Résumé automatique multi-document dynamique

Maali Mnasri

NNT : 2018SACLS342

Résumé automatique
Multi-document et dynamique

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'université Paris-Sud

École doctorale n°580 Sciences et Technologies de l'Information
et de la Communication (STIC)
Spécialité de doctorat: Informatique

Thèse présentée et soutenue à Gif-sur-Yvette, le 20/09/2018, par

Maâli Mnasri

Composition du Jury :

Sophie Rosset Directrice de Recherche, LIMSI CNRS	Président
Jean-Luc Minel Professeur Emérite, Université Paris Nanterre	Rapporteur
Juan-Manuel Torres-Moreno Maître de Conférences HDR, Université d'Avignon	Rapporteur
Antoine Doucet Professeur des Universités, Université de La Rochelle	Examinateur
Gaël de Chalendar Ingénieur chercheur, CEA LIST	Directeur de thèse
Olivier Ferret Ingénieur chercheur HDR, CEA LIST	Encadrant scientifique

Thèse de doctorat

Remerciements

Je tiens à remercier tout d'abord mon directeur de thèse Gaël de Chalendar et mon encadrant scientifique Olivier Ferret pour leur disponibilité, leurs conseils et leur implication dans cette thèse.

Mes remerciements vont également à Jean-Luc Minel et Juan-Manuel Torres-Moreno pour avoir accepté de rapporter sur cette thèse et à Antoine Doucet et Sophie Rosset pour leur participation au jury.

Je remercie vivement aussi Aurélien Bossard et Aurélien Max pour leurs conseils précieux lors de la soutenance à mi-parcours.

Enfin, je tiens à exprimer ma gratitude à ma famille notamment mes parents et mon mari pour leur encouragement et leur soutien.

Table des matières

1 Introduction **1**

1.1 Contexte 1

1.2 Les enjeux du résumé automatique 2

1.3 Problématique 3

1.4 Plan de la thèse 4

2 État de l'art **7**

2.1 Introduction 7

2.2 Les types des résumés automatiques 8

2.2.1 Résumé générique et résumé orienté 8

2.2.2 Résumé indicatif et résumé informatif 9

2.2.3 Portée du résumé 9

2.2.4 Résumé abstractif et résumé extractif 9

2.3 Les méthodes du résumé par abstraction 11

2.4 Les méthodes du résumé par extraction 11

2.4.1 Les critères de sélection des phrases du résumé 12

2.4.2 Exploitation et intégration des critères 16

2.5 Le résumé multi-document et le résumé de mise à jour 21

2.5.1 Résumé multi-document 21

2.5.2 Résumé dynamique : une dimension temporelle 23

2.6 L'évaluation du résumé automatique 25

2.6.1 ROUGE 26

2.6.2 PYRAMID 26

2.6.3 Autres méthodes d'évaluation automatique 27

2.7 Synthèse : tableau comparatif des travaux récents en RA 29

2.7.1 Résumé multi-document 31

2.7.2 Résumé dynamique 37

2.8 Conclusion 44

3 Intégration de la similarité sémantique pour le RA 45
3.1 Introduction . . . 45
3.2 Représentation et similarité sémantique de phrases . . . 47
3.2.1 Que sont les *word embeddings* ? . . . 49
3.2.2 Le framework Word2Vec . . . 51
3.2.3 L'algorithme GloVe . . . 51
3.2.4 Modification des word embeddings : Retrofitting . . . 53
3.2.5 Calcul de la similarité de phrases à partir des *word embeddings* 55
3.3 Clustering sémantique . . . 56
3.4 Sélection de phrases pour le résumé mis-à-jour . . . 59
3.4.1 Formalisation du problème . . . 59
3.4.2 ICSISumm pour le résumé mis-à-jour . . . 61
3.4.3 Prise en compte du clustering sémantique . . . 61
3.5 Conclusion . . . 62

4 Évaluation de l'intégration de la similarité sémantique 65
4.1 Introduction . . . 65
4.2 Cadre d'évaluation . . . 65
4.2.1 Méthode d'évaluation . . . 65
4.2.2 Données d'évaluation . . . 66
4.2.3 Étalonnage des paramètres . . . 67
4.3 Limite supérieure des systèmes extractifs . . . 68
4.3.1 Génération des résumés Oracle . . . 68
4.3.2 Évaluation des résumés Oracle . . . 69
4.4 Systèmes évalués . . . 71
4.4.1 Baselines . . . 71
4.4.2 Systèmes de l'état de l'art . . . 71
4.4.3 Systèmes proposés . . . 72
4.5 Résultats et analyse . . . 72
4.5.1 Influence des paramètres . . . 73
4.5.2 Évaluation intrinsèque de la similarité sémantique . . . 80
4.6 Conclusion . . . 83

5 Exploitation de la Structure Rhétorique pour le RA 85
5.1 Introduction . . . 85
5.2 La Théorie de la Structure Rhétorique (RST) . . . 86
5.3 Travaux précédents sur la RST pour le résumé automatique . . . 89
5.3.1 Méthodes par classement des EDUs . . . 89
5.3.2 Méthodes par élagage de l'arbre RST . . . 92
5.4 Application de la RST pour le résumé mis-à-jour . . . 94
5.4.1 Analyseurs RST . . . 95
5.4.2 Intégration de la RST dans l'ILP . . . 97
5.4.3 Méthode de pondération des EDUs . . . 98
5.4.4 Méthode de pondération des bigrammes . . . 99
5.4.5 Évaluation du système avec les nouveaux poids . . . 100
5.5 Fusion de systèmes de résumé . . . 101
5.5.1 Travaux précédents . . . 102
5.5.2 Limite supérieure de la fusion de systèmes . . . 103
5.5.3 Méthodes de fusion utilisées . . . 104
5.5.4 Mise en oeuvre de la fusion de systèmes et évaluation des résultats . . . 106
5.6 Conclusion . . . 108

6 Bilan et perspectives 109
6.1 Bilan . . . 109
6.2 Perspectives . . . 111

A Annexe : Conception et implémentation 115
A.1 Logiciels utilisés . . . 115
A.2 Ressources et données . . . 116
A.3 Temps d'exécution . . . 116
A.4 Diagrammes de flux de données . . . 116

Bibliographie 121

Table des figures

3.1 Différence entre les architectures CBOW et Skip-Gram de Word2Vec 50
3.2 Exemple de matrice de cooccurrence utlisée par GloVe 52
3.3 Exemple de matrice de cooccurrence obtenue par GloVe après l'exécution 53
3.4 Architecture de ConceptNet Vector Ensemble (Speer and Chin, 2016) 55
3.5 Principe du clustering des phrases pour la détection de nouveauté . 57
3.6 Exemple d'application des marches aléatoire dans un graphe 58

4.1 Influence du seuil de similarité sur TAC 2008 en utilisant les vecteurs Word2Vec 75
4.2 Influence du seuil de similarité sur TAC 2008 en utilisant les vecteurs GloVe 75
4.3 Influence du seuil de similarité sur TAC 2008 en utilisant les vecteurs ConceptNet 76
4.4 Influence du seuil de similarité sur TAC 2009 en utilisant les vecteurs Word2Vec 77
4.5 Influence du seuil de similarité sur TAC 2009 en utilisant les vecteurs GloVe 78
4.6 Influence du seuil de similarité sur TAC 2009 en utilisant les vecteurs ConceptNet 78
4.7 Influence du facteur de pénalisation sur TAC 2008 en utilisant les vecteurs Word2Vec 79
4.8 Influence du facteur de pénalisation sur TAC 2009 en utilisant les vecteurs Word2Vec 79

5.1 Arbre RST d'un court éditorial de journal politique 88
5.2 Arbre RST d'un extrait d'une lettre personnelle 88
5.3 Arbre RST d'un extrait d'une lettre personelle pondéré par la méthode de Ono 90
5.4 Arbre RST pondéré par la méthode de Marcu 1998 91
5.5 Exemple d'un arbre RST (Hirao et al., 2013) 92

5.6 L'arbre de dépendances obtenu à partir l'arbre RST de la figure 5.5 (Hirao et al., 2013) 93
5.7 Exemple de l'arbre RST imbriquant les arbres de dépendances (Kikuchi et al., 2014) 94
5.8 Exemple d'un article de journal 98
5.9 Arbre RST généré par DPLP 98

A.1 Diagramme de flux de données de l'intégration de la similarité sémantique dans le modèle ILP 117
A.2 Diagramme de flux de données de la prise en compte de l'analyse du discours RST dans le modèle ILP 119

Chapitre 1

Introduction

1.1 Contexte

Ce travail s'intéresse à la tâche du Résumé Automatique (RA) de texte, qui représente un des défis importants du Traitement Automatique des Langues (TAL). Le nombre de recherches abordant cette thématique ne cesse d'augmenter pour répondre au besoin croissant d'outils de filtrage et de synthèse d'information destinés à faire face la grande richesse informationnelle qui nous entoure. En effet, de nos jours, Internet est accessible à presque tout le monde et tout le monde participe à la production d'informations disponibles largement. Être connecté et utiliser des services en ligne contribue aussi de façon directe ou indirecte à alimenter les centres de données des entreprises offrant des services en ligne, à savoir, les réseaux sociaux comme Google, Facebook et Twitter et les sites de e-commerce comme Amazon. Les données disponibles peuvent être de natures différentes : données numériques, images ou textuelles. L'abondance de données textuelles impose d'avoir des outils permettant d'analyser et de comprendre rapidement leur contenu sans avoir à les lire toutes. De plus, il est déniable que le monde devient de plus en plus numérique et que les gens sont "hyper-connectés". Une étude d'un bureau de marketing a révélé qu'une personne consulte son smartphone en moyenne 221 fois par jour [1]. C'est la façon dont nous collectons les informations, découvrons les nouvelles et suivons des événements. Au vu de la diversité des sources, l'utilisateur peut se trouver face à la même information plusieurs fois ce qui représente une perte de temps importante et ralentit l'accès aux nouvelles recherchées. En fait, ce cas représente une des applications d'une variante particulière du RA : le RA mis-à-jour (dit aussi dynamique ou temporel). Il s'agit de présenter un ou plusieurs

1. Étude réalisée par Tecmark au Royaume-Uni sur 2000 individus

textes sous une forme condensée contenant les informations les plus importantes tout en tenant compte de l'aspect temporel de l'information et de l'historique de l'utilisateur. Le résumé mis-à-jour vise à faciliter la tâche de filtrage de l'information par l'utilisateur en lui présentant seulement du contenu nouveau, c'est-à-dire, du contenu dont il n'a pas eu connaissance par une quelconque source auparavant. Cette thèse s'inscrit dans ce cadre particulier du résumé mis-à-jour. Cet exercice est difficile à accomplir par la machine dans la mesure où pour produire un résumé mis-à-jour, il faut idéalement pouvoir comprendre le texte source en passant par une analyse sémantique et repérer la nouveauté qu'il apporte par rapport à un historique. De façon générale, il existe deux approches principales pour aborder la tâche de résumé automatique : une approche par extraction et une approche par abstraction. Pour créer un extrait, un système doit simplement identifier les éléments textuels les plus importants et les renvoyer au lecteur. Bien que ce résumé ne soit pas nécessairement cohérent, il doit permettre au lecteur de se faire une opinion sur le contenu du texte original. Dans l'approche abstractive, le système de RA repère les informations principales et les utilise pour générer un résumé. Même si les approches abstractives focalisent actuellement l'attention au travers des modèles fondés sur les réseaux de neurones, la plupart des systèmes de RA aujourd'hui produisent des extraits seulement. Nous avons aussi choisi de suivre cette lignée de résumé extractif.

1.2 Les enjeux du résumé automatique

La thématique du résumé automatique de texte pose divers problèmes complexes. Ces problèmes sont liés non seulement à la modélisation de la tâche du résumé en elle-même mais aussi à l'avancement actuel des thématiques connexes dont dépend le RA. Globalement, il faut savoir exploiter de façon efficace les outils fournis par les différents sous-domaines du TAL pour imiter l'action de synthèse humaine des textes et produire un résumé concis et précis. Cet objectif global est souvent adapté en fonction de contraintes applicatives liées au domaine abordé, l'utilisation finale, la structure des documents, etc. Le premier obstacle est la numérisation des données textuelles dans un espace où l'on peut définir des distances entre les unités textuelles de façon à refléter leur sens. Construire un modèle re-

présentant toute l'information sur le texte (relations de niveau lexical, syntaxique, sémantique et discursif entre tous les éléments du texte) de façon exhaustive est très complexe et peut être inutile. Il faut pouvoir sélectionner les techniques qui améliorent la qualité des résumés. L'évaluation automatique ou semi-automatique des systèmes de RA est aussi un obstacle qui freine l'avancement des travaux. La plupart des outils utilisés aujourd'hui nécessitent des résumés de référence qui ne sont pas assez abondants. De plus, malgré les différences entre les meilleurs systèmes de RA, les outils d'évaluation actuels ont tendance à leur attribuer presque les mêmes scores, ce qui met en question les méthodes d'évaluation (Hong et al., 2014).

1.3 Problématique

Le problème du résumé automatique peut être défini généralement par la sélection des informations les plus marquantes d'un ou de plusieurs documents source tout en minimisant la redondance au sein de cette sélection et en respectant une taille maximale du résumé final. Notre stratégie pour résoudre ce problème est guidée par deux lignes directrices majeures : l'une s'exprime en termes d'objectifs tandis que l'autre concerne les moyens.

La première ligne directrice s'articule autour de deux dimensions du résumé : la non-redondance et la saillance des informations avec un intérêt spécifique pour la non-redondance étant donné que nous traitons le résumé mis-à-jour. Le premier objectif consiste à modéliser explicitement la redondance entre les éléments du texte à résumer. Nous considérons que la redondance informationnelle est un axe majeur dans notre contexte dans la mesure où elle représente un critère important pour orienter le choix des phrases dans le résumé multi-document mais aussi un critère central de sélection de phrases dans le résumé mis-à-jour. Ce dernier peut être considéré en effet comme un compromis entre le résumé multi-document et la détection de nouveautés. Notre second objectif consiste à prendre en compte des critères d'importance et de saillance afin de garantir que les informations sélectionnées soient non seulement nouvelles mais aussi pertinentes. Une caractéristique du travail réalisé est que la prise en compte de la non-redondance et de la saillance s'effectue dans un même cadre, une approche de type ILP (Integer Linear Pro-

gramming), et que contrairement à Gillick and Favre (2009) par exemple, cette prise en compte s'appuie sur une modélisation explicite des phénomènes considérés s'appuyant sur des traitements linguistiques assez élaborés.

Notre seconde ligne directrice concerne plus spécifiquement les outils et les techniques à exploiter pour modéliser et satisfaire les contraintes du problème. De ce point de vue, les travaux actuels sur le résumé automatique exploitent majoritairement des méthodes issues de la Recherche d'Information. Nous nous proposons d'étudier pour notre part comment l'information provenant de traitements linguistiques plus élaborés peut contribuer favorablement au problème de résumé que nous considérons. En premier lieu, nous nous sommes intéressés à des traitements opérant au niveau du contenu du texte et particulièrement à des mesures de similarité sémantique au niveau phrastique. En second lieu, nous visons des traitements à la fois plus discursifs et plus fonctionnels permettant de déterminer le rôle rhétorique d'un segment de phrase ou d'une phrase dans un texte. Une telle analyse s'inscrit plus globalement dans la mise en évidence de la structure discursive des textes, structure n'ayant jusqu'à présent été exploitée que de façon assez limitée dans le cadre du résumé automatique (Marcu, 1998a).

1.4 Plan de la thèse

Ce travail est organisé de la façon suivante. Dans le chapitre 2, nous décrivons l'état de l'art du résumé automatique multi-document. Nous nous focalisons surtout sur les méthodes extractives en abordant aussi plus globalement les méthodes d'évaluation du résumé automatique. Ce chapitre se termine par un tableau de synthèse rassemblant plusieurs systèmes de résumé multi-document et de résumés mis-à-jour organisés par type d'approches afin de repérer facilement les approches qui fonctionnent mieux que d'autres sur le même jeu de données. Ensuite, nous abordons dans le chapitre 3 la problématique de minimisation de la redondance via l'intégration de la similarité inter-phrase dans un système de résumé mis-à-jour, décrit en détail dans le même chapitre. L'évaluation de cette approche fait l'objet du chapitre 4 où nous vérifions l'efficacité de notre approche et étudions l'effet des différents critères et paramètres sur son fonctionnement afin d'expliquer l'amélio-

ration obtenue. Le chapitre 5 étudie la contribution de l'analyse discursive dans le cadre du résumé mis-à-jour. Différentes approches d'agrégation de systèmes sont appliquées et présentées dans le même chapitre afin de fusionner les résultats des approches sémantique et discursive. Finalement, nous consacrons le chapitre 6 aux conclusions tirées de ce travail ainsi qu'aux orientations de recherches futures dans le domaine.

Chapitre 2

État de l'art

2.1 Introduction

Notre thèse prend en compte comme objet le résumé multi-document dynamique, appelé aussi résumé mis-à-jour (*update summarization*) ou résumé temporel. Ce résumé est une variante du résumé multi-document générique prenant en compte l'aspect chronologique des documents. Cette variante du résumé automatique a gagné beaucoup en intérêt dernièrement avec l'avènement d'Internet et l'explosion de la masse des données auxquelles les utilisateurs sont exposés quotidiennement. Avec cette surcharge d'informations reçues en temps réel, il est difficile à l'utilisateur de filtrer les nouvelles informations au fur et à mesure de leur production. Le but du résumé mis-à-jour est de faciliter l'appréhension de ces données. Le résumé mis-à-jour reçoit en entrée un flux de documents traitant traitant un sujet précis et organisé en différentes collections successives. Chaque collection apporte des informations nouvelles par rapport aux collections de documents qui la précèdent. En supposant qu'à chaque instant, l'utilisateur a pris connaissance des documents reçus aux instants précédents, le résumé mis-à-jour vise à produire un résumé des documents reçus instantanément en se focalisant sur les nouvelles apportées par ce document tout en essayant de ne pas répéter les informations déjà présentes dans les anciens documents et par conséquent déjà lues.

Du point de vue pratique, l'approche dominante pour traiter la problématique du résumé mis-à-jour est d'utiliser un système de résumé multi-document générique, qui accomplit une partie de la tâche du résumé dynamique. Pour les adapter au résumé mis-à-jour, ces systèmes sont enrichis par des critères liés à la détection de nouveauté pour orienter la sélection d'information vers les informations exclusive-

ment présentes dans les nouveaux documents. Une grande partie de ces méthodes génère un résumé multi-document classique en premier lieu. Ensuite les phrases redondantes avec les anciens documents sont éliminées. La conclusion à tirer de cette observation est que le résumé multi-document et le résumé mis-à-jour sont étroitement liés et une amélioration dans la tâche du résumé multi-document se répercuterait sans doute sur la tâche du résumé mis-à-jour. En partant de cette conclusion, nous allons nous intéresser dans ce chapitre à l'état de l'art du résumé multi-document et du résumé mis-à-jour. Nous commençons par inventorier les différentes catégories de résumé automatique de façon générale selon plusieurs critères. Ensuite, bien que notre thèse s'inscrive dans le cadre du résumé extractif, nous décrivons brièvement les méthodes du résumé par abstraction pour couvrir autant que possible les méthodes existantes. La section suivante fait le tour des différentes méthodes de résumé par extraction. Nous détaillons, ensuite, les tâches du résumé multi-document et du résumé dynamique en insistant sur les contraintes imposées par chaque type de résumé. Finalement, nous présentons un tableau regroupant quelques systèmes de résumé multi-document et dynamique de l'état de l'art, afin de comparer leurs performances.

2.2 Les types des résumés automatiques

Les résumés automatiques et leurs méthodes peuvent être catégorisés selon différents critères Nenkova and McKeown (2012). Nous citons les plus importants et les plus utilisés dans la littérature.

2.2.1 Résumé générique et résumé orienté

Un résumé de texte est soit générique, soit orienté. Le résumé générique est produit en se référant uniquement au contenu du texte source, indépendamment de son contexte. En revanche, le résumé orienté est guidé par une tâche ou une requête. Dans ce cas, seule l'information en relation avec la tâche ou la requête est sélectionnée. Ce type de résumé dépend donc fortement du contexte. Ce dernier peut être défini comme un ensemble de facteurs d'entrée du système de résumé automatique (Spärck Jones, 2007). Il couvre l'audience, l'usage, le cadre spatio-

temporel, etc.

2.2.2 Résumé indicatif et résumé informatif

Un résumé est soit informatif, soit indicatif. Le résumé informatif est un modèle rétréci du texte d'origine relatant le plus largement possible les informations du document. En revanche, un résumé indicatif liste les sujets les plus importants évoqués par le texte. Certains systèmes de résumés guidés (Saggion and Lapalme, 2002) génèrent un résumé indicatif du texte comme étape initiale. L'utilisateur choisit parmi les sujets proposés par le résumé ceux qui l'intéressent. Le système produit alors un résumé informatif du texte guidé par la requête de l'utilisateur. La requête dans ce cas est l'ensemble des sujets sélectionnés à partir du résumé indicatif.

2.2.3 Portée du résumé

Les systèmes de résumé automatique peuvent être mono-document ou multi-document. Les premiers produisent des résumés pour un seul document et peuvent être plus ou moins adaptés à des tailles différentes de documents : résumer un article ne pose pas tout à fait le même problème que résumer un rapport scientifique. Le système CHORAL (García Flores et al., 2009) fondé sur l'analyseur linguistique LIMA (de Chalendar, 2014) se distingue ainsi par son efficacité sur les documents longs. Il produit des résumés de 1 à 5 pages pour un rapport de thèse. Les systèmes de résumé multi-document, plus récents, génèrent des résumés de taille ajustable d'un ensemble de documents.

2.2.4 Résumé abstractif et résumé extractif

Nous distinguons les méthodes extractives (Dalal and Malik, 2013) des méthodes abstractives (Genest and Lapalme, 2012). Le résumé extractif est formé de segments de texte extraits du texte source. Ces segments peuvent être des phrases, des propositions ou n'importe quelle unité textuelle. Les premiers travaux en résumé automatique se sont appuyés sur cette approche (Luhn, 1958) en exploitant la fréquence des mots. Les critères de sélection ont ensuite été enrichis

en tenant compte du contenu et de la structure du texte (Edmundson, 1969) (cf. section 4.1). Ces méthodes ont été, initialement, les plus exploitées parce qu'elles évitent le problème de la génération de texte, toujours considéré comme une tâche complexe. Bien que le résumé extractif peut manquer de cohérence, il est grammaticalement correct d'où sa lisibilité par rapport aux approches par génération. Les méthodes abstractives ont été inspirées, à l'origine, des travaux en psycholinguistique cognitive et en intelligence artificielle, notamment du modèle théorique de la compréhension de van Dijk et Kintsch (Kintsch and van Dijk, 1978). Ce dernier considère le résumé d'un texte comme le produit de sa compréhension. Celle-ci est modélisée par la mise en relation sémantique des composants du texte dans une structure adaptée (par exemple un graphe de cohérence). Un résumé abstractif est le produit de la synthèse de la représentation sémantique du texte source avec des phrases générées automatiquement. Ces méthodes n'ont pas été très largement exploitées. Ceci peut être dû à la rareté des outils de génération de texte et à leur performance modeste. La majorité des travaux s'étant intéressés aux méthodes extractives, ces dernières ont connu un développement important, favorisé par des prérequis peu exigeants. Cependant, les méthodes neuronales récentes de type *sequence-to-sequence* (Sutskever et al., 2014) ont montré leur intérêt dans le domaine du TAL et particulièrement pour la traduction automatique (Cho et al., 2014). Ces approches ont aussi marqué un changement important dans le RA. En effet, elles ont prouvé qu'il est possible, dans une certaine mesure, de générer des résumés sans passer par une étape de compréhension profonde. Simultanément, des interrogations sur les performances maximales des techniques extractives ont été soulevés. Des travaux récents se sont intéressés à vérifier s'il existe encore une marge d'amélioration dans le paradigme du résumé extractif pour s'assurer de l'utilité des travaux en cours sur cet aspect (Schluter, 2017, Hirao et al., 2017). L'approche consiste à générer des résumés oracle en extrayant les phrases maximisant le score d'évaluation. Des méthodes dites *greedy*[1] ont été utilisées ainsi que des méthodes d'optimisation globale. Dans les deux cas, il a été prouvé que les systèmes extractifs actuels sont relativement loin de la limite supérieure atteignable. Par conséquent, contrairement à ce qui peut être pensé, la problématique

1. Les méthodes greedy ou gloutonne sont des méthodes incrémentales qui consistent à faire à chaque incrément, le choix optimum local.

du résumé par extraction n'est pas encore résolue.

