                ?IRIciting    rel:est-ecrit-par               ?IRIcitingA .
                ?IRIref       rel:citing                      ?IRIciting .
                ?IRIref       rel:cited                       ?IRIcited .
                ?IRIcited     rel:est-ecrit-par               ?resourceiri .
                ?resourceiri  rel:auteur-surname              ?lastname .
                ?resourceiri  rel:auteur-first-name-initials
                              ?firstnameinitials .
                OPTIONAL { ?resourceiri  rel:auteur-first-name  ?firstname }
}
```

4. Identifier les articles cités par l'article X et écris par le même auteur.

```
SELECT ?resourceiri ?doi ?pmid ?title
WHERE
{
        FILTER ( ?X = "paper␣title␣here" ) .
        ?IRIciting  rel:article-titre  ?X .
        ?IRIciting  rel:est-ecrit-par  ?IRIauteur .
        ?IRIref     rel:citing         ?IRIciting .
        ?IRIref     rel:cited          ?resourceiri .
        ?resourceiri   rel:est-ecrit-par  ?IRIauteur .
        ?resourceiri   rel:article-titre  ?title .
        OPTIONAL { ?resourceiri  rel:doi  ?doi } .
        OPTIONAL { ?resourceiri  rel:pmid  ?pmid }
}
```

5. Identifier les articles cités plusieurs fois par l'article X.

```
SELECT ?resourceiri ?doi ?pmid ?title ?numberofcitations
WHERE
{
        FILTER ( ?X = "paper title here" ) .
        FILTER ( ?numberofcitations != 1 && ?numberofcitations != 0 ) .
        ?IRIciting    rel:article-titre ?X .
        ?IRIref       rel:citing        ?IRIciting .
        ?IRIref       rel:acount        ?numberofcitations .
        ?IRIref       rel:cited         ?resourceiri .
        ?resourceiri  rel:article-titre ?title .
        OPTIONAL { ?resourceiri  rel:doi   ?doi  } .
        OPTIONAL { ?resourceiri  rel:pmid  ?pmid }
} GROUP BY ?resourceiri ?doi ?pmid ?title ?numberofcitations
```

6. Identifier les articles cités plusieurs fois dans un même paragraphe d'un article X.

```
SELECT ?resourceiri ?doi ?pmid ?title ?numberofcitationsinparagraph
       ?paragraphiri ?paragraphxmlcontent
WHERE
{
        FILTER ( ?X = "paper title here" ) .
        FILTER ( ?numberofcitationsinparagraph != 1 &&
                 ?numberofcitationsinparagraph != 0 ) .
        ?IRIciting     rel:article-titre ?X .
        ?IRIref        rel:citing        ?IRIciting .
        ?IRIb          rel:refere-a      ?IRIref.
        ?IRIb          rel:pcount        ?numberofcitationsinparagraph .
        ?IRIb          rel:in-p          ?paragraphiri .
        ?paragraphiri  rel:contient      ?paragraphxmlcontent .
        ?IRIref        rel:cited         ?resourceiri .
        ?resourceiri   rel:article-titre ?title .
        OPTIONAL { ?resourceiri  rel:doi   ?doi  } .
        OPTIONAL { ?resourceiri  rel:pmid  ?pmid }
} GROUP BY ?resourceiri ?doi ?pmid ?title ?numberofcitationsinparagraph
           ?paragraphiri ?paragraphxmlcontent
```

7. Identifier la section *'literature review'* de l'article X.

```
SELECT ?sectioniri ?sectiontitle
WHERE
{
        FILTER ( ?X = "paper␣title␣here" ) .
        ?IRIa rel:article-titre  ?X .
        ?sectioniri rel:in-article ?IRIa .
        ?sectioniri rel:ref-count ?count .
        ?sectioniri rel:titre-sec ?sectiontitle .
}
ORDER BY DESC(?count) LIMIT 1
```

Annexe C

Article présenté dans le cadre de TAC 2014

Rali System Description For TAC 2014 Biomedical Summerization Track

Bruno Malenfant
Université de Montréal
CP 6128, Succ Centre-Ville
Montréal, Québec, Canada, H3C 3J3
malenfab@iro.umontreal.ca

Guy Lapalme
Université de Montréal
CP 6128, Succ Centre-Ville
Montréal, Québec, Canada, H3C 3J3
lapalme@iro.umontreal.ca

Abstract

We present our solution to the 2014 Biomedical Summerization Track. We propose a technique to determine the discourse role of a sentence. We differentiate words linked to the topic of the paper from the ones that link to the facet of the scientific discourse. Using that information, simple histograms are built over training data to infer a facet for each sentence of the paper (result, method, implication, hypothesis and discussion). This helps us isolate the sentences best representing a citation of the same facet.

1 Introduction

One's task in research is to read scientific paper to be able to compare them, to identify new problems, to position a work within the current literature and to elaborate new research propositions(Jaidka et al., 2013). This implies reading many papers before finding the ones we are looking for. With the growing amount of publications, this task is getting harder. It is becoming important to have a fast way of determining the utility of a paper for our needs. A first solution is to use web sites such as *CiteSeer, arXiv, Google Scholar* and *Microsoft Academic Search* that provide cross reference citations to papers. Another approach is automatic summarization of scientific paper. This year's TAC competition for summarization of biology papers proposes a community approach to summarization; it is based on the assumption that citances, the set of citation sentences to a reference paper, can be used as a measure of its impact. This task implies identifying a facet (discussion, result, method, implication and hypothesis) for each citance and the text it refers to in the reference paper. To solve this task, we propose to build sets of words that identify these facets. Our method takes into account words that are present in any scientific paper without consideration for the subject. Patrick Drouin (2010a; 2010a) developed such a set, the *Lexique scientifique transdisciplinaire* (LST).

To assess the facet and find the reference text, we have tested with different subsets of the LST with the hypothesis that words from the LST can help identify the facets of sentences in a paper and words outside of the LST represent the topics. Our experiments show that indeed the words from the LST are good indicators for identifying the facet of a passage.

We had already some experience in dealing with scientific papers and their references, having participated to Task2 of the Semantic Publishing Challenge of ESWC-2014 (Extended Semantic Web Conference) on the extraction and characterization of citations. A short review of previous work follows in Section 2. We will present the preprocessing steps we use over the data in Section 3 and the techniques for extracting information in Section 4. Finally, Section 5 will show our results.

2 Previous Work

There has been a growing attention toward the information carried by citations and their surrounding sentences (citance). These contain information useful for rhetorical classification (Advaith Siddharthan and Simone Teufel, 2007), for technical

surveys (Saif Mohammad et al., 2009) and for emphasizing the impact of papers (Qiaozhu Mei and ChengXiang Zhai, 2008). Qazvinian (2013) and Elkiss (2008) showed that citations provide information not present in the author's abstract.

Since the first works of Luhn (1958) and Edmundson (1969) many researchers have developed methods for finding the most relevant sentences of papers to produce abstracts and summaries. Many metrics have been introduced to measure the relevance of parts of text, either using special purpose formulas (Peter N. Yianilos and Kirk G. Kanzelberger, 1997) or using learned weights (Julian Kupiec et al., 1995). The hypothesis for TAC 2014 task is that important sentences can be pointed out by other paper : the citation indicates a section of paper that was considered important by a reader.

Another area of study of scientific papers is the classification of their sentences. Teufel (2002) identified the rhetorical status of sentences using Bayes classifiers.

To find citation inside paper, we need to analyse the references section. Dominique Besagni et al. (2003) developed a method using pattern recognition to extract fields from the references while Brett Powley and Robert Dale (2007) looked citations and references simultaneously using informations from one task to help complete the second task.

3 Building an XML Version of the Topics

Our first concern was to make sure that once we feed the texts into our system, we could reproduce the offsets found in the training data. As there were significant differences in the format between files because of the seemingly random appearance of byte-order marks, different types of line terminators, we had to develop scripts to make all of them uniform. When we worked on the ESWC-14 task to extract information of biological paper, we found useful to have the information in XML format which allow us to extract the information inside a program using existing XML tools. To simplify subsequent access to the data, we used a set of Python scripts to transform the 11 papers of each topic into a single XML file in which the papers were identified with their roles (Research Paper or Citation Paper).