2.3 Les méthodes du résumé par abstraction

Bien que nous nous intéressions surtout aux systèmes de résumé extractifs, les systèmes abstractifs partagent avec le résumé dynamique une certaine forme de modélisation du contenu des documents, même si les critères d'extraction dans le cas dynamique sont généralement sémantiquement moins profonds. Les méthodes de résumé abstractives imitent, jusqu'à un certain degré, le processus naturel accompli par l'homme pour résumer un document. Par conséquent, elles produisent des résumés plus similaires aux résumés manuels. Ce processus peut être décrit par deux étapes majeures : la compréhension du texte source et la génération du résumé (Khan and Salim, 2014). Ces deux tâches sont assez complexes. C'est pourquoi elles ont été simplifiées. La première étape vise à analyser sémantiquement le contenu du texte et à identifier les parties à exprimer dans le résumé. Elle a parfois pris la forme d'une tâche d'extraction d'information liée au domaine abordé (Genest and Lapalme, 2011, 2012) ou de regroupement des phrases du texte source (Filippova, 2010). La génération de texte est un domaine en soi. Une des approches simplifiées consiste à appliquer des techniques de génération text-to-text : utilisation de paraphrases (Madnani and Dorr, 2010) ou fusion et compression de phrases (Filippova, 2010). Une alternative consiste à induire un modèle textuel du domaine (patron) et de l'instancier lors de la génération (Cheung et al., 2013).

2.4 Les méthodes du résumé par extraction

Le point fort du résumé par extraction est qu'il évite la génération de texte. Ceci permet d'une part, de se concentrer sur la sélection du contenu pertinent et d'autre part, d'obtenir un résumé lisible et linguistiquement correct. La cohérence n'est en revanche pas garantie. Par exemple, si le système de résumé sélectionne des phrases contenant des références (acronyme, pronom personnel, etc.) et ne sélectionne pas les phrases contenant leurs antécédents, il est fort probable que le résumé produit soit incompréhensible. Pour pallier ce problème, certains travaux considèrent le paragraphe comme unité d'extraction au lieu de la phrase (Salton

et al., 1996). Ceci permet de garder la cohérence du texte source mais ne peut pas être applicable dans le cas de résumés courts. De plus, il est évident que cette méthode réduit la précision du résumé en y incluant des phrases peu importantes juste pour améliorer la cohérence. D'autres chercheurs procèdent à des étapes de pré/post-traitement du texte qui améliorent partiellement la cohérence globale du résumé, comme par exemple la résolution des références anaphoriques dans le texte source (Trandabâţ, 2011). Le processus principal dans le résumé extractif est la sélection des segments de textes (généralement les phrases) pertinents et non redondants sans dépasser une taille limite du résumé. Ce principe limite la couverture des informations apportées par le texte source. Les résumés abstractifs souffrent moins de ce problème puisque l'information peut y être reformulée

2.4.1 Les critères de sélection des phrases du résumé

Dans cette partie nous détaillons les critères de sélection des unités textuelles utilisés par les systèmes de résumé. Ces unités peuvent être des phrases, des N-grammes ou n'importe quel segment du texte. Ces critères ne sont pas spécifiques d'une méthode bien déterminée mais sont applicables à tous les types de résumés extractifs qu'ils soient mono-document, multi-document ou dynamiques.

Critères liés au contenu du texte

Cet ensemble de critères s'intéresse au contenu du texte et aux informations qu'il apporte. Le contenu est analysé soit par des approches de surface, comme le calcul des fréquences d'occurrence des mots, soit par des approches sémantiques qui exploitent le sens des mots et leurs relations sémantiques, comme avec l'annotation en rôles sémantiques. Nous citons, dans ce qui suit, les critères les plus utilisés.

Fréquence d'occurrence des mots. Ce critère a été introduit initialement par Luhn (Luhn, 1958). L'idée est que les mots les plus fréquents sont les plus liés au sujet du texte. La fréquence d'occurrence des mots est largement exploitée, même dans des systèmes récents où elle est combinée à d'autres critères. Même les méthodes reposant sur l'analyse sémantique des mots utilisent la fréquence d'occurrence comme première étape pour déterminer les thèmes principaux abordés

par le texte. Le point fort de ce critère est qu'il est totalement indépendant de la langue.

Similarité entre les phrases. La similarité textuelle est une notion très importante en TAL comme en témoignent les évaluations SemEval par exemple. De nombreuses mesures de similarité textuelle ont ainsi été établies (Bär et al., 2015). Dans le domaine du résumé automatique, cette similarité est d'abord exploitée pour l'élimination de la redondance mais aussi plus indirectement pour la sélection de phrases pertinentes, sans oublier la comparaison avec des résumés modèles lors de l'évaluation. Certaines méthodes de résumé s'appuient uniquement sur ce critère. Tel est le cas de l'algorithme de résumé mono-document *TextRank* (Mihalcea, 2004). Ce critère est par ailleurs particulièrement important dans le cas multi-document. Dans ce contexte, les documents sont généralement représentés par des vecteurs de mots pondérés avec une mesure comme TF*IDF (Term Frequency * Inverse Document Frequency) (Sammut and Webb, 2010) et regroupés selon la similarité de leurs vecteurs. Plus une phrase est similaire au barycentre du regroupement, plus elle décrit les informations caractéristiques du groupe de documents considéré (Radev et al., 2004, Neto et al., 2003) et peut être alors considérée comme représentative de ce groupe, ce qui est un critère de sélection important.

Reconnaissance d'entités nommées / Annotation en rôles sémantiques. La reconnaissance des entités nommées dans un texte améliore le filtrage des informations pertinentes (Hassel, 2003). Elle permet aussi de répondre à des requêtes factuelles (OÙ, QUI, QUAND, etc.) dans le résumé guidé (Tan, 2011). Certains vont au-delà de cette étape et déterminent les rôles sémantiques des entités reconnues (Trandabăţ, 2011). L'entité la plus fréquente est identifiée et considérée comme entité principale. Par la suite, les phrases contenant cette entité sont sélectionnées. Enfin, seules les phrases où l'entité principale possède un rôle sémantique fondamental (non auxiliaire) sont gardées pour le résumé. Les rôles sémantiques peuvent aussi être utilisés pour simplifier les phrases complexes, c'est-à-dire les phrases contenant deux prédicats ou plus. Le prédicat est généralement un verbe. Dans ce cas, les prédicats pour lesquels l'entité principale a un rôle auxiliaire sont éliminés.

Ces critères mettent l'accent sur le contenu du texte et le message qu'il com-

munique. Il existe d'autres critères qui ne s'intéressent pas au contenu du texte, mais qui renferment des informations très importantes et décisives dans l'étape de sélection. Elles font l'objet du paragraphe suivant.

Critères liés à la forme et à la structure du texte

La structure du texte est très importante dans le jugement de la pertinence d'une phrase. En effet, lors de la rédaction d'un texte, l'ordre des phrases n'est pas arbitraire. De plus, les styles de rédaction diffèrent d'un domaine à l'autre. Par exemple, dans le domaine journalistique, les informations les plus importantes sont souvent mentionnées au début du texte. Ceci n'est pas toujours le cas dans un article scientifique ou un roman. Ce facteur a été exploité par les chercheurs en TAL pour déterminer l'importance des segments textuels. Nous expliquons dans cette partie les critères les plus importants.

Position de la phrase. Ce critère dépend de la nature du document et de son genre. Les phrases se trouvant au début sont généralement plus informatives et décrivent le sujet principal du document. De plus, les phrases situées au début de chaque paragraphe tendent à apporter plus d'informations pertinentes (Lin and Hovy, 1997, McKeown et al., 1999). Dans le résumé des articles scientifiques, certains travaux se sont appuyés principalement sur la structure des articles (Jaidka et al., 2013) pour générer des revues scientifiques. Les revues descriptives (résumé informatif) sont formées par les phrases des parties *Résumé* et *Introduction*. En revanche, dans le cas des revues intégratives (critique et comparaison des études), les phrases les mieux notées sont celles des parties *Résultats et discussion* et *Conclusion*. Cette approche est déduite de l'analyse d'un corpus de 20 revues scientifiques et de 349 références pointées par ces revues. Il a été constaté que plus que 25% des informations contenues dans les revues ont été extraites de la partie *Résumé* des articles source.

Similarité avec le titre. Plus une phrase est similaire avec le titre, plus elle est liée au sujet principal du texte (Edmundson, 1969) étant donné que dans la majorité des cas le titre informe de façon très brève sur le contenu principal du texte. La similarité avec les sous-titres est aussi considérée comme indicateur de pertinence.

Longueur de la phrase. La longueur moyenne d'une phrase dans un texte dépend de son genre. Généralement, les phrases très courtes sont considérées comme peu informatives alors que les phrases très longues sont présumées détailler des informations déjà exprimées dans l'ensemble des documents par des phrases plus courtes et donc favoriser la redondance. Cette caractéristique est exploitée en fixant un intervalle de longueur (entre 15 et 30 mots). Une phrase ayant une longueur en dehors de cet intervalle est pénalisée (Schiffman et al., 2002).

Les mots indices (*cue word*). Ce critère prend la forme d'une liste de mots activant ou inhibant la sélection d'une phrase, généralement en fonction du rôle qu'ils permettent d'attribuer à la phrase dans laquelle ils apparaissent (exemple, conclusion, etc.) (Edmundson, 1969). Ces listes sont constituées manuellement ou définies par apprentissage à partir d'un corpus de documents représentatifs (Mani, 2001). Elles peuvent inclure des noms propres (Neto et al., 2003) et des dates.

Analyse du discours. L'analyse du discours est l'ensemble des théories et des modèles qui expliquent comment les énoncés individuels se situent les uns par rapport aux autres dans un discours cohérent et rationnel. Généralement, les théories, les modèles et les implémentations du traitement du langage naturel plaident en faveur d'une mesure de cohérence basée sur trois thèmes : le sens, la structure et l'intention (Mc Kevitt et al., 1992). L'analyse du discours permet ainsi de contextualiser les énoncés et de leur donner un rôle par rapport à l'ensemble du texte, rôle qui peut être exploité pour leur sélection dans le cadre du résumé (Ferret et al., 2001). Parmi les méthodes d'analyse du discours qui ont été largement appliquées pour le résumé, on peut citer la *Rhetorical Structure Theory* (RST) (Mann and Thompson, 1988). La RST s'appuie sur une segmentation des textes en unités discursives élémentaires classées selon leur importance en noyaux (information essentielle) et satellites (information marginale). Elle représente la cohérence et la structure du texte par un ensemble de relations rhétoriques entre les noyaux et les satellites : exemplification, preuve, justification, etc. La plupart des analyseurs RST produisent soit un arbre binaire dont les feuilles représentent les EDUs et les arcs représentent les rôles rhétoriques (Joty et al., 2013, Feng and Hirst, 2014) soit un graphe dont les nœuds sont les EDUs et les arcs sont les rôles (Wolf and Gibson, 2005, Louis et al., 2010b). La structure produite par l'analyse du discours a été exploitée, par exemple, afin de déduire un ordre d'importance des EDUs

(Marcu, 1997, 1998a). Nous décrivons plus en détail les méthodes fondées sur la RST, en particulier, dans le chapitre 5

Nous avons cité les critères de sélection les plus utilisés pour le résumé automatique. Le choix de bons critères n'est pas suffisant pour obtenir un bon résumé. Il faut savoir comment les utiliser et quel degré d'importance accorder à chacun pour produire un résumé satisfaisant. La section suivante s'intéresse à cette problématique.

2.4.2 Exploitation et intégration des critères

Il est très rare qu'un système de résumé automatique utilise un seul critère pour sélectionner les phrases du texte source. Plusieurs critères sont combinés. Les méthodes d'intégration sont assez nombreuses. Nous décrivons dans cette partie les différentes méthodes pour combiner les critères et les utiliser pour sélectionner les phrases du résumé.

Méthodes par apprentissage automatique

Du point de vue de l'apprentissage, le résumé automatique par extraction a été formalisé en différents problèmes qui ont été résolus par des méthodes d'apprentissage automatiques variées.

i. Formalisation du problème. la plupart des travaux sur le résumé automatique ont considéré ce dernier aussi bien comme un problème de classification que comme un problème de régression. Étant donné un ensemble de textes source et leurs résumés, les méthodes par apprentissage visent à apprendre un modèle de choix des phrases du résumé. Les phrases des textes source sont caractérisées par divers critères de sélection.

— **Un problème de classification.** Dans l'approche par classification, le modèle choisi distingue les phrases du texte à inclure dans le résumé et celles à ne pas inclure dans le résumé. Le modèle bayésien naïf donne généralement les meilleurs résultats (Neto et al., 2003).

— **Un problème de régression.** Dans l'approche par régression, le modèle prédit les scores des phrases (Conroy et al., 2011). La décision est alors quantifiée. L'ordonnancement des phrases reste à la charge du système de

résumé.

ii. Méthodes. Différentes méthodes d'apprentissage classiques ont été appliquées au résumé automatique dont nous détaillons quelques unes dans les tableaux 2.7.1 et 2.7.2 de la fin de ce chapitre. Compte tenu des développement actuels en apprentissage automatique, nous mettons l'accent particulièrement sur les réseaux de neurones en tant que méthode d'apprentissage faisant l'objet des développements les plus actifs en ce moment.

Globalement, les réseaux de neurones ont montré leur efficacité dans différents domaines comme le traitement d'images. Dans le résumé automatique, les tout premiers travaux ont utilisé des réseaux de neurones simples pour la sélection des phrases du résumé. (Kaikhah, 2004) a modélisé chaque phrase de texte par un vecteur de n composantes, chacune correspondant à un critère de sélection. Ensuite, il met en place un réseau de neurones composé de n neurones d'entrées (un neurone par critère), une couche cachée de p neurones et un seul neurone de sortie. Ce dernier indique si la phrase en entrée doit être incluse dans le résumé. La phase d'apprentissage permet d'adapter le poids des liaisons entre les couches sur les données d'entraînement. À l'issue de cet apprentissage, les liaisons de très faible poids sont éliminées, de même que les neurones isolés.

Néanmoins, avec le développement des réseaux de neurones, notamment le *deep learning*, des approches plus sophistiquées ont été proposées. Il se trouve que les méthodes neuronales sont plus exploitées pour les résumés abstractifs étant donné leur capacité à représenter l'information de façon condensée. Cependant, le problème de cohérence et de lisibilité représente toujours une limite pour de telles approches.

Récemment, certains travaux se sont intéressés à appliquer ces méthodes neuronales pour le résumé extractif. Cao et al. (2015) ont ainsi exploité des réseaux de neurones convolutifs pour capturer les caractéristiques du résumé à partir de séquences de mots de longueurs différentes. Ces caractéristiques sont ensuite couplées à des critères de sélection standard dans un modèle de régression visant à classer les phrases par ordre d'importance. Les phrases du résumé sont finalement sélectionnées selon une méthode gloutonne similaire à la MMR (Maximum Marginal Relevance) (Carbonell and Goldstein, 1998).

Cheng and Lapata (2016) ont proposé un modèle neuronal pour le résumé mono-

document. Leur modèle repose sur deux composantes principales : un lecteur et un extracteur. Le lecteur a pour but de représenter la sémantique des phrases et du document. L'extracteur sélectionne les phrases séquentiellement en veillant à ce qu'elles soient pertinentes et non mutuellement redondantes. Pour pallier au problème du manque des données d'entraînement, des ensembles de texte-résumé ont été extrait automatiquement de *DailyMail*.

Nallapati et al. (2017) utilisent un réseau de neurones récurent pour classifier les phrases en phrases de résumé ou non. Les phrases sont représentées dans la première couche du réseau par leurs *word embeddings* où chaque neurone représente un mot et une séquence de neurones représentent une phrase. Jain et al. (2017) s'inscrivent dans la même approche mais au lieu d'utiliser les *word embeddings* comme critère, utilisent plusieurs critères comme la longueur et la position de la phrase, l'occurrence des noms propres, etc. De plus, ils représentent chaque phrase par le vecteur moyennant les vecteurs des mots qui la composent. Le vecteur obtenu est utilisé comme critère parmi les autres critères cités.

Ren et al. (2017) proposent un modèle neuronal qui prend en compte les relations contextuelles des phrases avec leurs phrases environnantes pour l'extraction du résumé mono-document. Ce modèle apprend conjointement les représentations vectorielles des phrases et de leurs contextes ainsi que leurs similarité afin de détecter par la suite à quel point une phrase résume son contexte.

Enfin, Zhang et al. (2017) proposent une approche fondée sur des réseaux de neurones convolutifs dit *multiview*, c'est-à-dire à plusieurs perspectives. Utilisés à la base dans l'imagerie 3D, ces réseaux ont été appliqués pour le résumé multi-document. En partant des *word embeddings* des mots des documents à résumer, ces réseaux de neurones *multiview* permettent de produire des représentations vectorielles des phrases selon différents points de vue. Ces représentations sont finalement combinées. La dernière couche du réseau de neurones utilise les représentations des phrases pour leur attribuer un score.

Méthodes fondées sur les graphes

En représentant un texte sous la forme d'un graphe de phrases, il devient possible d'appliquer un certain nombre d'algorithmes génériques, comme l'algorithme PageRank (Page et al., 1999), pour déterminer l'importance relative de celles-ci.

PageRank est un algorithme de classement utilisé par le moteur de recherche de Google. Il représente les pages Web par les sommets d'un graphe et les hyperliens entre ces pages par des arcs entre ces sommets. Il attribue récursivement à chaque nœud un score dépendant à la fois de ses arcs entrants et du score des nœuds source de ces arcs. TextRank est un algorithme pour le résumé automatique mono-document fondé sur les graphes (Mihalcea, 2004). Le texte est représenté par un graphe où les sommets sont tout simplement les phrases du texte. Alors que les arcs des arbres rhétoriques (cf. section 4.1.2) représentent des relations rhétoriques entre les phrases, les arcs dans TextRank représentent leurs similarités. Pour ne pas favoriser les phrases longues au détriment des phrases courtes, la valeur de la similarité entre deux unités textuelles est divisée par la somme de leurs longueurs. Initialement, à chaque sommet est attribué un score aléatoire. Par la suite, à chaque itération de l'algorithme TextRank, le score de chaque nœud est calculé récursivement en fonction de sa similarité avec ses voisins et des scores de ces derniers. La même approche a été appliquée pour le résumé multi-document, notamment en français par Boudin and Torres-Moreno (2009b). Pour éliminer la redondance inter-phrases, un simple seuil de similarité a été utilisé.

Méthodes fondées sur l'ILP

À l'origine des approches fondées sur l'ILP (Interger Linear Programming [2]), McDonald (2007) a proposé d'exprimer le problème du RA sous la forme d'un problème ILP dont la fonction objectif cherche à maximiser le poids des phrases sélectionnées. Ce poids est pénalisé par la redondance avec les phrases déjà incluses dans le résumé. Le modèle intègre en outre la contrainte de la taille maximale du résumé. Ce problème a ensuite été reformulé par Gillick and Favre (2009) en définissant une fonction objectif se focalisant sur la maximisation du poids des bigrammes de mots sélectionnés, toujours sous la contrainte de la longueur maximale du résumé. La non-redondance est quant à elle favorisée de façon implicite. Le poids de chaque bigramme n'étant comptabilisé qu'une seule fois dans la fonction objectif, indépendamment de son nombre d'occurrences dans le résumé final, cette fonction tend à être d'autant plus élevée qu'un nombre plus large de bigrammes

2. Optimisation linéaire en nombres entiers

est sélectionné, ce qui conduit aussi à limiter le nombre d'occurrences de chaque bigramme et donc, la redondance.

Par la suite, différents travaux ont proposé des modèles ILP plus élaborés tandis que d'autres ont mis en œuvre des traitements plus ciblés en amont de la phase ILP. Li et al. (2011a) suggèrent ainsi de regrouper les phrases par aspect, en l'occurrence de nature événementielle (qui, où ...), et de garder un représentant par cluster, constitué par la compression de ses phrases. La partie ILP se charge de sélectionner les phrases maximisant l'inclusion des aspects les plus importants dans le résumé. La performance de cette approche dépasse légèrement celle des baselines classiques. L'approche de Woodsend and Lapata (2012) répartit quant à elle la sélection de phrases sur des modules indépendants. Chacun prend en compte un critère différent (ex. couverture en bigrammes, position et style des phrases, compression des phrases, etc.). La sortie de ces modules est passée au programme ILP dont l'objectif est de maximiser le score des phrases donné par la contribution des différents modules de sélection de phrases. Des méthodes supervisées ont aussi montré leur efficacité, particulièrement l'estimation de la fréquence des bigrammes dans le résumé par un modèle de régression. Les fréquences prédites sont considérées comme les poids des bigrammes dans le modèle ILP (Li et al., 2013). D'autres travaux (Li et al., 2015b) se sont intéressés spécifiquement à la pondération des bigrammes en combinant l'utilisation des critères internes, comme les fréquences et les positions des bigrammes dans les documents, et des ressources externes comme WordNet, Wikipédia et DBpedia. Le problème du compromis entre la performance et l'efficacité a aussi été abordé. Il a ainsi été établi que l'élagage des bigrammes peu fréquents améliore la vitesse de l'optimisation mais se fait aux dépens des scores ROUGE (cf. 2.6.1). Une approche approximative par agrégation de plusieurs solutions optimales a été proposée comme solution possible à ce problème (Boudin et al., 2015).

2.5 Le résumé multi-document et le résumé de mise à jour

Depuis quelques années déjà, les recherches se concentrent beaucoup plus sur le résumé multi-document que sur le résumé mono-document. Plus récemment, a émergé le résumé dynamique. Nous décrivons dans ce paragraphe les spécificités de chaque type de résumé et les contraintes qu'il impose.

2.5.1 Résumé multi-document

Un système de résumé multi-document permet de produire un résumé d'une collection de textes en rendant compte de ses idées principales. Les méthodes de résumé citées à la section précédente peuvent être appliquées pour le résumé mono ou multi-document. Cependant, certaines sont plus adaptées que d'autres au résumé multi-document. Par exemple, les méthodes fondées sur la programmation linéaire ont montré plus de succès que les méthodes fondées sur les graphes. En effet, la pluralité des documents impose de nouvelles contraintes que nous détaillons ci-dessous.

Redondance inter-document

Le problème de la redondance est davantage présent dans le cadre du multi-document, apparaissant à deux niveaux : entre les phrases du même document et entre les phrases de différents documents. Il se pose de façon plus aiguë encore lorsque les documents à résumer sont thématiquement homogènes, ce qui est souvent le cas. Par exemple, si les textes source sont des articles de presse concernant le même événement, il est très probable que les phrases les plus importantes de chaque texte soient très similaires. L'adoption d'une approche de résumé statistique fondée sur la fréquence d'occurrence des mots conduit à un résumé tendant à surreprésenter la même information. Bien que cette information soit la plus pertinente dans tous les documents, le résumé obtenu est pauvre et ne rappelle pas tout ce dont parle l'ensemble des textes. Ce type d'approches convient plus à l'identification du sujet principal d'une collection de documents. Pour résoudre le problème de redondance inter-document, différentes solutions ont été propo-

sées. La première famille d'approches commence en premier lieu par ordonner les phrases par ordre décroissant de pertinence. Dans un second temps, les phrases du résumé sont sélectionnées en débutant par les phrases les mieux notées et en comparant chaque phrase aux phrases déjà choisies pour le résumé. Si leur similarité dépasse un seuil donné, la phrase n'est pas retenue pour le résumé. Dans cette lignée, on trouve en particulier la MMR. Ce critère attribue aux phrases un score consistant en une combinaison pondérée de pertinence et de redondance avec les phrases déjà incluses dans le résumé. Les résumés sont créés selon une procédure gloutonne ajoutant de manière incrémentale les phrases qui maximisent ce critère. Les algorithmes fondés sur la MMR ont constitué pendant des années une référence notable dans le résumé.