3.1 Encoding Uniformisation

We wanted to make sure that all documents were following some basic format : all files to be utf-8, without BOM and using cr-lf for the EOL. We made a checker to see if the given offset from the annotation were the one we obtain from reading the files. This helped us to find files that need modification. The BOM and ASCII problems were solved by saving the file in another format. The EOL were rectified with a `sed` command.

3.2 Sections Identification

We transformed each topic into a simple XML file containing eleven documents : The research paper (RP) and ten citation papers (CP). Each paper was divided into two sections : body and references. Inside the body, we identified the position of the abstract. The body is further divided into paragraphs and the paragraphs into sentences. Inside a sentences we identified the citations and the tokens.

An important element of this transformation was to detect the references part of the paper. The words 'References', 'Selected reading' or 'Further reading' were usually present, indicating the beginning of the references section. For 6 papers, we had to manually insert the word 'References'. The lines before the references have been identified as the 'body' of the paper.

Each line was considered a paragraph. We used the Punkt sentences tokenizer from NLTK to separate sentences and words within them. We kept the offset for each paragraph, sentences and tokens The title of the paper is the first non-empty line that doesn't start with one of these words : Article, BMC, Cell, Cover, doi, Mol, PLoS, PMCID, Published, RESEARCH, Volume, Cancer Cell.

We needed to identify the abstract. Most of the time, it is preceded by 'abstract' or 'summary'. If these words aren't present then it follows 'Open Archive'. Sometimes we found the sentence 'Get rights and content' between 'Open Archive' and the abstract.

Inside the reference section we eliminate all lines beginning with 'View Record', 'View Article', 'Full Text', 'Article |' and 'Corresponding author'.

3.3 Reference Processing

The next step separates entities in the reference section. When a reference uses more than one line, it is separated by an empty line from the next one. If not, each line corresponds to one reference.

Once references are separated, we needed to extract their marker. Some used a number, other used the name of the authors with the year of publication. For each reference, we built two kinds of markers.

The marker is the concatenation of the name, a comma and the year, adding 'and' between the names when there are two of them or adding 'et al' at the end when there are more then two. Some author's name are composed of more then one word (for example : de Cristofaro, van Leuken), in these cases two different markers were created, one with the last name and one with the full name.

It is easier to extract the information for reference that used many lines. The first line is always a marker, but it isn't always the marker that is used in the text. The next line gives the authors, the title is found on the following line. The last line had the rest of the information (year, journal, ...). There were some variations to this scheme, for instance, when the reference uses only three lines there is no title.

3.4 Citation Identification

Citation identification was a bit tricky because of the many different conventions used for references and the citations (numbers in different brackets, authors' names, etc.). To find a citation, we first needed to determine if they use a numerical form or an author/year form, are they between parentheses, brackets or nothing? We started by with the hypothesis that the marker was in name/year form. With that in mind we searched for a marker in the paper, if that was not found, we searched for the numerical form. Each search was done with a regular expression that contained the marker in parentheses and in brackets. If it is a name/year marker then it is searched without parentheses or brackets.

After having performed all this clerical work for this year's task, we realised that the XML transformation was an overkill because citations had been already identified in the input data. Still, we believe that for any future work it is better to have access to formatted text and that the time invested in this part of our software will eventually be useful for other tasks.

4 Finding Facet and Extracting References

4.1 Facet Identification

In the first task, we identify the facet of a citance and the text it points to in the RP. by finding all sentences of the RP with the same facet. We hypothesized that the reference text should be within sentences sharing the same facet as the citance. We now present how we determine the facet of a citance, then the facet for sentences in the RP and finally the reference text.

We determine the word distribution for each facet using an histogram This computation yielded a sum of each words present in each facet. To find the facet of a citance we simply needed to add the score of words that were present for each facet. The facet with the highest score was chosen for that citance.

4.2 Finding the Sentences Refered to by Citances

For that task, we want to tag a facet to sentences in the RP. In the training data, annotators had identified a facet for citation and corresponding sentences. We assume that the same facet can be attributed to the sentences they choose for reference. Using this data we train the same system as before to identify the facet of the sentences inside the RP.

To find a better list of words to ignore, we used a genetic algorithm that uses a population of list of words to take out. Each new generation will build new list from the best one in the last generation. New lists were built by removing and adding a word to an already existing list. Other lists were built by merging two sublists of the last generation.

Once the facet of the sentences were known, we needed to choose a set of sentences that represented the citance. We compared the words of the sentences with same facet as the citance with a set of words that can mesure similarity with the citance. To find these words we built 15 sets over two features : words from the abstract and/or from the citances, words in the LST and/or words not in the LST. The 3 sentences with the highest number of words in common were chosen.

4.3 Summarization

For Task2, because of time constraints, we used a simple-minded approach: we merely extracted sentences with the greatest number of words from the LST also present in the citances.

5 Results

On the training set we obtained a success rate of 47.2% for finding the facet of citance. We tried taking out each word from the computation, finding those that contributed in a positive way and those who had a negative impact on the computation (table 1).

Word not used	Success rate
human	0.43
response	0.45
development	0.45
number	0.45
expression	0.48
small	0.48
function	0.49
as	0.49

Table 1: Success rate when a word is taken out.

The set of words $P = \{$'expression', 'small', 'function', 'as' $\}$ had a positive impact when they were not used for the computation. The set of words $N = \{$'human', 'response', 'development', 'number'$\}$ had a negative impact when they were not used for the computation. We tested using different combinations of words to exclude from the computation. The result can be seen in table 2.

For finding the facet of the RP sentences, using all the words from the LST we obtained a success rate of 57.7%. Testing for each word, we were able to find 25 words that had a positive contribution when they were used and 10 words that had a negative contribution. When all the negative words are taken out we obtained a success rate of 64.8% and when only the positive ones are used we obtained 67.2%. Over 15 generation of 1000 lists each, the genetic algorithm was able to reach a success rate of 74.4%.

For the reference extraction task we considered comparing against words from the abstract and from the citance. In each case we used words belonging to the LST or/and not into the LST. The F1 score for each case are presented in table 3. The best result comes from using words not from the LST that appear in the abstract. We did a second test where we multiplied the score of a sentence by a constant if it contained the word 'we', these result are shown in table 4.

Words not used	Success rate
A	0.47
P	0.44
N	0.52
$A - P$	0.53
$A - N$	0.43
$U - P$	0.59
$U - N$	0.55

Table 2: Success rate when many words are taken out, were A is the set of words encountered in each facet, U is the set of words encountered in at least one facet, $P = \{$'expression', 'small', 'function', 'as' $\}$, and $N = \{$'human', 'response', 'development', 'number'$\}$.

In this case using words that are common with the abstract, from the LST or not, yield the best result with a constant of 1.2.

6 Discussion

We used words from the LST to identify facet of sentences in a research paper. We had received 313 citations, each annotated by four annotators. The distribution of each facet is shown in table 5 and the number of times annotators all agreed on the same facet for one citation is shown in table 5. The facet **Result** and **Discussion** were the most often used and **Hypothesis** was almost never used.

Facet	Count	Distribution
Results	490	39%
Discussion	446	36%
Method	155	12%
Implication	140	11%
Hypothesis	21	2%

Table 5: Facet distribution over 313 citances and 4 annotators

So in the best case, choosing the facet that appear the most often, we could only obtain a score of 0.66 over the training data. Our script yielded a score of 0.50, agreeing in average with two annotators out

		Abstract			
		None	Not in LST	In LST	All
Citance	None	—	0.040	0.032	0.037
	Not in LST	0.038	0.031	0.032	0.032
	In LST	0.036	0.031	0.037	0.034
	All	0.037	0.033	0.033	0.034

Table 3: F1 for different cases.

		Abstract							
		None		Not in LST		In LST		All	
		k	F1	k	F1	k	F1	k	F1
Citance	None	—	—	1.6	0.044	1.6	0.037	1.2	0.049
	Not in LST	4.0	0.045	1.4	0.041	2.6	0.045	1.4	0.042
	In LST	4.0	0.047	1.7	0.043	1.4	0.045	1.1	0.043
	All	4.1	0.045	1.4	0.042	2.4	0.046	1.4	0.043

Table 4: F1 for different cases, the first column is the constant that yield the best result and the second is the F1 score.