La deuxième famille d'approches repose dans la plupart des cas sur l'analyse thématique des documents exploitant des facteurs superficiels comme la similarité lexicale ou des facteurs plus profonds comme la similarité sémantique. Cette dernière permet de détecter la redondance des informations au-delà d'une similarité de surface. En effet deux phrases peuvent exprimer exactement la même information sans pour autant avoir des mots en commun.

Clustering de documents. Une manière assez répandue d'aborder le résumé multi-document est d'adopter une approche en deux temps (Radev et al., 2004). Les documents similaires sont d'abord regroupés en *clusters*. Chaque cluster est ensuite résumé en extrayant une ou plusieurs phrases des documents qu'il contient. Cette extraction peut le cas échéant être réalisée par des méthodes mono-document en considérant le cluster comme un unique document. Le résumé final est la concaténation des phrases représentant chacun des clusters. Cette façon de faire permet en particulier de limiter la combinatoire de recherche des redondances entre phrases.

Segmentation thématique. La segmentation thématique permet de découper les textes en segments contigus thématiquement homogènes en s'appuyant sur la distribution du vocabulaire dans les textes ou sur des marques linguistiques. Dans le cadre du résumé multi-document, elle constitue un outil permettant de travailler avec des unités textuelles homogènes entre le niveau du texte et celui de la phrase et facilite ainsi la détection des similarités thématiques tout en réduisant la combi-

natoire des comparaisons (Angheluta et al., 2002, Ferret et al., 2004). TextTiling (Hearst, 1997) est un exemple d'algorithme de segmentation thématique très utilisé dans ce cadre (Neto et al., 2000).

Identification des thèmes. D'autres chercheurs ont choisi d'identifier d'abord les thèmes ou les événements majeurs mentionnés dans le texte (Arora and Ravindran, 2008). Ensuite, ils classifient les phrases par thème et choisissent une ou plusieurs phrases pour couvrir chaque thème.

Problème combinatoire

Nous avons montré précédemment que les approches classiques du résumé automatique mono-document ne sont pas toujours adéquates dans le cas du multi-document parce qu'elles ne prennent pas en compte la redondance inter-document. Une autre raison de l'insuffisance de ces méthodes est qu'elles ont été conçues pour opérer sur un seul document à la fois, c'est-à-dire sur des données de petite taille. Le passage au résumé multi-document signifie le passage à des données plus volumineuses. Certaines approches, (Li et al., 2011a) pour ne citer qu'un exemple, organisent les traitements effectués sur les textes en pipeline. Cette architecture oblige à parcourir l'ensemble des documents autant de fois que le nombre de traitements à réaliser. Une telle organisation est très coûteuse et peut réduire considérablement l'utilisabilité du système proposé. La grande taille des données doit être prise en compte en amont de la conception du modèle de façon à réduire les parcours séquentiels des documents et paralléliser les opérations au maximum.

2.5.2 Résumé dynamique : une dimension temporelle

Le résumé dynamique est une variante du résumé automatique multi-document incluant la dimension supplémentaire du temps. Alors que dans le problème du résumé multi-document les données d'entrée sont statiques, le résumé dynamique introduit une difficulté supplémentaire en faisant varier les données d'entrée sur l'axe du temps. Les travaux sur ce type de résumé peuvent être classés en deux catégories. Les systèmes de résumé dynamiques séquentiels produisent un résumé rendant compte de l'information portée par les documents couvrant une période donnée en prenant comme point de référence les informations connues juste avant

cette période, incarnées par un résumé (Yang et al., 2013, Xu et al., 2013). Les systèmes de résumé dynamiques incrémentaux produisent quant à eux des mises à jour d'un résumé initial à chaque fois que des informations nouvelles apparaissent concernant l'objet du résumé initial (Chowdary and Kumar, 2008, McCreadie et al., 2014).

Formalisation du problème

La formalisation classique du problème considère deux instants t et $t+1$. Étant donné un ensemble de documents A à l'instant t et un autre ensemble B à l'instant $t+1$ plus récent, il s'agit de produire un résumé des textes de l'ensemble B sous l'hypothèse que le lecteur a déjà pris connaissance de toutes les informations apportées par l'ensemble A. Autrement dit, il faut résumer les documents de B sans répéter ce qui a été évoqué dans A. Ceci peut être considéré comme la combinaison des problèmes du résumé automatique et de la détection de nouveauté.

Redondance à travers le temps

La nouvelle contrainte imposée par ce type de résumé est la gestion de la redondance à travers le temps entre les deux ensembles A et B, qui s'ajoute aux contraintes de redondance inter-document et intra-document héritées respectivement du résumé multi-document et mono-document. Une première approche adoptée pour répondre à cette question a réduit le problème du résumé dynamique en un problème de résumé multi-document. Un résumé est généré d'abord pour chaque ensemble (A et B). Ensuite, le résumé de B est modifié de façon à éliminer ce qui est redondant avec le résumé de A. Cette méthode n'est pas très performante car seul le contenu du résumé de l'ensemble A n'est pas autorisé à apparaître dans le résumé de B. Rien n'empêche alors que d'autres informations des documents de A soient incluses dans le résumé de B. C'est pourquoi les meilleurs systèmes de résumé dynamiques actuels considèrent la totalité des textes de A pour l'élimination de la redondance. Les solutions peuvent être plus précisément classées en deux catégories : des solutions par élimination et des solutions par évitement. Les solutions par élimination traitent l'ensemble de documents B sans aucune prise en compte de l'ensemble A. Une fois les phrases sélectionnées ou or-

données, la redondance est éliminée par la suppression des phrases similaires au contenu de l'ensemble A. Les solutions par évitement considèrent au contraire la redondance comme critère lors de l'attribution des scores. Dans certains cas, la redondance peut être justifiée voire bénéfique. En effet, les informations marginales de l'ensemble A peuvent acquérir plus d'importance à travers le temps. Elles apparaissent alors comme des informations principales dans B. Actuellement, les systèmes conçus ne considèrent pas ce cas.

2.6 L'évaluation du résumé automatique

L'évaluation des résumés automatiques est une problématique importante à laquelle les travaux de recherche n'ont répondu que partiellement. Avec le développement du domaine et l'abondance des travaux proposés, des campagnes d'évaluation annuelles (DUC, TAC, TREC) ont été organisées afin de comparer les systèmes de résumé. Les premières évaluations reposaient sur le jugement des lecteurs concernant la qualité linguistique et le contenu du résumé, soit en estimant la similarité des résumés candidats avec un résumé manuel (évaluation objective), soit en jugeant la qualité du résumé sans se référer à un modèle (évaluation subjective). La dernière variante correspond à la mesure *Responsiveness* utilisée jusqu'à aujourd'hui pour évaluer le résumé de point de vue du contenu et de la qualité linguistique. Ces méthodes nécessitent un fort investissement en temps et en effort, ce qui pose problème pour le développement des systèmes de résumé. C'est pourquoi des métriques standard, avec une mise en œuvre automatique, ont été proposées pour rendre plus facile la comparaison des différentes approches. Les méthodes répondant à cette problématique s'intéressent plus à l'évaluation du contenu sélectionné qu'à la qualité linguistique ou grammaticale. Par ailleurs, l'automatisation n'est que partielle. En effet, pour juger un résumé, celui-ci est comparé à un résumé manuel (idéal, modèle ou de référence). Ces systèmes dépendent donc de la disponibilité des résumés manuels.
Dans ce qui suit, nous présentons les deux méthodes d'évaluation semi-automatique les plus utilisées : ROUGE et PYRAMID. Leur succès est en particulier lié à leurs fortes corrélations avec les jugements humains. Nous donnons ensuite un aperçu sur les travaux en cours sur l'automatisation complète des systèmes d'évaluation.

2.6.1 ROUGE

ROUGE évalue les résumés en les comparant à des résumés modèles. Cette comparaison est automatique et ne nécessite pas de prétraitement particulier. Elle est déduite à partir du recouvrement entre les N-grammes des deux textes. Elle utilise trois métriques pour quantifier la comparaison.

Précision. Elle traduit à quel point les données sélectionnées sont pertinentes. Concrètement, il s'agit du rapport du nombre d'unités textuelles (N-grammes) communes au résumé candidat et aux résumés de référence sur le nombre de toutes les unités textuelles du résumé candidat.

Rappel. Il reflète à quel degré le résumé candidat rappelle (évoque) des données pertinentes qu'il est censé inclure. Il désigne le rapport des unités textuelles communes aux résumés candidat et de référence sur le nombre de toutes les unités textuelles du résumé de référence.

F-mesure. C'est la moyenne harmonique de la précision et du rappel. D'après les résultats d'évaluation des systèmes de résumé, le rappel est généralement plus difficile à obtenir que la précision.

Cette méthode a montré une forte corrélation avec les jugements humains (Lin, 2004). La corrélation de Pearson des scores ROUGE-2 [3] avec les jugements humains, pour le résumé multi-document, varie entre 0,85 et 0,94 en utilisant 3 résumés de référence et en éliminant les mots vides. Cette corrélation augmente avec le nombre de résumés modèles. Il existe plusieurs variantes de ROUGE exploitant des modèles autres que les N-grammes, comme la plus longue sous-séquence commune ou les bi-grammes distants. Comme l'indique ses premières lettres, ROUGE est orienté rappel (*Recall Oriented*). La dernière implémentation de ROUGE permet de calculer en plus la précision et la f-mesure. Jusqu'à présent ROUGE est l'outil d'évaluation le plus utilisé (Hong et al., 2014).

2.6.2 PYRAMID

Cette méthode permet de comparer un résumé candidat à un ensemble de résumés de référence (Nenkova and Passonneau, 2004). Étant donné qu'un résumé idéal n'existe pas et que les styles de rédaction diffèrent d'une personne à l'autre, l'uti-

3. ROUGE-N avec des N-grammes des longeur N

lisation d'un seul résumé de référence ne satisfait pas la condition d'équité entre les résumés candidats. Pour relaxer cette contrainte, les campagnes d'évaluation présentent au moins 4 résumés modèles. Le principe de la méthode PYRAMID consiste à annoter les résumés de référence afin d'identifier les unités appelées SCUs (*Summary Content Units*). Un SCU est un ensemble d'unités textuelles des résumés de référence exprimant la même information. Il lui est assigné un poids égal au nombre de résumés de référence qui l'instancient. Ces SCUs peuvent être organisés en pyramide où chaque couche regroupe les SCUs de même poids. Pour évaluer un résumé, ce dernier est annoté afin de repérer les SCUs candidats qu'il contient. Par la suite, chaque SCU candidat hérite du poids du SCU le plus similaire dans la pyramide. Le score PYRAMID du résumé est finalement le rapport de la somme des poids de tous ses SCUs candidats sur la somme des poids d'un résumé idéal ayant le même nombre de SCUs. L'inconvénient de cette méthode est qu'elle nécessite une étape d'annotation des résumés. Le calcul du score PYRAMID a été automatisé en utilisant la sémantique distributionnelle (Passonneau et al., 2013). Malheureusement, l'annotation des résumés modèles reste difficile à automatiser.

2.6.3 Autres méthodes d'évaluation automatique

Les méthodes d'évaluation citées ci-dessus restent assez coûteuses en termes de temps et de ressources humaines à mobiliser. Elles ne permettent pas ainsi de mettre en œuvre des évaluations à large échelle. De plus, il faut noter que les résumés manuels ne sont pas forcément idéaux. Cette imperfection provient en partie de la subjectivité de la personne qui rédige un résumé et de ce fait, on peut remarquer dans certains cas un manque de consensus entre les différents résumés manuels d'un même texte source (Van Halteren and Teufel, 2003). Ce problème a motivé les chercheurs pour proposer des méthodes d'évaluation entièrement automatiques, c'est-à-dire sans avoir besoin de résumés de référence. Ces méthodes sans référence permettent non seulement de réduire le coût de l'évaluation mais aussi de contourner le problème de la qualité des résumés manuels et de leur disponibilité. Une des solutions proposées dans ce contexte consiste à utiliser un moteur de recherche pour ordonner un ensemble de documents et un ensemble

de leur résumés par ordre de pertinence par rapport à une requête donnée (Radev et al., 2003). Les meilleurs systèmes de résumé sont ceux qui préservent le plus l'ordre de classement entre les documents source et leurs résumés. Ensuite, Louis and Nenkova (2008) ont proposé le système SIMetrix [4], implémentant une méthode d'évaluation automatique fondée sur des métriques de similarité entre les documents source et les résumés générés. Cette méthode calcule plus précisément la divergence entre la distribution de probabilité du vocabulaire du résumé produit et celle du texte source. SIMetrix permet d'atteindre une corrélation de 0,88 avec le score PYRAMID et 0,74 avec le score Responsiveness (Louis and Nenkova, 2013). Fresa [5] est un autre système d'évaluation sans référence (Torres-Moreno et al., 2010). Comme SIMetrix, il utilise des mesures de divergence pour estimer la qualité d'un résumé. Il permet d'utiliser la divergence de Kullback-Leibler et celle de Jensen-Shanon. Il offre également la possibilité de calculer les divergences en utilisant différents types de ngrammes, comme dans ROUGE. Le même système a été étendu dans la perspective du multilinguisme (Saggion et al., 2010). Toujours dans la même idée, SummTriver (Cabrera-Diego et al., 2016, Cabrera-Diego and Torres-Moreno, 2018) est un modèle fondé sur plusieurs divergences mais qui propose d'ajouter une composante supplémentaire afin de mieux évaluer les résumés. En effet, ce système calcule simultanément la dissimilarité entre trois éléments : le résumé à évaluer, ses documents source et un ensemble de résumés automatiques de la même source mais produits avec des méthodes différentes. Les dissimilarités entre ces éléments sont calculées en utilisant la divergence de Kullback-Leibler ou de Jensen-Shanon et sont par la suite fusionnées en utilisant la trivergence de distributions de probabilité (Torres-Moreno, 2015).

Entre 2009 et 2011, TAC a proposé la tâche AESOP (*Automatically Evaluating Summaries Of Peers*) pour encourager le développement des méthodes automatiques d'évaluation des résumés. Dans ces tâches, des résumés de référence sont fournis et l'objectif consiste à proposer des méthodes d'évaluations qui dépassent les baselines en termes de corrélation avec les métriques Pyramid et *Responsivness*. Deux variantes de la tâche sont proposées : *all peers* et *no models*. La première

4. http://homepages.inf.ed.ac.uk/alouis/IEval2.html
5. http://fresa.talne.eu

consiste à évaluer les résumés système et les résumés de référence avec la mesure d'évaluation proposée alors que la deuxième consiste à évaluer les résumés systèmes contre les résumés de référence. L'intention derrière la première variante est de vérifier à quel point la méthode d'évaluation est capable de différencier entre les résumés automatiques, généralement extractifs, et les résumés de référence abstractifs. La seconde variante a pour but de vérifier la capacité de la méthode proposée à évaluer des résumés automatiques. En 2010, 27 systèmes ont participé à la seconde variante *no models*. ROUGE-2 et ROUGE-SU4 étaient parmi les baselines à dépasser. Globalement, ROUGE-2 était toujours parmi les meilleures méthodes (Conroy et al., 2010). Il était en dessus de toutes les méthodes en termes de corrélation avec Pyramid et en troisième position en termes de corrélation avec *Responsivness*. Ces résultats prouvent que ROUGE est une métrique performante et promettent en même temps une future amélioration de l'évaluation automatique des résumés.

2.7 Synthèse : tableau comparatif des travaux récents en RA

Afin de pouvoir comparer les systèmes de l'état de l'art, nous avons essayé de présenter un tableau comparatif rassemblant un large ensemble de méthodes différentes. Nous décrivons brièvement le principe de chaque méthode présentée et nous reportons sa performance en termes de scores ROUGE-2 et ROUGE-1 ou ROUGE-SU4, notés respectivement R-2, R-1 et RSU4. Le choix de ces trois métriques pour la comparaison était presque imposé étant donné que la majorité absolue des travaux de recherche sur le RA utilisent ROUGE comme méthode d'évaluation et particulièrement ses métriques R1, R2 ou RSU4 pour leur haute corrélation avec le jugement humain. ROUGE-1 (respectivement ROUGE-2) mesure le chevauchement entre les unigrammes (respectivement bigrammes) des résumés produits et des résumés manuels. ROUGE-SU est l'acronyme de *ROUGE skip-bigram and unigram*. Un *skip-gram* est une paire de mots présentés dans l'ordre dans la phrase et séparés de n mots au maximum. Dans ROUGE-SU4, n vaut 4. Par exemple, dans la phrase suivante :

« Ce travail porte sur le résumé automatique »

ROUGE-SU4 va extraire les 7 unigrammes de la phrase ainsi que les *skip-grams* : « Ce travail », « Ce porte », « Ce sur », « Ce le », « Ce résumé », « travail porte », et ainsi de suite.

Nous regroupons les systèmes présentés par familles de méthodes. Dans les tableaux présentés, nous indiquons pour chaque méthode le jeu de données utilisé (notés « JD »). Les jeux de données reportés sont les données d'évaluation proposées dans le cadre des campagnes d'évaluation DUC 2001, DUC 2002, DUC 2003, DUC 2004, TAC 2008, TAC 2009, TAC 2010, TAC 2011 (notées respectivement D01, D02, D03, D04, T08, T09, T10, T11). Dans le tableau, les valeurs de ROUGE précédées par la lettre F font référence à la F-mesure de ROUGE alors que celles précédées par la lettre R font référence à son rappel.

2.7.1 Résumé multi-document

Méthodes fondées sur les graphes

Système	R-1	R-2	JD	Description
(Shen and Li, 2010)	F39,95	F10,48 F09,01	D04 T08	Cette approche représente classiquement les documents à résumer par un graphe dont les nœuds sont les phrases. Deux phrases sont connectées par un arc si leur similarité est supérieure à un seuil donné. Le problème se ramène ensuite à trouver l'ensemble dominant D du graphe qui est un sous-ensemble de sommets tels que tout sommet qui n'appartient pas à D possède au moins un arc en commun avec un des sommets de D.
(Bhaskar and Bandyopadhyay, 2010)	R53,21	R10,31	T08	Cet article décrit une méthode de résumé multi-document orientée requête. Le système commence par regrouper les phrases similaires à partir du graphe de similarité entre phrases. Ensuite, un score de proximité avec la requête est attribué à chaque nœud. La phrase la mieux classée de chaque cluster est retenue. Les phrases retenues sont compressées en utilisant un analyseur syntaxique.

Méthodes fondées sur l'ILP

Système	R-1	R-2	JD	Description
(Sripada and Jagarlamudi, 2009)	F38,60	F09,00	D04	Cette approche part de l'hypothèse qu'un résumé qui représente bien un ou plusieurs documents doit avoir la même distribution de probabilité d'unigrammes de mots que celles de ces documents. Par conséquent, pour produire le résumé, le système sélectionne des phrases de manière à ce que la distribution de probabilité des unigrammes de mots au sein du résumé produit soit comparable à la distribution de probabilité des unigrammes dans les documents source à résumer.
(Takamura and Okumura, 2009)	F28,30	F08,30	D04	Ce travail traite le problème de RA comme un problème de couverture maximale. L'objectif est de maximiser la couverture en unités conceptuelles (les mots) mais aussi d'optimiser la pertinence par rapport à la collection de documents à résumer. Pour résoudre ce problème, plusieurs algorithmes ont été utilisés, dont une approche gloutonne qui a donné les meilleures performances.

Système	R-1	R-2	JD	Description
(Woodsend and Lapata, 2012)	-	R11,83	T08	Cet article propose une solution essayant de couvrir différents aspects du résumé multi-document tels que la sélection du contenu, la minimisation de la redondance, la génération de paraphrases, etc. Ces aspects sont appris séparément en utilisant des prédicteurs spécifiques entraînés sur des documents source et des résumés manuels. Ensuite ces critères sont optimisés conjointement dans un modèle (ILP) pour générer le résumé final. La partie ILP permet de combiner les décisions des prédicteurs qui collaborent pour réécrire le contenu en utilisant des règles extraites de groupes de documents et de résumés de modèles.
(Li et al., 2013)	-	F10,76	T08	Cet article propose d'exploiter le modèle ILP proposé par (Gillick and Favre, 2009) dans un cadre supervisé. Pour chaque bigramme, au lieu d'utiliser la fréquence comme poids, un modèle de régression estime la fréquence de ce bigramme dans le résumé de référence. Le modèle de régression se sert d'un ensemble de critères (fréquence, position et longueur de la phrase qui contient le bigramme, etc) et vise à minimiser la distance entre la fréquence estimée et la fréquence réelle. Ensuite le modèle ILP se charge de sélectionner les phrases du résumé.

Système	R-1	R-2	JD	Description
(Martins and Smith, 2009)	F40,30	F18,00	D02	Ce système opère simultanément sur la sélection et la compression de phrases dans le cadre d'un modèle ILP. La compression est formulée en utilisant l'analyse syntaxique en dépendances dans les modèles à bigrammes de mots

Méthodes fondées sur l'apprentissage automatique

Système	R-1	R-2	JD	Description
(Ren et al., 2016)	R36,31 R37,80 R39,60	R08,49 R09,61 R10,57	D01 D02 D04	Cette méthode utilise un modèle de régression mais à la différence des systèmes de RA classiques fondés sur la régression qui modélisent l'importance et la redondance séparément, cette méthode estime l'importance d'une phrase relativement à un ensemble de phrases sélectionnées pour le résumé.
(Hong et al., 2015)	F35,26 F38,23 F39,59 F39,95 F39,78	F07,88 F09,46 F10,18 F10,48 F12,08	D01 D02 D03 D04 T08	Ce système permet d'agréger de façon optimale différents systèmes de RA en supposant que même si les performances de différents systèmes sont proches, chacun de ces systèmes capte une partie différente de la solution souhaitée. Ce système supervisé commence par générer des résumés candidat à partir des résumés des systèmes à agréger en énumérant les combinaisons de phrases possibles. Ensuite, un modèle de régression utilisant un ensemble de 360 critères de sélection détermine le meilleur résumé candidat.

Système	R-1	R-2	JD	Description
(Ryang and Abekawa, 2012)	F39,01	F09,47	D04	Ce travail construit un résumé extractif par un processus d'apprentissage par renforcement. Un état correspond à un résumé candidat et une action correspond à une insertion d'un élément textuel. À chaque fois qu'une action est exécutée, le système reçoit soit une récompense, le score du résumé actuel selon la fonction de score, soit une pénalité si la longueur du résumé dépasse la longueur maximale tolérée. La fonction de score est un compromis entre la pertinence et la non redondance des phrases en question.
(Louis, 2014)	R34,10	R07,95	D04	Cette approche utilise un modèle bayésien dédié à la détection de la surprise par rapport à un background représentant les connaissances acquises dans un contexte déterminé. Cette méthode quantifie la capacité d'une information à changer la croyance de quelqu'un par rapport à un contexte donné. Pour modéliser le background, 5000 articles du corpus English Gigaword ont été aléatoirement sélectionnés.
(Rioux et al., 2014)	R40.33	R11,39	D04	Ce travail étend les travaux de (Ryang and Abekawa, 2012) en utilisant l'apprentissage par renforcement mais avec une nouvelle fonction de récompense et un ensemble de critères différent.