Facet	**4-0**	**3-1**	**2-1-1**
Results	17	53	53
Discussion	12	55	33
Method	15	12	8
Implication	1	3	3
Hypothesis	0	0	0
Total	45	123	97

Table 6: Agreement of 4 annotators over 313 citances

of four. Using only words from the LST gave good result on the biology papers, it remains to check if the same performance can be achieved on other domains.

Over the reference sentences, the same technique obtained better results, probably because of the increased number of sentences to train with. Applying this over the research paper, divided the sentences in five subsets, leaving a smaller group of sentences to look into for the reference sentences. Even with those smaller set, the task of finding the reference sentences proved to be difficult. The distinction between word from the LST didn't help either to a better score. The only positive element was the use of the word 'we' that raised the score by 10% to 30%.

7 Conclusion

We discussed how the preprocessing of the data was done to facilitate the analysis of the papers. We presented the use of distinguishing between topic and non-topic (LST) words for determining the facet of sentences in a paper. We obtained good results with a simple histogram. We described the technique for extracting the references sentences. The distinction between topic and non-topic words did not improve the results for this extraction.

References

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

Patrick Drouin. 2010. Extracting a bilingual transdisciplinary scientific lexicon. *Proceedings of eLexicography in the 21st century : New challenges, new applications*, 7:43–54. Presses universitaires de Louvain, Louvain–a-Neuve.

Patrick Drouin. 2010. From a bilingual transdisciplinary scientific lexicon to bilingual transdisciplinary scientific colloations. *Proceedings of the 14th EURALEX International Congress*, 296–305. Fryske Akademy, Leeuwarden/Ljouwert, Pays-Bas.

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

Aaron Elkiss, Siwei Shen, Anthony Fader, Günes Erkan, David J. States, and Dragomir R. Radev. 2008. Blind men and elephants: What do citation summaries tell us about a research article?. *Journal of The American Society for Information Science and Technology - JASIS*, 59(1):51–62

C. Lee Giles and Kurt D. Bollacker and Steve Lawrence. 1998. CiteSeer: an automatic citation indexing system. *CiteSeer: an automatic citation indexing system*, 89–98.

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

Julian Kupiec, Jan O. Pedersen, and Francine Chen. 1995. A trainable document summarizer *Research and Development in Information Retrieval - SIGIR*, 68–73.

Hans P. Luhn. 1958. The Automatic Creation of Literature Abstracts. *Ibm Journal of Research and Development - IBMRD*, 2(2):159–165.

Qiaozhu Mei, and ChengXiang Zhai. 2008. Generating Impact-Based Summaries for Scientific Literature. *Meeting of the Association for Computational Linguistics - ACL*, 816–824.

Saif Mohammad, Bonnie J. Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishnan, Vahed Qazvinian, Dragomir R. Radev, and David M. Zajic. 2009. Using Citations to Generate surveys of Scientific Paradigms. *North American Chapter of the Association for Computational Linguistics - NAACL*, 584–592.

Brett Powley, and Robert Dale. 2007. Evidence-based information extraction for high accuracy citation and author name identification. *RIAO '07 Large Scale Semantic Access to Content*, 618–632.

Vahed Qazvinian, Dragomir R. Radev, Saif Mohammad, Bonnie J. Dorr, David M. Zajic, M. Whidby, and T. Moon. 2013. Generating Extractive Summaries of Scientific Paradigms. *Journal of Artificial Intelligence Research*, 46:165–201.

Advaith Siddharthan, and Simone Teufel. 2007. Whose Idea Was This, and Why Does it Matter? Attributing Scientific Work to Citations. *North American Chapter of the Association for Computational Linguistics - NAACL*, 316–323.

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

Peter N. Yianilos, and Kirk G. Kanzelberger. 1997. The LikeIt Intelligent String Comparison Facility. *NEC Research Institute*.