Système	R-1	R-2	JD	Description
(Rush et al., 2015)	R28,18	R08,49	D04	Cet article propose une méthode de résumé par abstraction orientée donnée, c'est-à-dire qui ne dépend pas de critères linguistiques ou syntaxiques comme la majorité des travaux sur le résumé abstractif. L'approche proposée est inspirée de la traduction automatique neuronale. Elle est comporte trois composants majeurs : un encodeur, qui apprend un alignement latent des données en entrée, est entraîné conjointement avec un modèle de génération du langage. Le troisième composant est le décodeur qui génère le résumé final en se basant sur l'algorithme Beam Search.
(Bhaskar, 2013)	R50,62	R10,54	T08	Cette méthode se focalise sur le résumé multi-document générique tout en l'abordant comme un résumé orienté requête. En premier lieu, les mots-clés des différents documents sont identifiés par une méthode reposant sur les champs aléatoires conditionnels (CRF). Les mots-clés extraits sont ensuite considérés comme une requête et un résumé orienté requête est généré de la même façon que dans le système ci-dessus.
(Almeida and Martins, 2013)	-	R12,30	T08	Ce travail présente une méthode de résumé fondée sur la décomposition Lagrangienne qui extrait et compresse les phrases conjointement. Ceci est réalisé en entraînant un modèle d'apprentissage multi-tâches sur les tâches d'extraction et de compression.

2.7.2 Résumé dynamique

Méthodes fondées sur les graphes

Système	R-2	RSU4	JD	Description
(Wenjie et al., 2008)	R08,95	R12,91	D07	Ce système s'inscrit dans la lignée des méthodes à base de graphes (Mihalcea and Tarau, 2004, Erkan and Radev, 2004a,b). Les phrases de A et B, avec A les documents initiaux et B les nouveaux documents, sont représentées par le même graphe sur lequel un algorithme de renforcement positif et négatif est exécuté. Il favorise le poids d'un nœud de la collection B si ce nœud est bien corrélé avec les autres nœuds de B : renforcement positif. Par contre, les nœuds de B ayant une forte corrélation avec les nœuds de A sont pénalisés : renforcement négatif.
(Katragadda et al., 2009)	R09,31	R12,82	D07	Ce système propose une nouvelle baseline pour remplacer la baseline des premières phrases. Il dérive une méthode d'ordonnancement des phrases en se basant sur une politique de position sous-optimale (OPP). L'OPP est induite en analysant la distribution des SCUs *(Summary Content Unit)*, comme définis par la méthode PYRAMID, dans les différentes positions de phrases dans l'ensemble des données DUC07.

(Li et al., 2011b)	F07,80	F11,97	T09	Dans cet article, les auteurs proposent d'appliquer une méthode de régularisation fondée sur les graphes appelée MarginRank pour le résumé de mise à jour. MarginRank étend la fonction de coût de la méthode d'ordonnancement sur des données Manifold (Zhou et al., 2004) en supprimant les termes de A apparaissant dans B. Elle classe les phrases dans B de manière à ce que les phrases les mieux classées soient les plus pertinentes mais aussi celles couvrant un contenu différent de A.
(Wang et al., 2015)	F09,10	F13,60	D07	Cette approche représente toutes les phrases (de A et B) dans un seul graphe où les phrases de A sont étiquetées 1 si elles ont été sélectionnées pour le résumé de A et 0 si elles n'ont pas été sélectionnées. Les nœuds correspondant aux phrases de B ne sont pas étiquetées. Les arcs représentent la similarité cosinus entre les phrases. Ensuite, une méthode de propagation d'étiquettes pour étiqueter les nœuds de B et retenir les phrases étiquetées 1 pour le résumé.

Méthodes fondées sur l'ILP

Système	R-2	RSU4	JD	Description
(Boudin and Torres-Moreno, 2009a)	F09,38	F13,05	D07	Ce système attaque le problème du résumé mis-à-jour guidé par la requête en proposant une fonction de score des phrases maximisant la pertinence par rapport au thème (requête) et minimisant le redondance avec l'historique (résumé de la collection A). Pour renforcer encore la détection de nouveauté, ce système représente chaque collection de documents par un sac des mots ayant les scores $tf \times idf$ les plus élevés. Ensuite, l'intersection des deux sacs de mots est éliminée du sac de mots de la collection B. L'ensemble obtenu caractérise exclusivement la collection B. Il est utilisé par la suite pour enrichir la requête/thème de l'ensemble B.
(Li et al., 2015a)	F09,99 F09,61 F09,99	F13,61 F13,77 F13,42	T08 T09 T11	Cet article propose deux nouvelles modifications dans le cadre de l'ILP pour le résumé mis-à-jour. L'idée clé est d'utiliser un modèle discriminant avec un ensemble de critères qui mesurent à la fois la saillance et la nouveauté des mots et des phrases. D'abord, ces critères sont utilisés dans un modèle supervisé pour prédire les poids des concepts utilisés dans le modèle ILP. En second lieu, des phrases candidates pour le résumé sont sélectionnées par le modèle ILP. Finalement, les phrases candidates sont classées en utilisant les critères de niveau phrastique.

(Nóbrega and Pardo, 2016)	R08,60	-	D07	Ce système utilise 3 critères pour pondérer les phrases. La MMR est utilisée pour donner un score plus important aux phrases plus similaires aux phrases des nouveaux documents qu'aux phrases des anciens documents. Le facteur de nouveauté d'une phrase indique si les mots qui la composent apparaissent plus dans les anciens documents ou dans les nouveaux. Finalement, quatre fonctions de classement des phrases selon la position sont utilisées. La nouveauté apportée par ce système est l'application de la segmentation thématique pour partitionner le texte en sections. Ensuite, pour pondérer les phrases, les critères décrits ci-dessus sont appliqués à chaque phrase et à chaque section à laquelle elle appartient.
(Boudin et al., 2010)	R07,45	R11,58	T08	Cet article décrit une méthode de résumé mis-à-jour qui repose sur une maximisation à double critère. Une MMR modifiée et appelée SMMR est utilisée pour sélectionner les phrases qui sont proches du thème/requête et en même temps, distantes des phrases contenues dans les documents déjà lus. Les résumés sont ensuite générés en assemblant les phrases les mieux classées et en appliquant des règles de post-traitement linguistique pour réduire la taille du résumé et garder sa cohérence.

Méthodes fondées sur l'apprentissage automatique

Système	R-2	RSU4	JD	Description
(Bysani et al., 2009)	R10,34	R14,26	TAC08	Ce système utilise des critères de saillance et de nouveauté pour pondérer les phrases. La position, la fréquence et le poids $tf \times idf$ sont utilisés pour quantifier la pertinence des phrases. L'apport de nouveauté d'un mot consiste en un ratio du nombre de nouveaux documents qui le contiennent par le nombre des anciens documents où il apparaît. Pour pondérer la contribution de ces critères dans la fonction de score, un modèle de régression est entraîné sur des données d'apprentissage.
(Louis, 2014)	R07,67	-	T09	C'est l'approche décrite dans 2.7.1 pour le résumé multi-document. Pour l'adapter au résumé mis-à-jour, l'ensemble des documents antérieurs A est considéré comme le background et le système de résumé sélectionne les phrases de forte valeur de surprise.

Autres méthodes

Système	R-2	RSU4	JD	Description
(Chali and Uddin, 2016)	R10,74	-	T11	Cette approche prend en compte les événements apparus dans chaque phrase et leurs relations temporelles pour estimer sa nouveauté ainsi que sa saillance. Le problème est ensuite ramené à un ordonnancement chronologique des phrases selon les expressions et les événements temporels qu'elles contiennent.

(Wu and Guo, 2015)	F10,20	F13,60	T08	Ce système utilise l'Analyse Sémantique Latente (LSA) pour la génération des résumés. Pour appliquer la LSA au résumé de mise à jour, il utilise les *topic signature* pour extraire les termes liés aux informations nouvelles. Cette information est exploitée pour déterminer l'apport de nouveauté de tous les termes. Finalement, les termes les moins pertinents et qui apportent peu de nouveauté par rapport aux anciens documents sont exclus lors de la génération de résumé.
(Bysani, 2010)	R10,27	R13,92	T09	Ce système commence par ordonnancer les phrases selon leur apport de nouveauté en utilisant deux critères : le facteur de nouveauté (rapport des nouveaux mots sur la somme des anciens mots et de tous les mots de l'ensemble A) ainsi que le nombre de nouveaux mots qu'une phrase contient normalisé par sa longueur. Ensuite, l'ensemble des phrases est réordonnancé en utilisant une MMR qui opère un compromis entre le rang d'origine d'une phrase et de sa nouveauté matérialisée par la non redondance avec les anciennes informations. Deux mesures de similarité sont utilisées pour détecter la redondance : ITSim est une mesure fréquentielle inspirée de la théorie d'information et CoSim est la similarité cosinus entre les vecteurs $tf \times idf$ de deux phrases.

(Delort and Alfonseca, 2012)	R09,24	R12,85 T11		Il s'agit d'un modèle probabiliste non supervisé qui vise à apprendre la différence entre les thèmes des anciens et nouveaux documents en modélisant la distribution des thèmes au niveau des documents contrairement à la majorité des travaux qui modélisent cette distribution au niveau des phrases.
(Huang and He, 2010)	R08,80	-	T11	Ce système commence par extraire les thèmes des deux ensembles A et B en utilisant l'Allocation de Dirichlet Latente (LDA). Ces thèmes sont catégorisés en 4 catégories : — émergents : les sujets qui émergent de B — activants : sujets appartenant aux deux ensembles mais plus cohérent avec l'ensemble B — non-activants : sujets appartenant aux deux ensembles mais très peu évoqués dans l'ensemble B — périssants : sujets évoqués seulement dans A La corrélation entre les thèmes des anciens et nouveaux documents est calculée pour capturer le changement de sujet entre les deux collections de documents. L'algorithme proposé classe les phrases de l'ensemble B en intégrant l'évolution du sujet dans le processus de classement et de sélection des phrases.

2.8 Conclusion

Nous avons caractérisé dans ce chapitre les différents aspects du résumé multi-document et du résumé mis-à-jour. Nous avons également présenté les méthodes de l'état de l'art les plus connues. Malgré la grande diversité des méthodes reportées dans le tableau récapitulatif, on peut constater que les performances des systèmes ne sont pas très différentes sur les mêmes jeux de données. Le succès des approches fondées sur la fusion de plusieurs systèmes de résumé (Hong et al., 2015) montre néanmoins que cette relative homogénéité des résultats recouvre une certaine diversité des résumés produits. Par ailleurs, on peut tout de même noter que les méthodes fondées sur l'ILP ainsi que les méthodes fondées sur l'apprentissage supervisé sont en général un peu au dessus des autres méthodes. Nous remarquons aussi que la performance moyenne des systèmes sur la tâche du résumé dynamique est inférieure à celle des systèmes de résumé multi-document générique. Ceci peut être dû à la complexité de cette tâche, qui est moins ancienne que la thématique du résumé générique. Globalement, il existe encore une marge d'amélioration dans les méthodes extractives pour le résumé mis-à-jour, une hypothèse que nous vérifions dans le chapitre 4. Techniquement, le défi qui s'impose est de pouvoir renforcer la détection de nouveauté.

Chapitre 3

Intégration de la similarité sémantique pour le RA

3.1 Introduction

Les travaux menés dans le cadre du RA par extraction ont vu la proposition d'un grand nombre de critères de sélection de phrases et d'intégration des résultats de cette sélection pour former un résumé. Toutes ces propositions ont plus ou moins explicitement pour objectif de faire un compromis entre le respect d'une contrainte de taille maximale du résumé à produire, la maximisation de son contenu informationnel et la non redondance des informations qu'il véhicule. Pour la tâche du résumé mis-à-jour, une nouvelle contrainte s'ajoute : la détection de nouveauté par rapport aux anciennes informations. La plupart des travaux portant sur le résumé de mise à jour se sont fondés sur des systèmes de résumé multi-document en y ajoutant une extension pour la détection de nouveauté. (Wan, 2012) intègre cette notion à un système de RA fondé sur les graphes alors que (Delort and Alfonseca, 2012) l'intègre à un système à base de *topics models* hiérarchiques. (Li et al., 2012) ont appliqué le modèle hiérarchique de Dirichlet dans le cadre des approches bayésiennes hiérarchiques pour la tâche du résumé mis-à-jour.

Depuis quelques années, les approches abordant la problématique du RA comme un problème d'optimisation de contraintes fondée sur la programmation linéaire en nombres entiers (ILP) ont montré des résultats intéressants. Cette approche présente l'avantage d'optimiser conjointement plusieurs critères exprimés de façon très déclarative, ce qui en fait un modèle assez flexible. Les systèmes fondés sur l'ILP ont été appliqués au résumé dynamique (Gillick and Favre, 2009). (Li et al., 2015a) a appliqué ce modèle dans un cadre supervisé en prédisant les poids des concepts considérés. Néanmoins, la déclinaison la plus répandue de ce mo-

dèle, incarnée par (Gillick and Favre, 2009), ne prend en compte la détection de nouveauté dans les informations que de façon implicite et s'interdit de ce fait de bénéficier des travaux sur la paraphrase et l'implicature textuelle, problématique particulièrement importante dans le cadre du RA. Nous nous proposons, alors, d'étendre les travaux de (Gillick and Favre, 2009) dans le cadre du résumé de mise-à-jour en intégrant de façon plus explicite la contrainte de non redondance avec les anciennes informations.

Afin de juger si une information se présente dans un texte de façon redondante, nous nous proposons d'appliquer un clustering sémantique des phrases afin de regrouper les phrases similaires au sens de la paraphrase. Ce clustering sémantique est la combinaison d'un algorithme de clustering et d'une mesure de similarité sémantique. Pour évaluer la similarité des phrases, nous choisissons de représenter ces dernières par des vecteurs issus des plongements lexicaux, connus aussi sous le terme de *word embeddings*. C'est pourquoi nous commençons dans ce qui suit par présenter les différentes méthodes pour créer des représentations vectorielles de mots et les utiliser pour composer des vecteurs de phrases. Nous décrivons particulièrement trois méthodes de génération de *word embeddings*, celles que nous avons testées dans notre travail. Ensuite, nous présentons la méthode de calcul de la similarité de phrases à partir de leurs vecteurs et l'algorithme de clustering que nous avons choisi pour regrouper les phrases redondantes. Enfin, nous présentons notre méthode de résumé qui repose sur deux étapes principales. Nous partons de deux ensembles de documents A et B sachant que A précède chronologiquement l'ensemble B. En premier lieu, nous effectuons une segmentation en phrases de tous les documents des ensembles A et B. Ensuite, nous effectuons le clustering sémantique de ces phrases. Finalement, nous modifions le modèle ILP de base de façon à prendre en compte le clustering sémantique produit. La figure A.1 de l'annexe A donne une vue d'ensemble en termes systémiques de ce processus que nous détaillons dans ce qui suit.

En termes de résumé automatique et en particulier de résumé multi-document, le clustering de phrases a été largement utilisé. Généralement, ces approches regroupent des phrases faisant plus ou moins référence au même sujet et choisissent enfin une seule phrase pour représenter chaque cluster dans le résumé final (Zopf et al., 2016). Nous nous différencions de ces approches sur deux plans. Le premier

plan concerne le clustering. En effet, nous faisons la distinction entre clustering sémantique et clustering thématique de phrases. Le clustering sémantique utilise une similarité sémantique qui prédit à quel point deux textes ont le même sens (paraphrase) ou à quel point une phrase implique une autre (implicature textuelle). Le clustering thématique utilise une similarité thématique qui est une notion plus vague et mieux connue par le terme de proximité sémantique (Budanitsky and Hirst, 2006). Cette similarité thématique indique à quel point deux textes renvoient au même thème et ont tendance à être évoqués ensemble dans un discours sans pour autant signifier la même chose. Nous choisissons intentionnellement de réaliser un clustering sémantique regroupant les phrases qui expriment plus ou moins la même information, ce qui nous permet d'identifier les nouvelles phrases n'apportant pas de nouveauté du fait de la présence de phrases sémantiquement équivalentes dans les anciens documents. Le second plan sur lequel notre travail se différencie est la façon dont nous utilisons le clustering de phrases. Alors que la plupart des travaux en RA l'utilisent pour sélectionner une phrase par cluster et ainsi maximiser la couverture thématique (Radev et al., 2004, Wan and Yang, 2008, Bossard and De Neef, 2011, Deshpande and Lobo, 2013), nous utilisons le clustering sémantique comme un critère de détection de nouveauté informationnelle, parmi d'autres critères, dans un modèle de résumé qui maximise la couverture de concepts.

3.2 Représentation et similarité sémantique de phrases

La similarité sémantique de texte mesure à quel point deux entités textuelles sont équivalentes. Bien qu'il s'agisse d'un exercice assez naturel pour les êtres humains, l'estimation de la similarité sémantique est subjective et diffère d'une personne à l'autre. Si l'on demande à plusieurs personnes d'attribuer à une paire de phrases un score de similarité sur une échelle fermée, il y a de fortes chances que leurs réponses soient différentes même si des tendances globales se dégagent. C'est pour cette raison que pour constituer des données de référence, on demande à plusieurs annotateurs d'estimer les similarités des phrases pour garder ensuite

la moyenne des différentes réponses des annotateurs comme valeur finale.
Construire des algorithmes et des machines qui imitent les tâches humaines est souvent confronté au problème de la compréhension de texte, qui implique souvent de détecter des similarités sémantiques. Si on considère l'exemple des agents conversationnels, comprendre son interlocuteur ou lui répondre peut passer par la recherche de la question/réplique la plus similaire dans une base de données contenant les tâches déjà traitées.
De ce fait, la similarité sémantique de texte a connu récemment un intérêt particulier découlant du développement des technologies de traitement et d'analyse d'information. Plus particulièrement, les recherches en TAL prêtent de plus en plus attention à la sémantique distributionnelle, qui fonde le calcul sémantique sur une représentation vectorielle des mots en fonction de leurs contextes. Elle suppose que deux mots sémantiquement proches ont tendance à apparaître plus souvent dans des contextes similaires. Deux méthodes principales ont été développées pour mettre en œuvre cette approche. La première méthode est une méthode explicite consistant à faire le décompte des contextes. La seconde se fonde sur l'apprentissage automatique et consiste à entraîner un modèle à prédire un mot en fonction de son contexte ou inversement un contexte en fonction d'un mot.
L'intérêt accordé à la similarité sémantique de texte se concrétise notamment au travers des campagnes d'évaluation internationales SemEval et particulièrement de la tâche STS (Semantic Text Similarity), qui est présente dans SemEval depuis 2012. Cette campagne d'évaluation fournit chaque année des données de test pour évaluer la performance des systèmes de similarité sémantique de texte.
Dans le cadre de ce défi international, différentes méthodes de similarité sémantique ont démontré leur intérêt. Cependant, en pratique, l'utilisation de ces systèmes se heurte à un problème de coût de calcul élevé. Les mesures de similarité sémantique sophistiquées, qui exploitent plusieurs traitements linguistiques, sont généralement plus précises. Cependant, les traitements linguistiques utilisés alourdissent le coût de calcul de ces mesures, ce qui rend leur utilisation restreinte. Ce phénomène n'est en général pas pris en compte dans le cadre des campagnes d'évaluation sur la similarité sémantique. L'évaluation STS est ainsi une évaluation intrinsèque où le système candidat doit prédire la similarité sémantique de paires de phrases. Les données de SemEval 2014, par exemple, comptent 3750 paires de

phrases. Sur 3750 paires de phrases, le coût de calcul reste raisonnable quelle que soit la mesure de similarité utilisée, rapide ou lente. En revanche, dans le cas du résumé multi-document, calculer la matrice de similarité des phrases des documents en entrée peut impliquer de calculer la similarité de plusieurs millions de paires de phrases, même en se limitant à quelques dizaines de documents eux-mêmes longs de quelque centaines de mots. Ainsi, la lourdeur d'une mesure de similarité sophistiquée a nécessairement un impact important dans ce contexte. La mesure de similarité à base d'alignement de mots qui fut le système gagnant de SemEval 2014 (Sultan et al., 2014) est bon exemple de ce problème. Malgré la capacité de ce système à détecter la similarité sémantique des phrases, son utilisation dans le cadre de nos expérimentations a été presque impossible étant donné le temps de calcul énorme (>24h) nécessaire pour calculer un nombre assez important de similarités.

Nous avons finalement favorisé l'utilisation des mesures de similarité fondées sur les représentations vectorielles des mots de faible dimensionalité. Ces représentations permettent en effet d'obtenir un compromis intéressant entre leur niveau de performance, établi comme parmi les plus élevés dans un contexte non supervisé par des travaux tels que (Hill et al., 2016), et leur rapidité de manipulation. Cette représentation vectorielle des mots est connue sous le nom de *word embeddings* ou plongements lexicaux.

3.2.1 Que sont les *word embeddings* ?

Les *word embeddings* sont une représentation vectorielle des mots, dense et obtenue par apprentissage à partir d'un corpus selon l'approche prédictive évoquée ci-dessus. Cette représentation est construite de façon à ce que les mots sémantiquement similaires au sens distributionnel aient des vecteurs proches dans l'espace vectoriel. Chaque vecteur n'apporte pas en soi d'information par rapport au mot qu'il représente. Il permet, néanmoins, de déterminer la relation de ce mot avec les autres mots représentés dans le même espace vectoriel. Les word embeddings sont considérés comme un apport important en TAL pour plusieurs raisons. L'une de ces raisons est la réduction du nombre de dimensions des vecteurs de mots tout en conservant la sémantique véhiculée par les vecteurs. En effet, un des problèmes

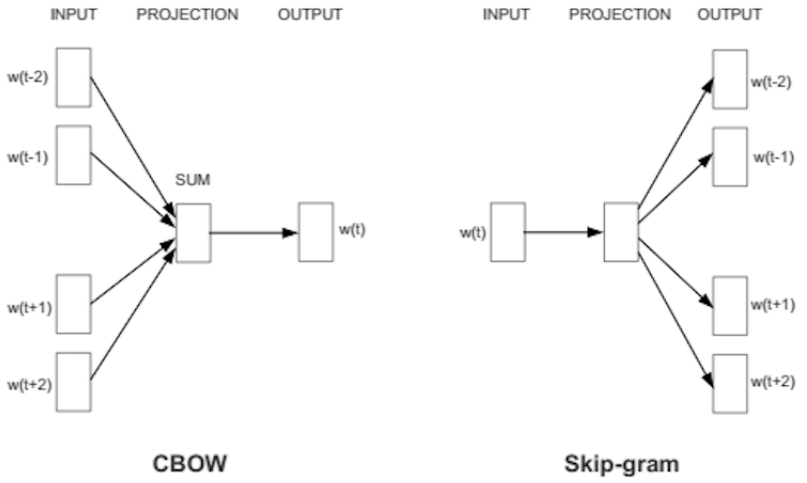

FIGURE 3.1 – Différence entre les architectures CBOW et Skip-Gram de Word2Vec

majeurs des calculs vectoriels en TAL est la structure creuse des données. Les modèles de type sac de mots représentent les mots, phrases ou document par un vecteur dont la taille est la taille du vocabulaire du document source notée T. Chaque composante du vecteur représente un mot. Par exemple, le vecteur d'une phrase est un vecteur de taille T contenant des valeurs non nulles dans les positions des mots qui forment cette phrase. Les valeurs non nulles peuvent être binaires ou encore plus informatives, comme par exemple, la fréquence du mot dans le document source. Quelle que soit la méthode utilisée, les modèles de type sac de mots produisent des représentations très creuses. L'un des autres points forts des *word embeddings* est qu'ils définissent la sémantique des mots en tenant compte de leurs contextes d'usage tout en gardant une faible dimensionalité des vecteurs. Nous détaillons ci-dessous deux méthodes populaires de création de *word embeddings* et une méthode d'enrichissement de ces représentations vectorielles que nous avons testées dans le cadre de notre approche et dont les résultats sont présentés dans le chapitre 4.

3.2.2 Le framework Word2Vec

Word2Vec est l'un des outils les plus populaires de génération de *word embeddings*, développé par Tomas Mikolov à Google en 2013 (Mikolov et al., 2013). Il s'agit d'un réseau de neurones ne comportant pas de couches cachées. Word2vec propose deux architectures possibles pour produire les embeddings de mots : CBOW (Continuous Bag Of Words) et le Continuous Skip-Gram. Comme le montre la figure 3.1, dans le modèle CBOW, le réseau de neurones prédit le mot actuel à partir d'une fenêtre de taille fixe centrée sur ce mot qui représente le contexte local de ce mot (généralement de 5 à 10 voisins). En revanche, dans le modèle Skip-Gram, c'est l'opération inverse qui a lieu : on prédit le contexte à partir du mot actuel. Alors que CBOW est plus rapide, le modèle Skip-Gram retourne des vecteurs de meilleure qualité pour les mots peu fréquents. Les deux modèles permettent de projeter les mots sur les axes paradigmatique et syntagmatique. En effet, le fait de considérer une fenêtre de voisins pour la génération des vecteurs mots permet de capturer non seulement la sémantique des mots (paradigmatique) mais aussi certaines propriétés syntaxiques (syntagmatique) comme la catégorie grammaticale. Par conséquent, dans la représentation vectorielle produite, les verbes seront plus proches des verbes que des noms par exemple.

3.2.3 L'algorithme GloVe

Comme Word2Vec, l'algorithme GloVe (Global Vectors for Word Representation) développé par Stanford produit des représentations vectorielles des mots en se référant à leurs contextes dans de larges corpus de texte (Pennington et al., 2014). La différence par rapport à Word2Vec est que Word2Vec est un modèle prédictif alors que GloVe est une approche à base de comptes. En premier lieu, une matrice de cooccurrences mots-mots est construite à partir de tout le corpus. Un élément de la matrice indique combien de fois le mot de la ligne est apparu dans le contexte du mot en colonne. Le contexte est défini comme l'ensemble des n mots précédant un mot et des n mots qui lui succèdent. Cette matrice de cooccurrences permet déjà d'obtenir des vecteurs de mots mais ces vecteurs présentent plusieurs points faibles. D'une part, la dimension des vecteurs dépend de la taille du vocabulaire et par conséquent, si l'on souhaite ajouter un mot au corpus, il fau-

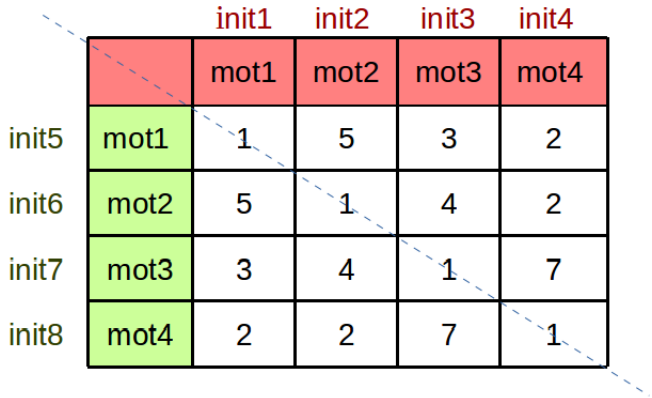

FIGURE 3.2 – Exemple de matrice de cooccurrence utlisée par GloVe

dra changer la dimension de tous les vecteurs. Une autre conséquence est la taille très importante de ces vecteurs ce qui nécessiterait une forte capacité de stockage, sans compter le fait qu'ils sont creux et donc moins adaptés à des applications en apprentissage automatique par exemple. À ce stade, certains travaux ont utilisé tout simplement la décomposition en valeurs singulières (SVD) pour réduire la dimension de la matrice, ce qui a permis d'obtenir de bons résultats (Landauer and Dumais, 1997, Rohde et al., 2006). GloVe reprend cette idée de factorisation de la matrice de cooccurrence entre mots mais permet son passage à l'échelle en ne faisant pas reposer cette factorisation sur une SVD, opération pouvant être trop coûteuse avec beaucoup de données. D'abord, les vecteurs des mots des lignes et des colonnes de la matrice sont initialisés aléatoirement. La figure 3.2 montre un exemple de cette matrice pour un ensemble de 4 mots. Pour chaque mot sont attribués deux vecteurs aléatoires, un pour son instance dans la première ligne et un pour son instance dans la première colonne. Ensuite, en se servant d'un modèle de régression log-bilinéaire, GloVe définit une fonction d'objectif aux moindres carrés qui vise à minimiser la différence entre le produit scalaire des vecteurs de deux mots et du logarithme de leur nombre de cooccurrences.

$$J = \sum_{i,j=1}^{V} f(X_{i,j})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - log X_{i,j})^2 \tag{3.1}$$

où w_i et b_i sont respectivement le vecteur et le biais du mot i, $\tilde{w}_j$ et $\tilde{b}_j$ sont respectivement le vecteur et le biais du mot j, $X_{i,j}$ est le nombre de fois où le

mot i apparaît dans le contexte du mot j et finalement f est une fonction de pondération qui évite d'apprendre seulement des mots très fréquents.
Comme chaque mot possède deux vecteurs différents à l'initialisation, (p. ex. *init1* et *init5* pour le mot *mot1*), l'algorithme GloVe produit deux ensembles différents de *word embeddings*.

		V1	V2	V3	V4
		mot1	mot2	mot3	mot4
V1'	mot1	0.22	0.75	1.12	0.34
V2'	mot2	0.51	1.81	1.75	0.25
V3'	mot3	0.46	0.18	1.61	1.08
V4'	mot4	1.35	0.06	0.35	0.68

FIGURE 3.3 – Exemple de matrice de cooccurrence obtenue par GloVe après l'exécution

En revenant à l'exemple précédent, l'algorithme produit ainsi les deux ensembles de vecteurs $\begin{bmatrix} V1 & V2 & V3 & V4 \end{bmatrix}$ et $\begin{bmatrix} V1' & V2' & V3' & V4' \end{bmatrix}$ comme le montre la figure 3.3. Les deux ensembles représentent des *word embeddings* aussi performants l'un que l'autre mais pour réduire l'effet du sur-apprentissage et le bruit, GloVe additionne ces deux ensembles pour constituer ses vecteurs.

3.2.4 Modification des word embeddings : Retrofitting

Un des reproches faits aux *word embeddings* est que leur production est purement statistique et ne tient pas compte des ressources linguistiques disponibles, comme les réseaux sémantiques WordNet et FrameNet ou encore la base de paraphrases PPDB. L'idée de (Faruqui et al., 2015) était d'enrichir les *embeddings* de mots par l'information relationnelle apportée par les réseaux sémantiques. Cet enrichissement consiste à faire en sorte que les mots ayant des liens dans les réseaux sémantiques aient des vecteurs plus proches. Le retrofitting s'effectue comme une étape de post-traitement, ce qui permet d'utiliser des vecteurs pré-entraînés par

n'importe quel modèle de production d'embeddings et ce, sans avoir besoin de modifier ce modèle. Le modèle de (Faruqui et al., 2015) représente le vocabulaire des embeddings couverts par l'ontologie par un graphe où les mots ayant des liens sémantiques dans cette ontologie sont connectés par des arcs. Les nœuds de ces graphes représentent les vecteurs modifiés, vus comme des variables latentes. À chaque nœud est associé un nœud représentant le vecteur original du mot en question, qui correspond à un observable. Ensuite, en utilisant un algorithme itératif de propagation de croyance, les nouveaux vecteurs sont modifiés de façon à minimiser la distance de chaque nœud à ses voisins et à son vecteur d'origine. (Mrkšić et al., 2016) se sont inspirés de la méthode de retroffiting et ont proposé le counter-fitting, qui comme son nom l'indique, prend en compte les relations de synonymie mais aussi d'antonymie dans l'enrichissement des *embeddings*. Le principe consiste à rapprocher les vecteurs des synonymes, selon la base de connaissances utilisée, et à éloigner les vecteurs des antonymes de la même façon. Dans le même cadre, d'autres chercheurs ont proposé une méthode pour combiner les *embeddings* de GloVe et Word2Vec et enrichir la combinaison par la base de paraphrases PPDB et le réseau sémantique ConceptNet (Speer and Chin, 2016). Nous avons utilisé les vecteurs produits par cette méthode dans notre modèle.

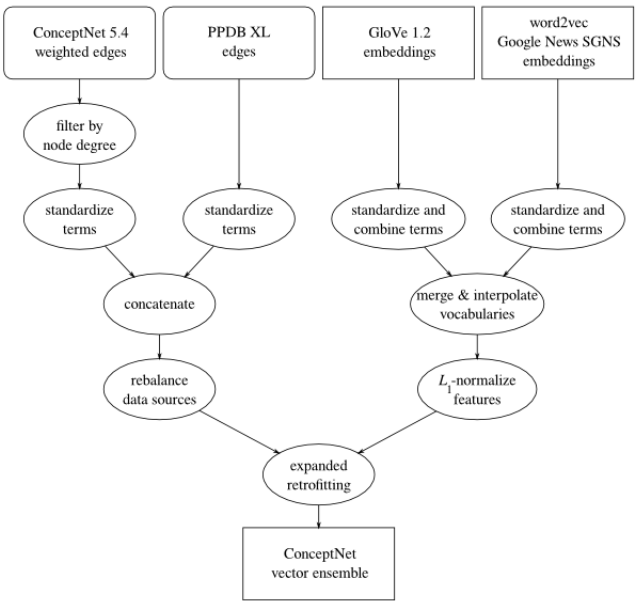

FIGURE 3.4 – Architecture de ConceptNet Vector Ensemble (Speer and Chin, 2016)

3.2.5 Calcul de la similarité de phrases à partir des *word embeddings*

Pour calculer la similarité entre phrases à partir des vecteurs de mots, la première idée qui vient à l'esprit est d'additionner les vecteurs des mots de chaque phrase, avec une possible normalisation, et ensuite de déduire une certaine différence ou distance entre les sommes calculées. Cette méthode, aussi simple qu'elle paraisse, fonctionne aussi bien que d'autres méthodes plus sophistiquées. L'objectif de toutes ces méthodes est de trouver une façon d'agréger des vecteurs de mots pour composer des vecteurs de phrases conservant la notion de distance sémantique. Selon (Le and Mikolov, 2014) la simple addition normalisée des vecteurs de mots est peu performante pour des tâches d'analyse de sentiments (*sentiment analysis*) parce qu'elle ne prend pas en compte l'ordre des mots. Toujours selon la même source, cette méthode est incapable de détecter des phénomènes linguistiques complexes comme le sarcasme. Néanmoins, pour la tâche de similarité sémantique de textes courts, la majorité des travaux se focalisant sur la

représentation des phrases pour la similarité sémantique admettent que le fait de calculer la moyenne des vecteurs des mots composant une phrase pour calculer la similarité des vecteurs de phrases constitue une baseline solide, qui atteint les performances des modèles élaborés (Le and Mikolov, 2014, Kenter et al., 2016, Das et al., 2016, Gershman and Tenenbaum, 2015) et qui les surpassent dans certaines tâches comme la classification sémantique (White et al., 2015). Des travaux ont comparé différentes méthodes de représentation de phrases et entre autres la méthode des moyennes des vecteurs de mots, dans le cadre d'une tâche de similarité sémantique (Hill et al., 2016). Sur les données de SemEval 2014, cette méthode a surpassé un grand nombre de systèmes de représentation de phrases plus ou moins élaborés, en particulier :

— ParagraphVector (Le and Mikolov, 2014), un algorithme qui représente des éléments textuels de longueurs différentes -phrases, paragraphes ou documents- par des vecteurs de taille fixe en entraînant le modèle à prédire les mots constituant une phrase donnée.
— SkipThought (Kiros et al., 2015) est un modèle qui, étant donné une phrase, vise à prédire la phrase qui la précède et la phrase qui la succède. La phrase source est encodée dans un réseau de neurones récurrent et ensuite décodée pour donner les deux phrases voisines.
— FastSent (Hill et al., 2016) est un modèle qui entraîne des vecteurs de phrases en prédisant pour chaque phrase source, les mots des phrases avoisinantes.

3.3 Clustering sémantique

Comme pour le résumé multi-document, le résumé dynamique a pour objectif d'optimiser le compromis entre l'information pertinente, la redondance informationnelle et la taille du résumé final. La contrainte supplémentaire est de détecter la nouveauté dans l'ensemble des documents récents afin d'éviter au lecteur de lire ce dont il a déjà pris connaissance. Puisque la détection de nouveauté dans la tâche du résumé dynamique revient à pénaliser les anciennes informations réapparues dans les nouveaux documents, nous pouvons commencer par identifier les

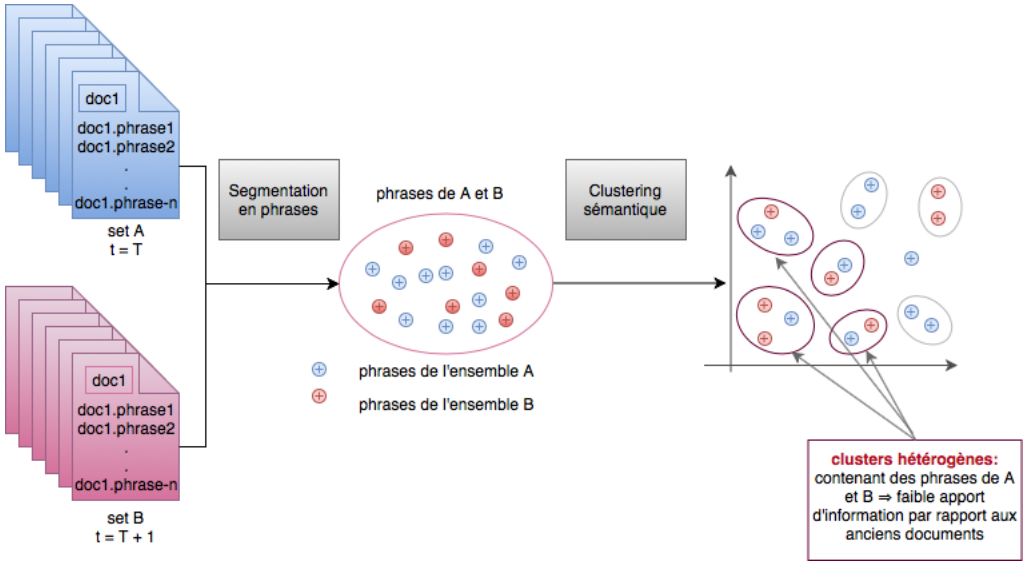

FIGURE 3.5 – Principe du clustering des phrases pour la détection de nouveauté

phrases de l'ensemble B qui sont équivalentes à des phrases de l'ensemble A. Une façon d'atteindre cet objectif est d'effectuer un clustering sémantique de toutes les phrases en entrée indépendamment de leur source (A ou B) comme le montre la figure 3.5. Un tel clustering permettrait de regrouper les phrases véhiculant plus ou moins la même information sans pour autant être exprimée de la même façon. L'intérêt de cette étape va au-delà de la détection de la redondance informationnelle à travers le temps. En effet, il s'agit aussi d'une étape de filtrage de données permettant de réduire le coût de sélection des phrases du résumé final par le modèle ILP, en réduisant le nombre de solutions possibles du problème. En outre, bien que le modèle ILP fonctionne au niveau des concepts (ngrammes de mots), le fait de considérer la similarité sémantique au niveau des phrases permet de réduire encore plus le coût de calcul des similarités étant donné que le nombre de paires de phrases est beaucoup plus faible que le nombre de paires de ngrammes de mots (p. ex. paires de mots).

Méthode de clustering : Clustering de Markov

Pour réaliser notre clustering sémantique, nous avons besoin d'un algorithme de clustering qui considère la similarité sémantique de phrases comme critère pre-

mier de regroupement. Notre objectif est d'obtenir de petits clusters de phrases contenant des phrases exprimant la même information. Nous n'avons donc aucune condition sur le nombre de clusters à produire qui dépend étroitement de la diversité des données en entrée et du nombre de documents à regrouper. Par conséquent, les algorithmes de partitionnement de type *k-means*, qui requièrent de fixer le nombre de clusters *a priori*, ne conviennent pas à notre besoin. Au contraire, fixer un seuil de similarité minimale pour évaluer l'équivalence entre les phrases est plus cohérent avec notre besoin. En effet, le seuil de similarité est un facteur dépendant seulement de la nature de la mesure de similarité sémantique et de l'interprétation des scores qu'elle produit. Selon l'objectif voulu, le seuil de similarité peut être calibré sur un ensemble de développement. De plus, étant donné que la mesure de similarité sémantique choisie est très rapide, le choix d'un algorithme nécessitant de pré-calculer la matrice des similarités mutuelles de toutes les phrases est un choix possible et réalisable. À la lumière de ces critères, nous avons choisi d'utiliser l'algorithme de Clustering de Markov (MCL) (van Dongen, 2000). L'algorithme MCL est un algorithme de regroupement pour les graphes fondé sur la simulation de flots stochastiques dans les graphes. Il suppose qu'une marche aléatoire visitant un cluster dense va plus probablement visiter tous les nœuds de ce cluster avant de pouvoir le quitter, tout simplement parce que les connexions intra-cluster sont beaucoup plus nombreuses que les connexions avec des éléments hors de ce cluster. Ceci revient à trouver les ensembles de nœuds fortement interconnectés. En lançant un grand nombre de marches aléatoires dans un graphe, il est possible de repérer où le flux tend à s'agglomérer, et par conséquent, où les clusters se situent. Pour le clustering de phrases, nous considérons que les nœuds

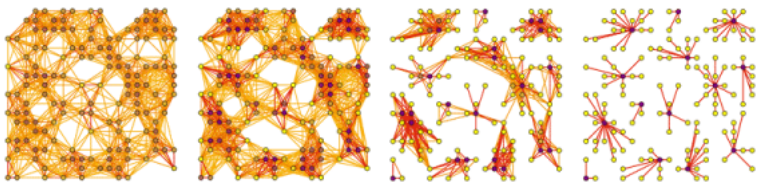

FIGURE 3.6 – Exemple d'évolution de MCL dans un graphe
source : https ://micans.org/mcl/

du graphe représentent les phrases et que les arcs représentent les similarités inter-

phrases. L'algorithme MCL commence par convertir le graphe en une matrice de transition T où chaque élément de la matrice $t_{i,j}$ représente la probabilité qu'une marche aléatoire au départ de i atteigne le nœud j. Ensuite, l'algorithme alterne successivement deux opérations sur la matrice de probabilités : l'expansion et l'inflation. L'expansion consiste à élever au carré (ou une autre puissance) la matrice de transition, ce qui conduit à effectuer des marches aléatoires plus longues. Ceci aide le flux à se disperser à l'intérieur des clusters. L'inflation consiste à calculer la puissance d'Hadamard de la matrice. En pratique, elle renforce les probabilités des marches intra-cluster et affaiblit celles des marches inter-cluster. Le facteur d'inflation I permet de déterminer la granularité des clusters finaux : plus le facteur d'inflation est important et plus la granularité des clusters est faible (clusters plus petits). Les opérations d'expansion et d'inflation sont alternées successivement jusqu'à convergence de l'algorithme.

3.4 Sélection de phrases pour le résumé mis-à-jour

Comme modèle ILP de base, nous avons adopté le modèle ICSISumm (Gillick et al., 2009), qui a été conçu initialement pour le résumé multi-document (Gillick and Favre, 2009) puis adapté à la tâche dynamique. Le modèle ICSISumm permet d'extraire les phrases du résumé final en opérant au niveau des concepts. Pour des raisons de simplicité, les concepts considérés sont les bigrammes de mots des documents d'origine. L'ensemble des bigrammes est constitué de toutes les paires de mots qui apparaissent consécutivement dans les textes source à condition que les mots du bigramme ne soient pas tous les deux des mots vides.

3.4.1 Formalisation du problème

Nous commençons d'abord par expliciter le modèle ICSISumm conçu pour la tâche du résumé multi-document. L'objectif de ICSISumm est de maximiser la couverture en bigrammes de mots du résumé final par rapport aux documents source dans la limite d'une taille maximale du résumé. Le score du résumé à maximiser est la somme des poids des bigrammes qu'il inclut, chaque bigramme y contribuant par son poids une seule fois, ce qui assure implicitement une forme de

non-redondance pour l'aspect multi-document du résumé. Dans le cadre du résumé dynamique, où l'on souhaite résumer une collection de documents B en supposant que le lecteur final a déjà lu une collection de documents antérieurs notée A, cette forme de non redondance garantit que les informations sélectionnées de la collection B ne sont pas redondantes avec le contenu de A. Ce modèle est formalisé comme suit :

$$\begin{aligned} \text{maximiser} : & \sum_i w_i.c_i \\ \text{avec} : & \sum_j s_j.l_j \leq L \qquad (3.2) \\ & s_j.Occ_{ij} \leq c_i, \forall i,j \qquad (3.3) \\ & \sum_j s_j.Occ_{ij} \geq c_i, \forall i \qquad (3.4) \\ & c_i \in \{0,1\}\, \forall i \text{ et } s_j \in \{0,1\}\, \forall j \end{aligned}$$

La variable c_i est une variable binaire indiquant la présence, ou non, du concept i dans le résumé. w_i est le poids du concept i, égal au nombre de documents où le bigramme apparaît au moins une fois. La longueur de la phrase j est notée l_j et la longueur maximale du résumé est la constante L. La variable binaire s_j indique la présence de la phrase j dans le résumé et Occ_{ij} indique l'occurrence du concept i dans la phrase j. La contrainte (3.2) garantit le non dépassement de la taille maximale du résumé. Les contraintes (3.3) et (3.4) garantissent la cohérence du système en opérant une correspondance entre la sélection des bigrammes de mots et la sélection des phrases. En effet, le système essaie d'optimiser la sélection des bigrammes de mots mais, finalement, il sélectionne des phrases et plus particulièrement les phrases qui contiennent les bigrammes qui satisfont le problème ILP. C'est pourquoi les deux contraintes (3.3) et (3.4) ont été mises en place. La contrainte (3.3) assure que si une phrase est sélectionnée alors tous les concepts qu'elle contient doivent être sélectionnés. Quant à la contrainte (3.4), elle garantit que la sélection d'un concept nécessite qu'au moins une phrase qui le contient soit sélectionnée.

Pour résumer, ICSISumm vise à sélectionner des phrases aussi différentes que possible et qui contiennent chacune l'information qu'on retrouve dans autant de

documents que possible.

3.4.2 ICSISumm pour le résumé mis-à-jour

Pour adapter le modèle ICSISumm à la tâche de résumé mis-à-jour, Gillick et al. (2009) sont partis de l'hypothèse que dans le domaine journalistique, les nouvelles informations sont généralement mentionnées au début de l'article. Cette hypothèse est liée à la façon dont sont rédigés les articles, qui constituent la plupart des données d'évaluation. En se fondant sur ce présupposé, sont ainsi favorisées les phrases apparaissant en première position de chaque document de l'ensemble B par rapport au reste des phrases. Pour ce faire, les bigrammes appartenant aux premières phrases de chaque document sont surpondérés en multipliant leur poids d'origine. Il faut noter que même pour la version dédiée au résumé multi-document de base, les bigrammes des premières phrases sont favorisés mais cette prime est plus prononcée pour le résumé mis-à-jour.

3.4.3 Prise en compte du clustering sémantique

Le clustering sémantique réalisé précédemment nous fournit des regroupements de phrases équivalentes issues des anciens et des nouveaux documents. Notre idée consiste à intégrer les résultats de ce clustering dans le modèle ILP décrit précédemment de façon à pénaliser les phrases trop similaires aux phrases issues des anciens documents. Nous avons pour cela choisi d'agir sur les poids des bigrammes puisqu'il a été démontré que, dans les systèmes de résumé fondés sur la maximisation de concepts, modifier les poids des concepts est la façon qui influe le plus sur la performance du système en comparaison avec d'autres méthodes (i.e. ajouter des contraintes, agir sur la sélection de phrases, etc.) (Li et al., 2015a). Comme notre objectif consiste à réduire la redondance par rapport aux anciennes informations, nous avons choisi de pénaliser les bigrammes des phrases contenues dans des clusters regroupant des phrases issues des anciens et nouveaux documents, que nous appelons dorénavant clusters mixtes. Si un bigramme est présent dans une phrase contenue elle même dans un cluster mixte, alors son poids est pénalisé par un facteur α de la façon suivante : $w'_i = \frac{w_i}{\alpha}$, où w'_i est le nouveau poids du bigramme c_i. Le paramètre α est déterminé expérimentalement sur un ensemble

de développement. Nous notons que tous les bigrammes sont remplacés par leurs radicaux (leurs mots racines) pour que le calcul de leur poids soit plus représentatif et plus pertinent. En effet, les différentes formes d'un même mot doivent être unifiées pour éviter que les bigrammes importants ayant des formes différentes aient le même poids faible que des bigrammes peu fréquents. Par exemple, si les bigrammes *found_ device*, *found_ devices* et *few_ minutes* apparaîssent tous dans 3 documents, alors les 3 bigrammes auront le même poids et donc la même importance. Au contraire, si on considère que *found_ device* et *found_ devices* représentent le même bigramme racine *found_ devic*, et s'ils apparaissent dans des documents différents, alors le bigramme racine aura un poids (6) deux fois plus élevé que celui de *few_ minut* et reflètera mieux la pertinence des bigrammes selon le texte. De plus, comme dans (Gillick et al., 2009), les bigrammes dont les poids sont inférieurs à un seuil d'élagage donné ne font pas partie de l'ensemble des bigrammes que l'on essaie de couvrir. En outre, comme il a été démontré que ce filtrage peut avoir un effet négatif s'il est trop violent (Boudin et al., 2015), nous avons choisi d'effectuer l'étape de pénalisation après la phase de filtrage pour ne pas filtrer des bigrammes dont le poids pénalisé se retrouve en dessous du seuil d'élagage.

3.5 Conclusion

Dans ce chapitre, nous avons présenté la famille de méthodes dans laquelle nous nous inscrivons : méthodes fondées sur l'ILP et un critère de maximisation de la couverture en bigrammes de mots. Nous avons ensuite présenté la méthode que nous avons définie pour prendre en compte de façon explicite la redondance entre phrases. Cette méthode est fondée sur l'association d'une mesure de similarité sémantique entre phrases et d'un algorithme de clustering s'appuyant sur une matrice de similarité. Nous avons plus spécifiquement détaillé ces deux dimensions en insistant tout particulièrement sur la problématique de la représentation sémantique des phrases dans le contexte de l'utilisation de plongements lexicaux. Finalement, nous avons montré comment nous avons intégré l'information issue du clustering sémantique dans le modèle de résumé par optimisation. Le chapitre suivant présente les expérimentations menées pour évaluer et comparer les différentes

versions du système tout en analysant les résultats obtenus.

Chapitre 4

Évaluation de l'intégration de la similarité sémantique pour le RA

4.1 Introduction

Dans ce chapitre nous allons, dans un premier temps, décrire le cadre d'évaluation mis en place pour évaluer l'efficacité de l'exploitation de la similarité sémantique de phrases pour le résumé mis-à-jour. Ensuite nous présenterons les résultats obtenus en les comparant à des baselines de référence. Finalement, nous analyserons en détail les résultats obtenus dans le but de comprendre l'origine de l'amélioration et les limites de notre approche. Nous avons aussi pris du recul par rapport à notre contexte précis en évaluant la limite supérieure des méthodes extractives afin de mieux situer la performance de notre méthode par rapport à la performance maximale atteignable. Ceci permettra aussi de vérifier si il existe encore une marge d'amélioration dans la famille des méthodes par extraction.

4.2 Cadre d'évaluation

Dans cette section, nous introduisons les données d'évaluation utilisées ainsi que la méthode d'évaluation choisie.

4.2.1 Méthode d'évaluation

Comme métrique d'évaluation nous avons choisi d'utiliser la mesure ROUGE décrite en 2.6.1 qui évalue les résumés en termes de contenu informationnel. D'une part, il s'agit d'une méthode automatique dès lors que l'on dispose de résumés manuels de référence. D'autre part, c'est la mesure d'évaluation la plus adoptée

par les travaux de l'état de l'art pour sa très haute corrélation avec les jugements humains du point de vue du contenu (Lin, 2004). Utiliser la même métrique permet de se situer par rapport aux travaux de l'état de l'art. En revanche, elle ne prend pas en compte la qualité linguistique du résumé et notamment l'ordre des mots. Le rappel de ROUGE d'un résumé est calculé selon l'équation suivante.

$$Rappel(Res) = \frac{\sum_{r \in Ref} \sum_{Ngrammes \in Res} card_{match(r,Res)}(Ngrammes)}{\sum_{r \in Ref} \sum_{Ngrammes \in r} card(Ngrammes)} \quad (4.1)$$

où Ref est l'ensemble des résumés manuels auxquels on se compare, Res est le résumé à évaluer et $card_{match(r,Res)}(Ngrammes)$ est le nombre de n-grammes communs entre le résumé à évaluer Res et le résumé manuel r. Quant à la précision, elle est calculé comme suit :

$$Precision(Res) = \frac{\sum_{r \in Ref} \sum_{Ngrammes \in Res} card_{match(r,Res)}(Ngrammes)}{\sum_{r \in Res} \sum_{Ngrammes \in r} card(Ngrammes)} \quad (4.2)$$

La F-mesure de ROUGE est la combinaison de la précision et du rappel. Elle est exprimé comme suit :

$$Fmesure = \frac{(1+\beta)^2 Rappel \cdot Precision}{Rappel + \beta^2 Precision} \quad (4.3)$$

où β est un paramètre permettant le cas échéant de donner une importance différenciée à la précision et au rappel. Dans le cas de ROUGE, les deux mesures ont la même importance et $= 1$.

4.2.2 Données d'évaluation

Les données d'évaluation les plus utilisées dans le domaine du résumé automatique et constituant le standard en termes d'évaluation sont issues d'une campagne d'évaluation internationale qui s'intéresse aux technologies du TAL. Appelée DUC pour *Document Understanding Conference* depuis l'année 2000, puis renommée TAC pour *Text Analysis Conference* depuis 2008, cette campagne d'évaluation organise chaque année une série d'ateliers fournissant l'infrastructure pour une évaluation à large échelle d'un certain nombre de tâches du TAL et particulièrement le résumé automatique. L'objectif de cette conférence est de mettre l'accent sur l'avancement de l'état de l'art dans le domaine visé au travers de résultats

	DUC 2007	TAC 2008	TAC 2009
nombre de collections de documents	10	48	44
nombre de documents/collection (A)	10	10	10
nombre de documents/collection (B)	10	10	10
nombre moyen de phrases/collection	173	253	254
nombre moyen de mots/collection	3478	5460	5635
Taux de Compression [1]	2,87%	1,83%	1,77%

TABLE 4.1 – Caractéristiques des données d'évaluation

d'évaluation. Elle offre pour ce faire un cadre d'évaluation commun à la communauté des chercheurs du TAL, ce qui facilite une comparaison équitable entre les systèmes. Plusieurs variantes de la tâche de résumé automatique ont été ciblées par cette campagne d'évaluation comme le résumé mono-document et multi-document générique, le résumé guidé par la requête, le résumé d'opinions issues de blogs, le résumé de documents bio-médicaux guidé par les citations et finalement le résumé de mise à jour. Ce dernier a fait partie des éditions de 2007, 2008, 2009 et 2011. Pour évaluer notre approche, nous avons utilisé les données des campagnes DUC 2007, TAC 2008 et TAC 2009. Le tableau 4.1 décrit la structure de ces quatre ensembles de données présentée par collection de documents. Une collection de documents est un cluster regroupant des documents thématiquement homogènes qui représentent les documents à résumer.

4.2.3 Étalonnage des paramètres

Bien que notre méthode soit non supervisée, nous avons un certain nombre de paramètres à fixer. Ces paramètres sont :

- le seuil d'élagage des bigrammes au dessous duquel le bigramme n'est pas pris en compte dans la fonction d'objectif ;
- le seuil de similarité minimal au dessous duquel nous considérons que la similarité des phrases est nulle ;
- le facteur de pénalisation des bigrammes appartenant à des phrases similaires aux phrases des anciens documents.

Afin de trouver la combinaison des valeurs optimales de ces paramètres, nous

utilisons la méthode du *Grid Search*. Cette méthode consiste à croiser des séries de valeurs possibles d'un ou de plusieurs paramètres et à tester la performance du système avec chaque combinaison. Ensuite, il suffit de comparer les performances avec chaque croisement afin de choisir le meilleur paramétrage du système. Dans notre cas, les grilles utilisées sont :

- seuil d'élagage : 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 ;
- seuil de similarité : 0,3 , 0,35 , ... , 0,9 , 0,95 , 0,96 , 0,97 , 0,98 , 0,99 ;
- facteur de pénalisation : 1,5 , 2 , 2,5 , 3 , 3,5 , 4 , 4,5 , 5.

Pour une évaluation correcte, il convient de découper les données d'évaluation en deux ensembles : un ensemble de développement sur lequel on applique le *Grid Search* pour déterminer les valeurs optimales des paramètres et un ensemble de test sur lequel on teste la performance du système avec les paramètres choisis. Dans notre cas, pour évaluer chacun des jeux de tests, DUC 2007, TAC 2008 et TAC 2009, nous utilisons les deux autres jeux de test comme ensembles de développement.

4.3 Limite supérieure des systèmes extractifs

Dans cette section, nous nous intéressons à la limite de performance supérieure des méthodes de résumé extractives. D'une part, ceci nous permet de vérifier qu'il y a encore un espace pour l'amélioration des méthodes par extraction. D'autre part, la limite supérieure constitue une référence par rapport à laquelle nous pourrons nous comparer.

4.3.1 Génération des résumés Oracle

Les résumés de référence utilisés dans les campagnes d'évaluation évoquées ci-dessus sont produits directement par des humains, sans la contrainte de reprendre des phrases des documents source. Par conséquent, ces résumés ne représentent pas une référence directe en termes de résumé extractif. Afin de constituer des références extractives, nous procédons à la génération automatique de résumés Oracle.

Un système Oracle de résumé est un système supposé produire des résumés idéaux

autant que possible selon une méthode d'évaluation bien déterminée. Dans notre cadre, un résumé Oracle est défini comme l'ensemble des phrases respectant la taille limite du résumé et maximisant un score ROUGE-N. Pour générer les résumés Oracle, nous nous servons des résumés de référence utilisés pour l'évaluation de notre système. L'objectif est de trouver un résumé des textes source (à résumer) aussi proche que possible des résumés de référence. La distance ici est déterminée par le score ROUGE-2, qui est le score que nous utiliserons pour évaluer notre approche. De cette façon, nous obtiendrons le meilleur résumé par extraction du point de vue de ROUGE. Pour ce faire, nous suivons une méthode gloutonne simple qui construit les résumés de façon incrémentale. Premièrement, nous commençons par sélectionner la phrase qui maximise le score ROUGE-2 parmi les textes en entrée. Ensuite, à chaque itération, nous sélectionnons la phrase qui maximise le score ROUGE-2 du résumé intermédiaire formé par les phrases déjà sélectionnées et la phrase en cours. Nous répétons l'opération jusqu'à ce que la taille maximale du résumé soit atteinte. Cette approche permet de fournir la limite supérieure par extraction d'après ROUGE-2 et ne correspond donc pas au meilleur Oracle possible. De plus, nous avons utilisé une méthode gloutonne simple alors que des méthodes d'optimisation globale fondée sur l'ILP sont susceptibles a priori de donner de meilleurs résultats (Schluter, 2017, Hirao et al., 2017). Cependant, (Schluter, 2017) a montré que la méthode gloutonne est aussi performante que la méthode exacte globale.

4.3.2 Évaluation des résumés Oracle

Nous avons généré les résumés Oracle pour les jeux d'évaluation TAC 2008 et TAC 2009 décrits précédemment à la section 4.2.2.

Jeux de données	ROUGE-1		ROUGE-2		ROUGE-L		ROUGE-SU4	
	R	F	R	F	R	F	R	F
TAC08	36,23	41,50	16,06	18,39	31,40	35,95	16,94	19,45
TAC09	36,87	41,86	16,19	18,41	22,66	25,73	17,68	20,14

TABLE 4.2 – Évaluation des résumés Oracle

La première observation que suscite l'évaluation de ces résumés Oracle rapportée par le tableau 4.5 est qu'il reste encore une marge d'amélioration pour les méthodes extractives. Par exemple, pour TAC 2008 la F-mesure de ROUGE-2 est à 18,39, ce qui est loin des meilleures performances des systèmes de l'état de l'art (en moyenne entre 10 et 12 cf. section 2.7.2). Cependant, d'un autre point de vue, ces résultats montrent que les scores ROUGE-2 des résumés idéaux n'atteignent même pas 20%, un score très loin de 100%. Ceci devrait être pris en compte dans l'analyse des performances des systèmes de résumé (extractif ou abstractif) en utilisant ROUGE. D'autre part, on peut expliquer la faiblesse des scores des meilleurs résumés extractifs par deux facteurs. Premièrement, la mesure d'évaluation ROUGE compare le résumé système aux résumés de référence au niveau lexical sans mesurer la proximité sémantique des deux résumés. Nous rappelons que les résumés de référence sont abstractifs et leur style d'écriture diffère d'un annotateur à l'autre. Par conséquent, en générant un résumé extractif ou abstractif, rien ne garantit qu'on reproduira les mêmes mots qu'un ou plusieurs résumés de référence. Deuxièmement, les données d'évaluation utilisées fournissent 4 résumés de référence par échantillon. Ceci permet d'obtenir une meilleure évaluation dans le sens où ROUGE compare les systèmes à la performance humaine alors que même les humains peuvent résumer de manière différente. ROUGE calcule le chevauchement entre le résumé système et chacun des résumés humains et retourne la moyenne. On pourrait aussi avoir le score correspondant au chevauchement maximal mais c'est la moyenne qui est utilisée dans les campagnes d'évaluation et dans la littérature pour garantir une meilleure stabilité des résultats. Une conséquence directe de l'utilisation de la moyenne est que le résumé idéal dont le score est de 100% n'existe pas : on ne peut pas produire un résumé de 100 mots qui couvre le vocabulaire de 4 résumés différents de 100 mots chacun qui est forcément supérieur à 100 mots et inférieur au total de 400 mots. Même chacun des résumés de référence ne pourrait pas atteindre la performance maximale.

4.4 Systèmes évalués

4.4.1 Baselines

- ISumm 2009. C'est le système ICSISumm que nous avons décrit à la section 3.4 et à partir duquel nous avons implémenté notre contribution. Nous utilisons ici la version n'incluant pas la compression de phrases. Il faut noter aussi que ICSISumm a été classé en tant que meilleur système dans l'étude comparative de Hong et al. (2014).
- ISumm-BG-DOWN-1. Cette baseline est une adaptation de ISumm 2009 où nous pénalisons les poids des bigrammes présents dans l'ancien ensemble de documents (A).
- ISumm-BG-DOWN-2. Dans cette version modifiée de ISumm 2009, nous pénalisons les poids des bigrammes dont la fréquence dans l'ensemble (A) est plus grande que leur fréquence dans l'ensemble des nouveaux documents (B).

Nous avons mis en place les deux dernières baselines, qui n'incluent pas le clustering de phrases, afin de vérifier l'apport et l'intérêt du clustering.

4.4.2 Systèmes de l'état de l'art

Nous avons retenu comme référence les systèmes de l'état de l'art obtenant les meilleurs résultats sur les jeux de test que nous considérons selon les expérimentations menées par Hong et al. (2014).

- Topic Modeling (Wang and Zhou, 2012). Ce système utilise les distributions de probabilité des thèmes du texte pour la détermination de la saillance et une approche de modélisation dynamique pour le contrôle de la redondance.
- CorrRank (Lei and Yanxiang, 2010). Cet algorithme sélectionne les phrases en fonction d'un modèle d'évolution des thèmes.
- Supervised ILP (Li et al., 2015a). Ce système prédit les poids des bigrammes par le biais d'un modèle supervisé utilisant des critères de saillance et de nouveauté au niveau de la phrase et du bigramme de mots. Les phrases du résumé sont ensuite sélectionnées par un modèle ILP et un modèle de régression est appliqué pour le ré-ordonnancement des phrases.

4.4.3 Systèmes proposés

Nous présentons les résultats de notre contribution en utilisant différents *word embeddings* pré-entraînés pour calculer les similarités entre phrases. Tous les seuils de similarité sont supérieurs à 0,95, ce qui garantit une précision de la mesure de similarité au moins égale aux précisions rapportées au tableau 4.5.

- MCL-W2V-ISumm. Dans cette version, nous utilisons des vecteurs pré-entraînés par l'outil Word2Vec (Mikolov et al., 2013), décrit à la section 3.2.2, sur le corpus Google News. Ce corpus comporte 100 milliards de mots. Les vecteurs utilisés ont été entraînés en utilisant la version CBOW de Word2Vec avec une fenêtre glissante de taille égale à 5. Les *word embeddings* obtenus comptent 3 millions de vecteurs de 300 dimensions.
- MCL-GLOVE-ISumm. Dans cette configuration, nous avons utilisé des *word embeddings* couvrant 2,2 millions de mots entraînés avec le modèle GloVe (Pennington et al., 2014), décrit à la section 3.2.3, et comportant eux aussi 300 dimensions. Le modèle a été entraîné sur le corpus Common Crawl couvrant 840 milliards de mots.
- MCL-CNet-ISumm. Cette version calcule les similarités sémantiques en se servant des *embeddings* du ConceptNet Vector Ensemble (Speer and Chin, 2016). Il s'agit d'une combinaison *a posteriori* des vecteurs produits par GloVe et Word2Vec. Cette fusion de *word embeddings* est enrichie par la connaissance que renferme le réseau sémantique ConceptNet ainsi que la base de paraphrases PPDB. Cette méthode d'enrichissement de *word embeddings* est décrite à la section 3.2.4

4.5 Résultats et analyse

Le tableau 4.3 montre que pour tous les *word embeddings* utilisés, notre système surpasse tous les systèmes de référence. L'amélioration est observée également pour les trois mesures de ROUGE utilisées. L'amélioration par rapport à ISumm 2009, qui possède la même configuration que notre système, confirme l'intérêt de la gestion explicite de la redondance pour le résumé mis-à-jour. D'autre part, l'amélioration par rapport à ISumm-BG-DOWN-1&2 montre que des méthodes basiques

System/dataset	DUC 2007			TAC 2008			TAC 2009		
	R1	R2	RSU4	R1	R2	RSU4	R1	R2	RSU4
Baselines									
ISumm 2009	33,73	7,59	11,23	38,28	11,19	14,46	37,40	10,37	13,86
ISumm-BG-DOWN-1	34,46	7,91	11,74	36,99	10,15	13,66	37,39	10,25	13,87
ISumm-BG-DOWN-2	33,71	7,55	11,22	38,02	11,05	14,18	37,27	9,91	13,62
Systèmes de l'état de l'art									
Supervised ILP	-	-	-	-	9,99	13,61	-	9,61	13,77
Topic Modeling	-	-	-	36,73	10,41	13,79	36,42	9,58	13,53
CorrRank	-	-	-	36,71	9,70	13,19	36,87	9,73	13,59
Systèmes proposés									
MCL-W2V-ISumm	34,99	8,14	11,79	38,52	11,49	14,68	37,50	10,48	13,98
MCL-GLOVE-ISumm	**36,08**	**9,46**	**12,96**	**38,62**	**11,57**	**14,75**	**37,53**	**10,60**	**14,08**
MCL-CNet-ISumm	35,23	8,30	11,98	38,28	11,21	14,49	**37,53**	10,38	13,91

TABLE 4.3 – Rappels moyens du score ROUGE sur les données de DUC 2007, TAC 2008 et TAC 2009

de minimisation de la redondance ne sont pas performantes dans le cadre des modèles ILP, contrairement à notre approche fondée sur le clustering sémantique des phrases. Notre deuxième version utilisant les vecteurs de mots GloVe obtient des scores ROUGE plus importants que ceux obtenus en utilisant les vecteurs Word2Vec ou les vecteurs ConceptNet Ensemble. Nous étudions plus bas (§4.5.2) les caractéristiques des différents *embeddings* utilisés pour comprendre pourquoi un ensemble de vecteurs se comporte mieux qu'un autre. Nous commençons pour le moment (§4.5.1) par étudier l'influence des paramètres sur la performance des méthodes proposées.

4.5.1 Influence des paramètres

Nous étudions, dans cette section, l'influence des différentes valeurs des paramètres sur le comportement de notre approche.

Influence du seuil de similarité

L'objectif de notre clustering de phrases est de mettre l'accent sur la proximité sémantique des phrases et non sur leur proximité thématique. Nous cherchons donc à fixer une valeur de seuil pour notre mesure de similarité en relation avec cet objectif nous permettant ainsi d'obtenir des clusters étroits contenant des phrases très similaires au sens de la paraphrase. A priori, on peut penser que cet objectif a d'autant plus de chances d'être atteint que le seuil de similarité minimale ε est élevé. Nous testons cependant des valeurs faibles et élevées afin de découvrir l'impact du seuil de similarité sur la performance du système. Nous rappelons que, une fois la valeur optimale de ε trouvée, nous calculons les similarités des paires de phrases comme suit :

1. Calcul de $sim(s_1, s_2)$ où $sim()$ est la mesure de similarité
2. Si $sim(s_1, s_2) < \varepsilon$ alors $sim(s_1, s_2) \leftarrow 0$

Nous commençons par vérifier l'impact du seuil de similarité sur le jeu de test TAC 2008 en utilisant les différents *word embeddings* présentés à la section §4.4.3.

En utilisant trois *embeddings* de mots différents, la meilleure performance est obtenue à des seuils de similarité élevés : 0,95 en utilisant W2V, 0,97 en utilisant GloVe et 0,9 en utilisant ConceptNet. Les courbes des figures 4.1, 4.2 et 4.3 partagent la même allure et chacune peut être divisée en trois parties :

1. une partie monotone correspondant aux seuils faibles ;
2. une chute de la performance aux alentours de 0,7 ;
3. une réaugmentation de la performance aux seuils élevés ($> 0{,}9$)

Ce phénomène n'est pas exclusivement remarqué sur les données de TAC 2008 mais aussi sur TAC 2009 (cf. figures 4.4, 4.5 et 4.6)
Les seuils faibles ($< 0{,}5$) n'ont presque aucune influence sur la performance du système. Ceci s'explique par le fait qu'à ce niveau de valeur, le clustering fournit un nombre très faible de clusters non singleton. Au seuil de 0,5, le nombre moyen de clusters non singleton sur TAC 2008 est égal à 6,83 sachant que le nombre moyen de phrases est de 253. À partir de 253 phrases, le clustering réussit à former seulement 6 clusters regroupant la majorité des phrases. Sur TAC 2009, sur 254

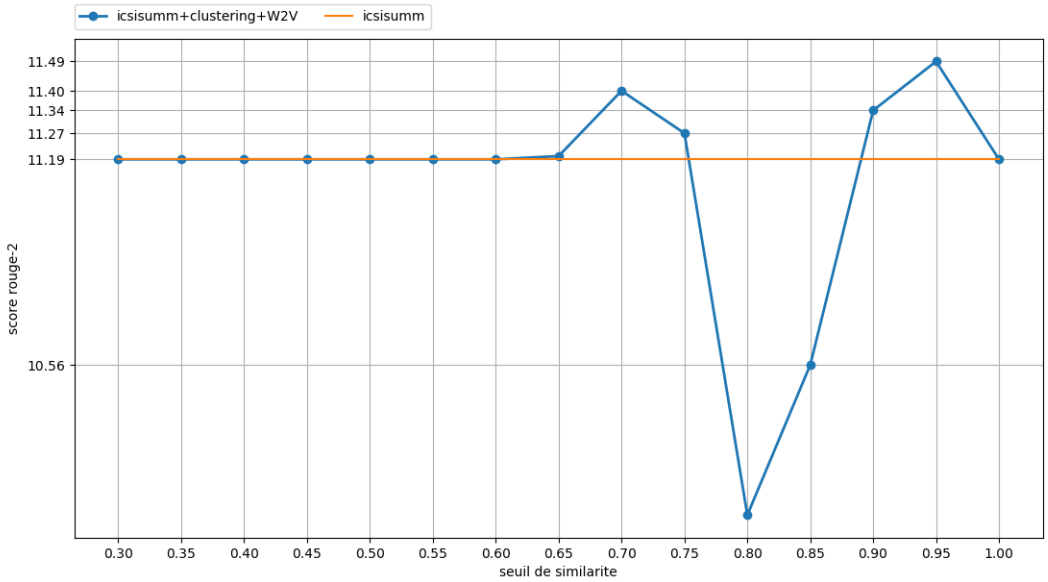

FIGURE 4.1 – Influence du seuil de similarité sur TAC 2008 en utilisant les vecteurs Word2Vec

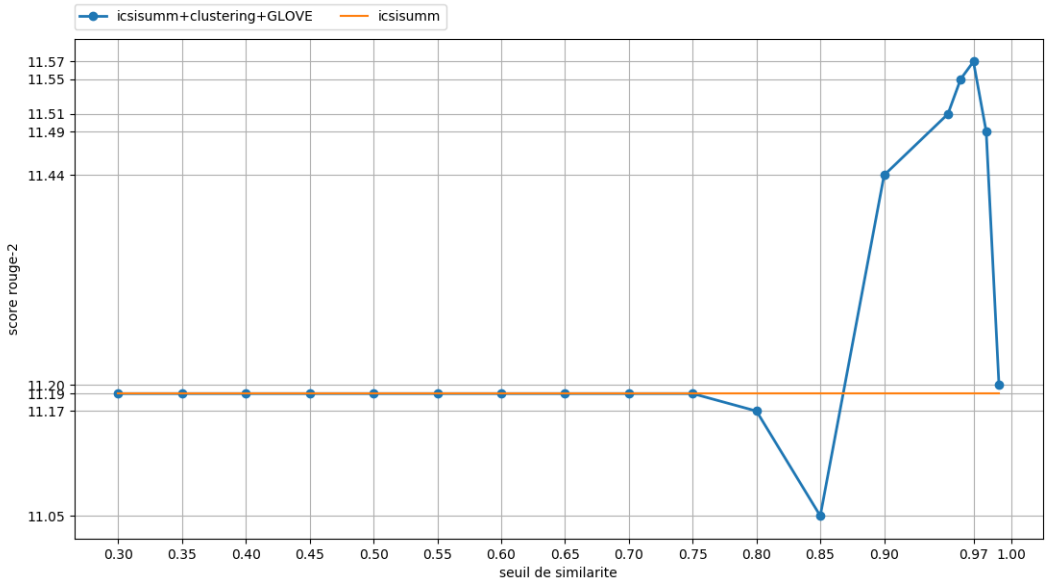

FIGURE 4.2 – Influence du seuil de similarité sur TAC 2008 en utilisant les vecteurs GloVe

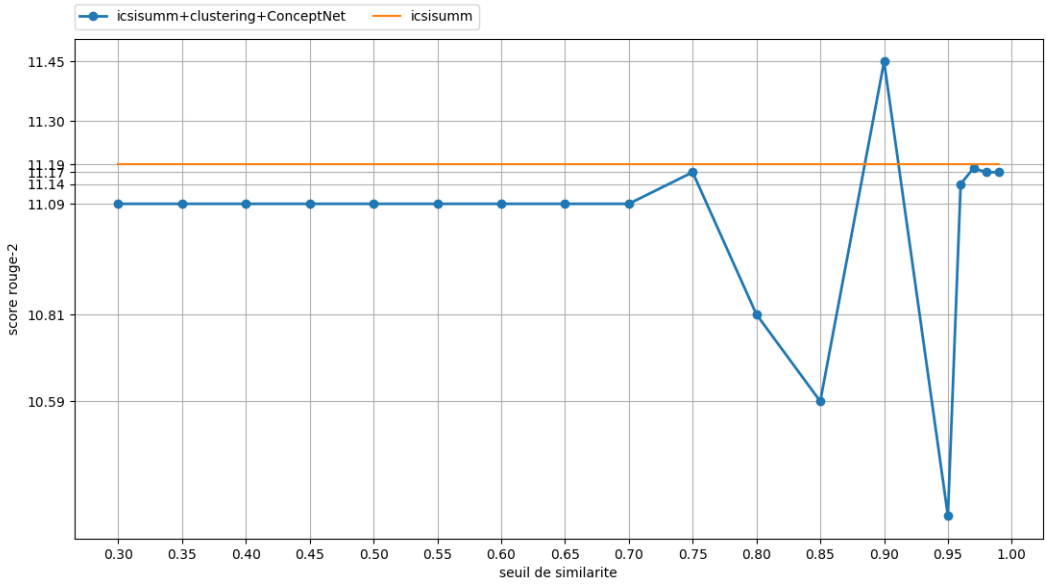

FIGURE 4.3 – Influence du seuil de similarité sur TAC 2008 en utilisant les vecteurs ConceptNet

phrases en moyenne, le clustering forme seulement 6,9 clusters. Plus le seuil de similarité est faible et plus le clustering a logiquement tendance à regrouper un maximum de phrases en un minimum de clusters. Dans cette configuration, tous ces gros clusters regroupent nécessairement des phrases issues des ensembles des anciens et nouveaux documents. Par conséquent, tous les bigrammes de toutes les phrases de l'ensemble des nouveaux documents seront pénalisés. Dans ce cas, la pénalisation appliquée à tous les bigrammes de mots ne permet pas de favoriser des bigrammes au profit d'autres. Ceci explique l'invariance de performance lorsque le seuil de similarité est faible.

On observe sur les trois courbes, une chute de performance avant une amélioration. Cette chute, atteignant le minimum de performance de notre solution, coïncide avec le maximum du nombre de clusters non singleton. Par exemple, sur la figure 4.1, aux seuils de 0,8 et 0,85, le nombre de clusters non singleton est important mais la taille de chaque cluster non singleton est aussi importante. Ceci nous fait pénaliser un nombre important de bigrammes, ce qui rend le système moins discriminant.

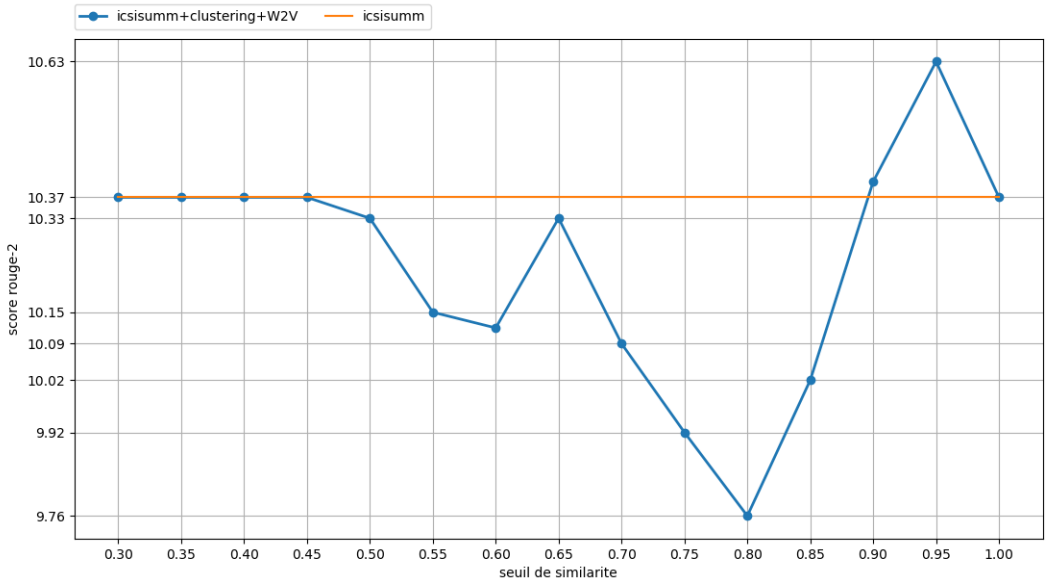

FIGURE 4.4 – Influence du seuil de similarité sur TAC 2009 en utilisant les vecteurs Word2Vec

Dans la troisième partie de chacune des courbes, la performance de notre approche augmente et atteint son maximum. À ces seuils élevés, le nombre de clusters non singleton est moins important mais chaque cluster contient, au plus, trois phrases très similaires. Dans cette configuration, même si on ne détecte pas toutes les phrases de B similaires aux phrases de A, on est sûr que les clusters produits regroupent des phrases très similaires issues de A et B, même si l'idéal serait d'obtenir un nombre maximal de petits clusters non singleton.

Influence du facteur de pénalisation

Nous évaluons aussi notre approche en étudiant l'influence du facteur de pénalisation *alpha* des bigrammes appartenant à des phrases similaires aux phrases des anciens documents. Pour ce faire, nous faisons varier ce facteur de 1,5 à 5 sur TAC 2008 et TAC 2009 en utilisant les vecteurs Word2Vec. Les résultats sont présentés dans les figures 4.7 et 4.8.

Pour les deux courbes, l'amélioration de performance par rapport au cas où

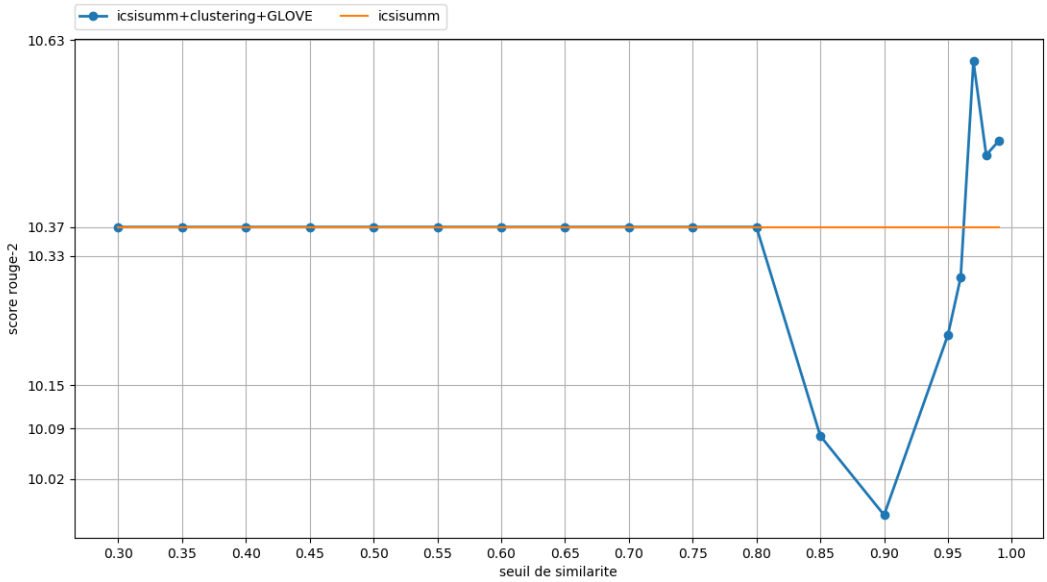

FIGURE 4.5 – Influence du seuil de similarité sur TAC 2009 en utilisant les vecteurs GloVe

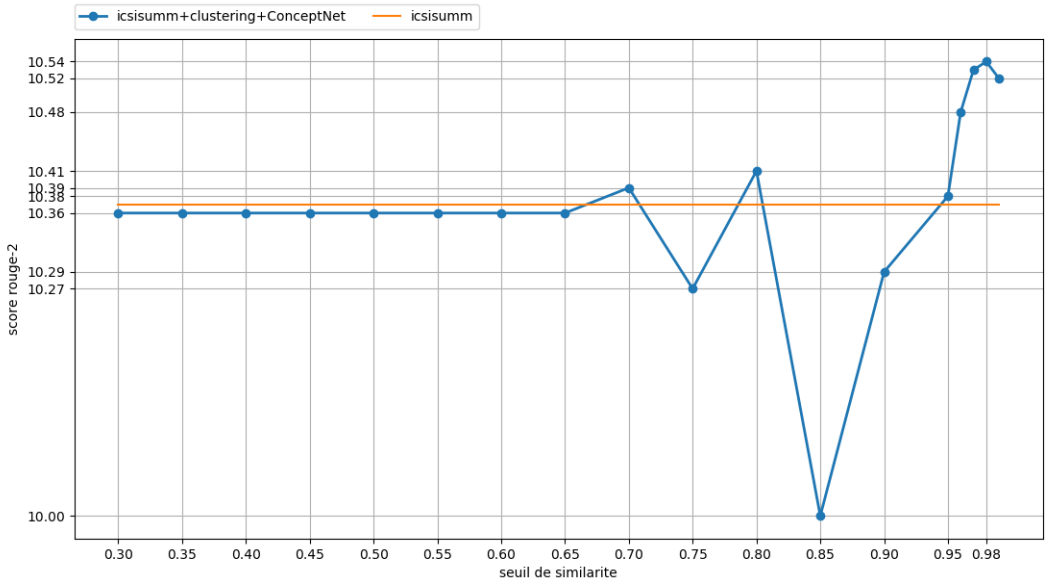

FIGURE 4.6 – Influence du seuil de similarité sur TAC 2009 en utilisant les vecteurs ConceptNet

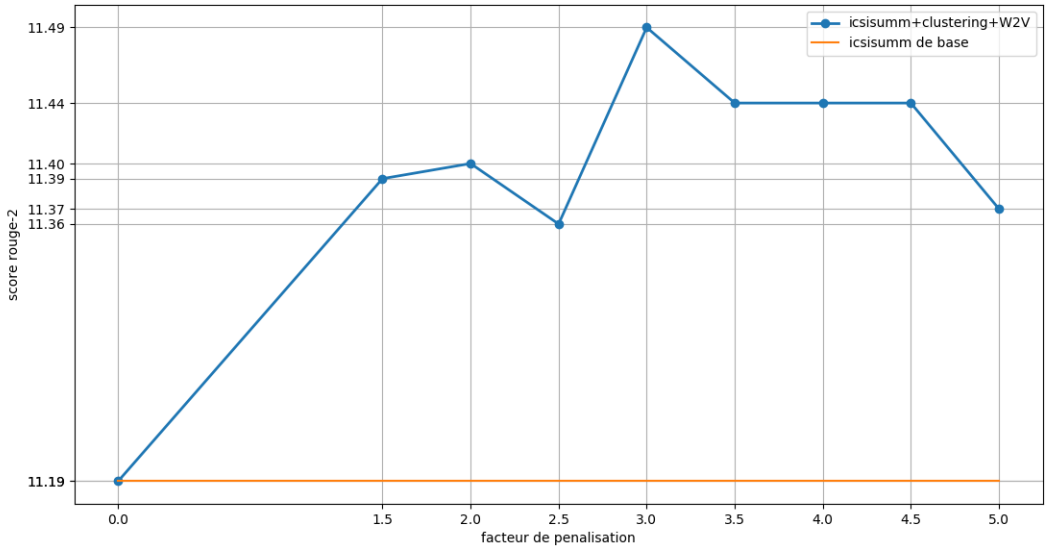

FIGURE 4.7 – Influence du facteur de pénalisation sur TAC 2008 en utilisant les vecteurs Word2Vec

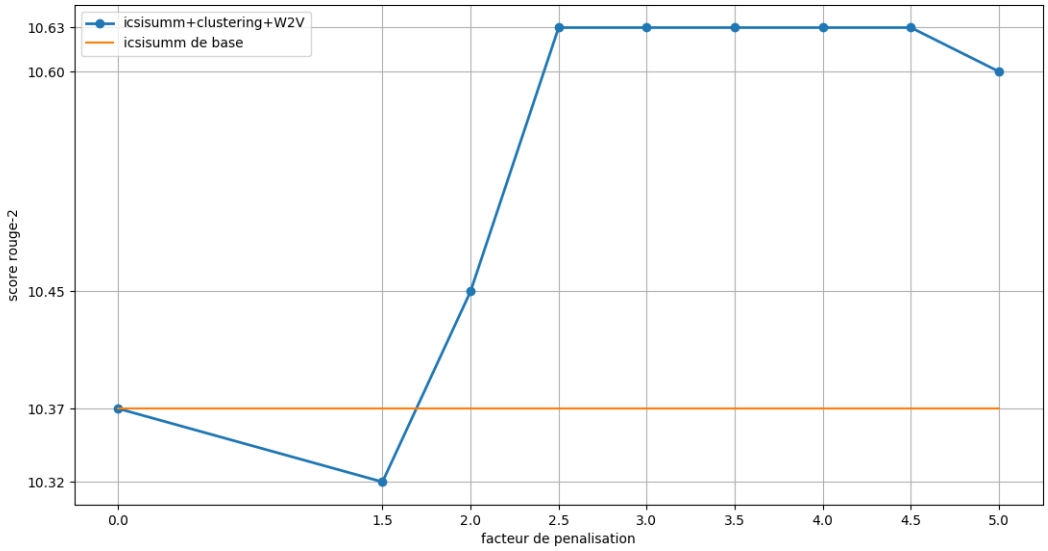

FIGURE 4.8 – Influence du facteur de pénalisation sur TAC 2009 en utilisant les vecteurs Word2Vec

nous n'effectuons aucune pénalisation est observée pour toutes les valeurs de *alpha* (sauf pour $alpha = 1,5$ sur les données de TAC 2009). Ceci confirme la stabilité de la solution et l'efficacité de la pénalisation. Sur TAC 2008, le meilleur score est obtenu pour $alpha = 3$. Sur TAC 2009, le meilleur score est obtenu pour les valeurs de *alpha* allant de 2,5 à 4,5. Nous choisissons donc d'utiliser la valeur 3 de *alpha* pour toutes nos expérimentations.

4.5.2 Évaluation intrinsèque de la similarité sémantique

Afin de confirmer l'efficacité du clustering sémantique et d'expliquer l'écart de performance entre les différents *embeddings* utilisés, nous avons d'abord évalué notre mesure de similarité sémantique sur des données d'évaluation dédiées à la similarité. Nous avons aussi étudié la couverture par les différents *embeddings* du vocabulaire des documents à résumer.

Évaluation de la similarité sémantique

Puisque la similarité sémantique des phrases est centrale dans notre approche, nous avons essayé de caractériser *a posteriori* notre mesure de similarité. Pour être fidèle à la configuration utilisée pour la production des résumés, dans cette évaluation, nous ramenons aussi à 0 les valeurs de similarité inférieures au seuil que nous avons optimisé sur les ensembles de développement. Nous avons appliqué notre mesure de similarité sémantique sur des données d'évaluation standard pour la similarité des phrases [2] : le corpus de paraphrases MSR Dolan et al. (2004) et les ensembles de données SemEval STS Agirre et al. (2016). Dans le cadre de SemEval, la référence est constituée par un ensemble de paires de phrases auxquelles des évaluateurs humains ont attribué un score entre 0 et 5. La signification des valeurs de ce score est explicitée au niveau du tableau 4.4. Le score de référence d'une paire de phrases correspond à la moyenne des scores qu'il leur a été attribué. De leur côté, les systèmes participants produisent eux-mêmes des scores pour les paires de phrases. Un système est évalué en calculant la corrélationde Pearson des scores

2. À notre connaissance, le seul jeu de données spécifiquement dédié à l'évaluation du clustering de phrases dans le contexte du MDS est décrit dans (Geiss, 2009) mais il n'est pas publiquement disponible.

Score de similarité	Signification
5	Les deux phrases sont complètement équivalentes, car elles signifient la même chose.
4	Les deux phrases sont en grande partie équivalentes, mais certains détails sans importance diffèrent.
3	Les deux phrases sont à peu près équivalentes, mais certaines informations importantes diffèrent ou manquent.
2	Les deux phrases ne sont pas équivalentes, mais partagent quelques détails.
1	Les deux phrases ne sont pas équivalentes, mais sont sur le même sujet.
0	Les deux phrases sont complètement dissemblables.

TABLE 4.4 – Signification des scores de similarité de SemEval

qu'il produit pour les paires de phrases test avec leur score de référence. Afin de calculer les scores de précision et de rappel sur ces données, nous considérons un résultat comme un vrai positif si notre similarité donne un score supérieur à 0,95 et que la similarité de référence est supérieure à 3, seuil à partir du quel deux phrases sont considérées comme équivalentes dans la configuration de SemEval. Dans le cas des données du corpus de paraphrases MSR, à chaque paire de phrases est attribué un score égal à 0 ou à 1, avec 0 signifiant que la première phrase ne constitue pas une paraphrase de la seconde et le score 1 signifiant qu'il s'agit d'une paraphrase. Ainsi, pour calculer la précision et le rappel, nous considérons un résultat comme un vrai positif si notre similarité donne un score supérieur à 0,95 et que la similarité de référence est égale à 1. Nous présentons dans le tableau 4.5, l'évaluation de notre mesure de similarité en utilisant les vecteurs de mots pré-entraînés de Google.

Sur tous les jeux de données, notre similarité montre une grande précision mais un faible rappel. Cette tendance est particulièrement perceptible sur le MSR Paraphrase Corpus : lorsque notre système regroupe deux phrases, ce sont des paraphrases dans 91,44% des cas, ce qui correspond à nos hypothèses initiales et illustre leur validité.

Dataset	Précision	Rappel
MSRpara	91,44	17,69
SemEvaL STS 2014	88,00	14,17
SemEvaL STS 2015	90,60	11,46
SemEvaL STS 2016	88,28	25,98

TABLE 4.5 – Évaluation de la similarité de phrases avec le seuil de similarité minimal de 0,95

Couverture du vocabulaire

Nous étudions dans cette section la différence entre les *word embeddings* en terme de couverture du vocabulaire des données d'évaluation en utilisant différentes configurations. Ce vocabulaire est formé par tous les mots des documents à résumer en éliminant les mots vides. Par ailleurs, nous étudions aussi la couverture en tenant compte des mots vides. Nous n'effectuons aucune normalisation des mots avant de calculer la similarité étant donné que les mots des *embeddings* ne sont pas normalisés. En effet, quand la taille du corpus sur lequel on entraîne les *embeddings* est importante, la lemmatisation des mots n'apporte pas d'amélioration. Cependant, quand la taille du corpus d'entraînement est moindre, la lemmatisation contribue à améliorer la représentation de chaque mot étant donné que sa représentation sera calculée à partir de plus d'occurrences. Dans le tableau 4.6, nous donnons les couvertures calculées sur TAC2008. Avec ou sans prise en compte des

	avec mots vides			sans mots vides		
	N tot mots	N mots trouvés	couverture	N tot mots	N mots trouvés	couverture
Google News	27 713	23 804	85,89%	27 584	23 679	85,84%
Glove840B	27 713	26 441	95,41%	27 584	26 312	95,38%

TABLE 4.6 – Couverture des mots simples de TAC 2008 par les word embeddings

mots vides, nous remarquons que les vecteurs GloVe couvrent mieux les données d'évaluation que les vecteurs Word2Vec avec un écart de presque 10% du nombre total de mots. Ceci explique en partie que les scores de MCL-GLOVE-ISumm

soient au-dessus des scores de MCL-W2V-ISumm. Outre la différence de performance, ces résultats montrent que les vecteurs pré-entraînés couvrent très bien notre vocabulaire. C'est pourquoi nous n'avons pas eu besoin d'entraîner des vecteurs sur d'autres corpus. Par ailleurs, ces *embeddings* comportent non seulement des représentations vectorielles des mots simples comme *phone* ou *car* mais aussi de mots composés comme *bad_habits, halloween_costume,* ou *cherry_tomatoes.*

	avec mots vides			sans mots vides		
	N tot mots	N mots trouvés	couverture	N tot mots	N mots trouvés	couverture
Google News	208 762	4 127	1,97%	204 823	4 294	2,09%
Glove840B	208 762	176	0,08%	204 823	115	0,05%

TABLE 4.7 – Couverture des mots composés de TAC 2008 par les word embeddings

Pour extraire les mots composés du jeu de test que nous considérons, nous avons tout simplement extrait toutes les paires de mots consécutifs. Contrairement au cas des mots simples, les *embeddings* recèlent très peu de ces mots composés : 0,05% pour GloVe comme le montre le tableau 4.7. La prise en compte de ces mots composés dans le calcul de la similarité sémantique n'a donc pas amélioré nos résultats. De ce fait, nous nous sommes contentés de prendre en compte seulement les mots simples.

4.6 Conclusion

Pour conclure, nous avons montré que la prise en compte de la similarité interphrase pour l'élimination de la redondance dans le cadre d'un problème de maximisation de couverture améliore la performance du résumé mis-à-jour. Cette amélioration est obtenue en modifiant les poids des bigrammes de mots dont on maximise la couverture. Cette modification est guidée par les résultats du clustering sémantique des phrases, qui nous aide à repérer les informations nouvelles. L'évaluation intrinsèque de la mesure de similarité utilisée a révélé d'autre part que son rappel est très faible par rapport à sa précision. Nous estimons qu'une mesure de similarité avec un rappel plus important serait capable d'améliorer encore plus la

performance de notre approche à condition de conserver une précision importante. Nous avons d'ailleurs montré l'existence d'une telle marge d'amélioration à l'aide d'oracles construits spécifiquement pour les approches extractives. Globalement, nous avons atteint le premier objectif de cette thèse, à savoir la minimisation de la redondance au travers de critères de sélection liés au contenu. Dans le chapitre suivant, nous examinerons comment la prise en compte explicite de la notion de saillance peut également améliorer une approche fondée sur l'ILP pour le résumé de mise à jour.

Chapitre 5

Exploitation de la Structure Rhétorique pour le RA

5.1 Introduction

Conformément aux lignes directrices que nous avons établies au début de ce travail, nous abordons notre second objectif, qui est la maximisation de la saillance à travers des critères de structure. En effet, la structure d'un texte apporte plusieurs éléments informatifs sur le contenu et aide à repérer les segments textuels apportant beaucoup ou peu d'information. Par exemple, dans les textes journalistiques les informations les plus importantes tendent à apparaître en première position tandis que dans les articles scientifiques, l'information pertinente est souvent localisée dans le résumé et la conclusion. Ces exemples illustrent des critères structurels dépendant du genre et du domaine du texte. Cependant, quelle que soit la nature du texte, il existe toujours une certaine cohérence entre les éléments du texte qui justifie la présence de chaque segment textuel à l'endroit où il apparait. L'analyse du discours est l'un des traitements linguistiques qui permettent de détecter cette cohérence. Dans ce chapitre, nous allons proposer une approche pour maximiser la saillance du résumé de mise à jour en nous appuyant sur une méthode particulière d'analyse discursive, la théorie de la Structure Rhétorique. Cette théorie n'est bien sûr pas la seule théorie en analyse du discours mais elle a fait l'objet d'un nombre conséquent de travaux en TAL ayant abouti à des outils exploitables sans restriction particulière quant aux textes traités. Par ailleurs, elle a déjà fait l'objet d'une utilisation dans le cadre du résumé automatique comme nous le montrerons à la section . C'est pourquoi nous l'adoptons ici. Nous la présentons à la section suivante avant de considérer ses applications dans le domaine du résumé automatique.

5.2 La Théorie de la Structure Rhétorique (RST)

La Théorie de la Structure Rhétorique a été fondée en 1987 par Bill Mann, Sandy Thompson et Christian Matthiessen dans le cadre de leurs travaux sur la génération de texte. Il s'agit d'une théorie descriptive de l'organisation du texte. Elle permet de décrire plus spécifiquement les textes en caractérisant leur structure principalement en termes de relations hiérarchiques et fonctionnelles reliant les différentes parties du texte. La spécificité de la RST est qu'elle ne fait aucune hypothèse sur le genre du texte et son formatage. Ceci permet d'appliquer cette théorie à tous les types de textes (p. ex. textes journalistiques, textes académiques, etc.) sans avoir à faire des adaptations. La RST a pour but d'expliquer et de décrire ce qui fait que la structure d'un texte le rend compréhensible dans le cadre d'une communication humaine. Cette théorie repose sur trois hypothèses importantes. Premièrement, elle suppose que la structure d'un texte n'est pas simplement linéaire, sous forme d'une liste de propositions. Selon la RST, le texte consiste en un ensemble de propositions organisées de façon hiérarchique. Des propositions et des groupes de propositions sont liés par différentes relations. La deuxième hypothèse porte sur le point de vue de ces relations. En effet, la RST suppose que le texte doit être annoté en relations tout en restant fidèle au point de vue de son rédacteur, des hypothèses qu'il a fait sur le lecteur final et de la manière dont il veut présenter les faits et les concepts. La troisième hypothèse souligne le fait que les relations asymétriques de type noyau-satellite qui reflètent le fait qu'une partie du texte est subordonnée à une autre partie, sont les plus courantes.
Techniquement, la RST décrit comment un segment de texte est construit à partir d'autres segments en partant de l'intégralité du texte jusqu'à arriver aux plus petits segments non-décomposables. Ces derniers sont généralement des propositions, appelées Unités Élémentaires de Discours (*Elementary Discourse Unit (EDU)*), et représentent les briques de construction du texte. Les relations rhétoriques de la RST sont des relations binaires. Par conséquent, le texte est représenté par un arbre binaire dont la racine représente la totalité du texte. Les différents nœuds représentent les relations qui lient les segments de texte constituants du nœud parent. Au niveau des feuilles, on retrouve les unités non-décomposables, les EDUs. Les deux tâches les plus complexes de la RST sont le typage des relations rhé-

toriques et l'identification du statut "nucléaire" des EDUs. En effet, à quelques

Relation	Noyau	Satellite
Élaboration	information de base	information supplémentaire
Préparation	texte présenté	texte préparant le lecteur à anticiper et à interpréter le texte qui va être présenté
But	le fait/l'action	le but derrière le noyau
Circonstance	le fait/l'action	les circonstances dans lesquelles le noyau a eu lieu
Attribution	ce qui a été dit	fait référence à qui a énoncé les paroles dans le noyau
Explication	le fait/l'action/l'information	explication du noyau
Exemplification	l'information générale	un exemple spécifique du noyau qui le clarifie

TABLE 5.1 – Exemples de relations rhétoriques dans la RST

exceptions près, toutes les relations de la RST sont des relations asymétriques reliant deux entités : le noyau et le satellite. Le noyau est l'élément qui véhicule l'information la plus importante pour le rédacteur et qui reste compréhensible, une fois isolé de son contexte. Le satellite, en revanche, renferme des informations annexes ou moins importante du point de vue du rédacteur. Son rôle est d'enrichir et de compléter le noyau. Généralement, il n'a pas de sens une fois isolé du noyau. Les relations et la nucléarité sont attribuées en utilisant un ensemble de contraintes et de règles appelées schémas. Une relation est attribuée à deux éléments si les deux éléments satisfont les conditions de la relation. Le tableau 5.1 présente quelques relations issues de la RST ainsi que les rôles de chacun de leur noyau et satellite.

Certaines relations rhétoriques sont symétriques et sont dites multi-nucléaires. Dans ces relations, les deux segments de texte sont aussi pertinents l'un que l'autre par rapport au discours du rédacteur. Nous citons à titre d'exemple les relations

de condition et de contraste. Les figures 5.1 et 5.2 illustrent des exemples d'arbres RST accompagnés de leur texte source [1].

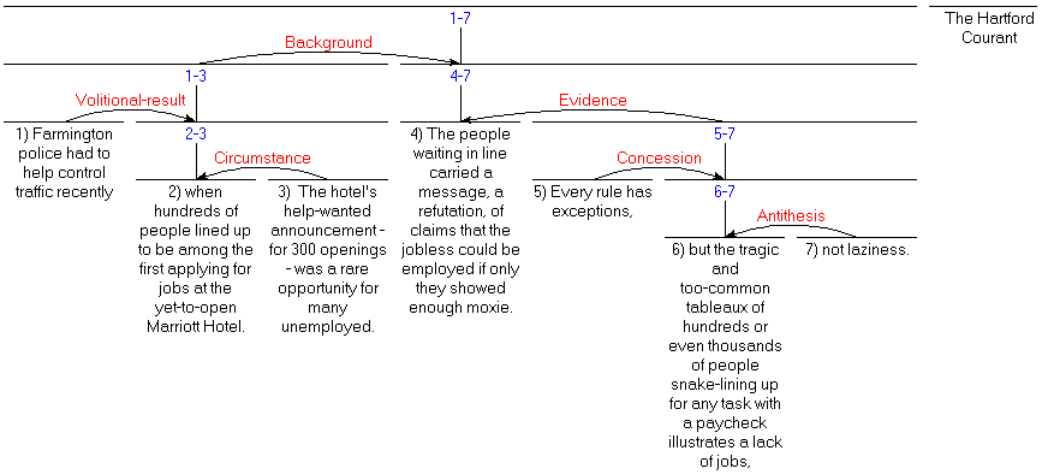

FIGURE 5.1 – Arbre RST d'un court éditorial de journal politique
source : http ://www.sfu.ca/rst/02analyses/published.html

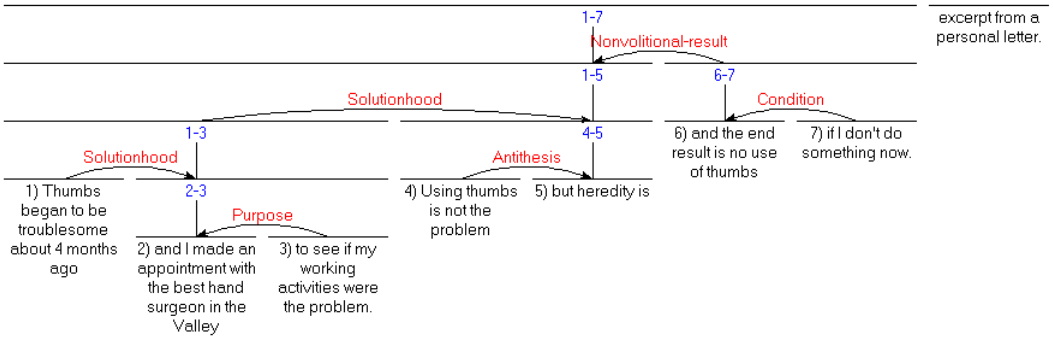

FIGURE 5.2 – Arbre RST d'un extrait d'une lettre personnelle
source : http ://www.sfu.ca/rst/02analyses/published.html

1. Les flèches des arbres RST pointent toujours vers le noyau

5.3 Travaux précédents sur la RST pour le résumé automatique

Bien que la RST aie été conçue initialement pour la génération de texte, la communauté du TAL l'a appliquée pour le résumé automatique ainsi que d'autres tâches. Le résumé automatique fondé sur la RST est considéré comme une thématique de recherche complexe étant donné qu'elle dépend de la performance des analyseurs RST disponibles. De ce fait, les méthodes de l'état de l'art du résumé à base de RST ne sont pas très nombreuses.
Le point commun entre ces différentes méthodes est qu'elles exploitent toutes la notion de nucléarité pour repérer l'information pertinente liée toujours aux noyaux plutôt qu'aux satellites. D'autres critères déduits de la RST sont aussi utilisés mais diffèrent d'une méthode à l'autre. Cependant, la différence majeure entre les méthodes fondées sur la RST est la manière dont elles exploitent l'arbre RST pour produire le résumé. On peut identifier deux catégories de méthodes. Les approches de la première catégorie exploitent l'architecture de l'arbre pour pondérer les unités du discours. Elles produisent comme sortie un ordonnancement des EDUs qui est généralement utilisé pour sélectionner les éléments les mieux classés en guise de résumé. La deuxième famille de méthodes procède à l'élagage des branches inutiles de l'arbre RST et forme le résumé à partir du/des sous-arbres restants. Nous détaillons dans ce qui suit 3 méthodes de référence de RA fondées sur la RST.

5.3.1 Méthodes par classement des EDUs

Méthode de Ono (1994). La méthode de (Ono et al., 1994) est parmi les méthodes les plus simples. En effet, elle n'exploite que la notion de nucléarité dans la RST. Elle a été proposée pour le Japonais mais étant donné sa simplicité elle est applicable à toutes les langues. En partant de l'arbre RST du texte à résumer, Ono et ses collègues attribuent un score initial à la racine de l'arbre correspondant à sa hauteur. Ensuite, ce score est propagé de père en fils à tous les nœuds de l'arbre en effectuant un parcours en profondeur. Un nœud fils noyau hérite du score de son nœud père alors qu'un nœud satellite acquiert le score de son nœud père décrémenté d'un point. On obtient ainsi au niveau des feuilles le score de

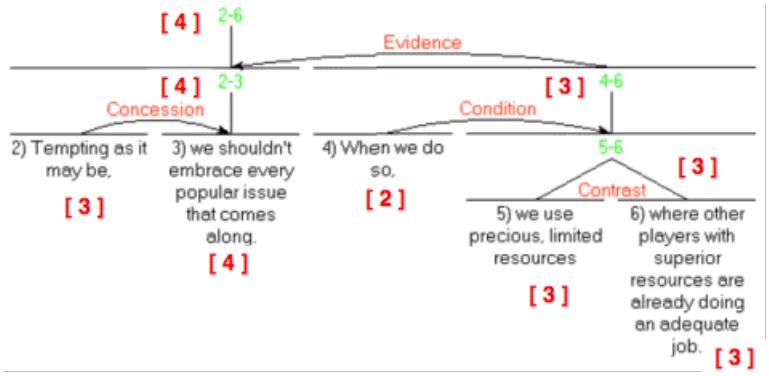

FIGURE 5.3 – Arbre RST d'un extrait d'une lettre personelle pondéré par la méthode de Ono
source : http ://www.sfu.ca/rst/02analyses/published.html

tous les EDUs , ce qui permet de les classer. Plus le score d'un EDU est grand et plus l'information qu'il véhicule est supposée importante. La figure 5.3 montre un exemple d'exécution de la méthode de Ono. Les numéros des EDUs sont indiqués au début du texte de chaque élément. Les chiffres en rouge représentent les poids des nœuds par cette méthode. L'ordre décroissant d'importance obtenu est ainsi : $3 > 2$, 5 et $6 > 4$.

Méthode de O'Donnel (1997) La méthode de (O'Donnell, 1997) part du même principe que la méthode de Ono. Elle propage un poids de la racine vers les feuilles en pénalisant le score hérité à chaque fois qu'un satellite est rencontré. Contrairement à la méthode de Ono, O'Donnel exploite les relations rhétoriques pour pondérer les EDUs. Il attribue un facteur d'importance à chaque relation allant de 0 à 1 en fonction de son type. Par exemple, la relation de concession possède le facteur 0,6. La racine de l'arbre se voit attribuer un score égal à 1. Par la suite, la méthode procède à un parcours en profondeur pour propager les poids des nœuds père aux nœuds fils. Mais ici, quand un satellite est rencontré, il hérite du poids de son père multiplié par le facteur d'importance. Le poids de ce satellite est nécessairement pénalisé puisque le facteur de pénalisation est inférieur à 1.

Méthode de Marcu (1997-1998) La méthode de Marcu (Marcu, 1998b) introduit la notion d'ensemble de promotion en parcourant l'arbre de façon ascendante avant de le pondérer de façon descendante. Dans la première étape, elle détermine

pour chaque nœud son ensemble de promotion en commençant par les feuilles. L'ensemble de promotion d'une feuille est composé de cette feuille elle-même. En remontant dans l'arbre, on construit les ensembles de promotion des nœuds comme étant l'union des ensembles de promotion de leurs fils noyaux.

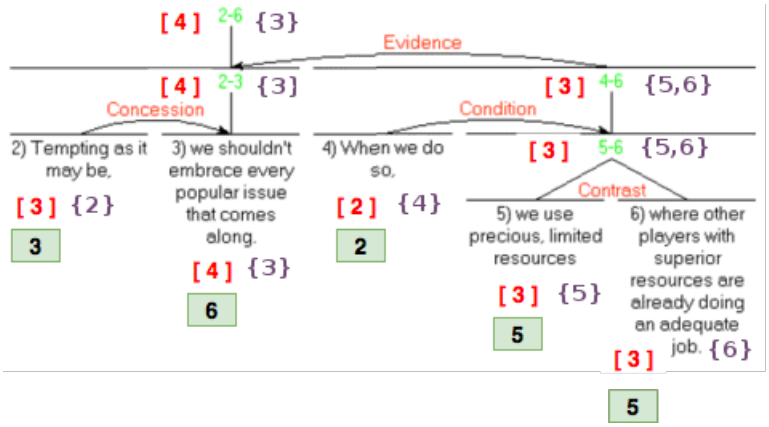

FIGURE 5.4 – Arbre RST pondéré par la méthode de Marcu 1998

Dans la figure 5.4 qui montre l'exécution de cette méthode sur le même exemple que précédemment, les ensembles de promotion des différents nœuds sont indiqués entre accolades. Par exemple, l'ensemble de promotion du nœud interne 5-6 est $\{5, 6\}$ étant donné que les nœuds 5 et 6 sont noyaux. En revanche, l'ensemble de promotion du nœud 2-3 est $\{3\}$ étant donné que le nœud 3 est le fils noyau unique du nœud 2-3. Pour pondérer les EDUs, Marcu suppose que les EDUs qui apparaissent dans les ensembles de promotion des nœuds proches de la racine sont plus importantes que celles apparaissant à des niveaux inférieurs (Marcu, 1997). Le score de l'ensemble de promotion de la racine correspond à la profondeur de l'arbre. Ainsi, plus une EDU est promue près de la racine, meilleur est son score. Dans la figure 5.4, les scores attribués par cette méthode sont indiqués en rouge entre crochets. L'EDU 3 étant promue jusqu'à la racine, elle reçoit le meilleur score. L'ordre décroissant d'importance qui en découle est $3 > 2$, 5 et $6 > 4$. Cette méthode repose globalement sur le même principe que la méthode de Ono et retourne par ailleurs les mêmes scores. Cependant, nous remarquons que grâce à cette méthode, les nœuds 5 et 6 ont été promus de deux niveaux et que le nœud 2 n'a pas été promu à un niveau supérieur alors que les trois EDUs

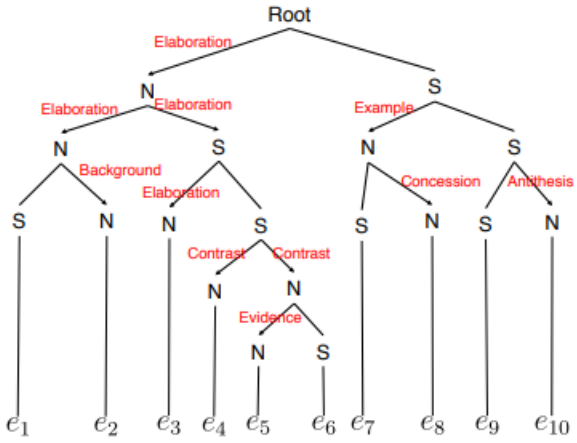

FIGURE 5.5 – Exemple d'un arbre RST (Hirao et al., 2013)

obtiennent le même score. Mais une EDU promue successivement sur plusieurs niveaux devrait être plus importante que celle promue moins souvent. Afin de faire cette distinction, un score de promotion a également été introduit par Marcu (1998b), qui est une mesure du nombre de niveaux sur lesquels une EDU est promue. Le score final d'une EDU, encadré en vert dans la figure 5.4, est le poids calculé initialement (en rouge dans la figure) incrémenté du score de promotion. Le nouvel ordre décroissant d'importance des segments par cette méthode est : 3 $>$ 5 et 6 $>$ 2 $>$ 4.

5.3.2 Méthodes par élagage de l'arbre RST

(Hirao et al., 2013) ont proposé une approche de résumé mono-document fondée sur l'élagage de l'arbre RST. Pour cela, ils commencent par fixer des règles de transformation afin de convertir un arbre RST en un arbre de dépendances. En effet, ils considèrent que l'arbre RST n'est pas adéquat pour l'élagage étant donné que les liens parent-fils n'y sont pas explicites. La figure 5.5 montre le résultat de la transformation en arbre de dépendances de l'arbre RST de la figure 5.6.

L'élagage de l'arbre est formulé comme un problème d'optimisation combinatoire de type sac à dos à contraintes multiples (Multiple-Constrained Knapsack problem). Les objets à mettre dans le sac sont les EDUs. Le poids de chaque EDU est représenté par la moyenne des fréquences des mots qui le constituent. Le

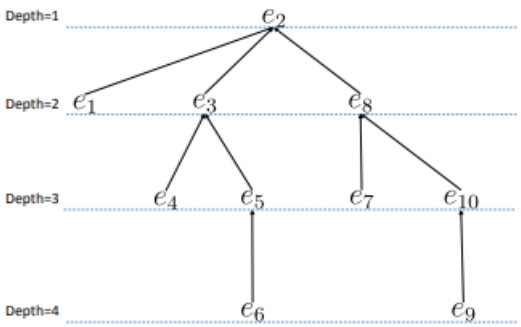

FIGURE 5.6 – L'arbre de dépendances obtenu à partir l'arbre RST de la figure 5.5 (Hirao et al., 2013)

sac à dos est représenté par le résumé final qui ne peut supporter qu'un nombre fixe de mots. À cette contrainte de la longueur du résumé final s'ajoute une autre contrainte qui impose de sélectionner des EDUs représentant un sous-arbre connecté à la racine. La contrainte garantit que si un EDU est sélectionné alors son nœud père doit l'être aussi. Ce problème du type sac à dos à contraintes multiples est ensuite résolu en utilisant la programmation linéaire en nombres entiers.

(Gerani et al., 2014) proposent une méthode abstractive de résumé multi-document pour résumer les avis clients sur des sites de e-commerce. Pour ce faire, cette méthode commence par appliquer un analyseur à chaque texte (avis client) pour obtenir son arbre RST. Ensuite, chaque arbre est modifié de façon à ce que chaque feuille de l'arbre contienne seulement les mots concepts sachant que tout ce qui est évalué dans l'avis client est considéré comme un concept, y compris le produit lui-même. Cette méthode agrège par la suite tous les arbres modifiés des différents avis en un seul arbre dont elle extrait un sous-arbre contenant les concepts les plus importants en utilisant l'algorithme PageRank. Enfin, elle se sert de ce sous-arbre pour générer un résumé en langage naturel contenant les concepts les plus importants et ce, en appliquant un modèle de génération de texte.
Une autre méthode fondée sur l'élagage de l'arbre RST a proposé d'imbriquer les arbres syntaxiques de dépendances des phrases au niveau de l'arbre RST liant ces phrases (Kikuchi et al., 2014). L'objectif est d'établir une forme de fusion entre

le niveau syntaxique et le niveau discursif (RST) afin de prendre en compte, dans le même modèle, les relations inter-phrases et les relations inter-mots. En premier lieu, l'arbre RST est modifié de façon à présenter seulement les résultats inter-phrases. Pour cela, l'arbre RST est converti en un arbre de dépendances comme le font Hirao et al. (2013). Ensuite, les EDUs appartenant à une même phrase sont fusionnés de façon à ce que chaque phrase ait un seul nœud racine. Les relations inter-phrases sont ainsi les relations entres les nœuds racine de chaque phrase. En deuxième lieu, chaque nœud représentant une phrase est remplacé par l'arbre syntaxique de dépendances de cette même phrase (voir figure 5.7). Enfin, la tâche de résumé est formulée comme un problème d'optimisation combinatoire pour le découpage de l'arbre imbriqué. Le découpage prend en compte conjointement les relations entre les phrases et les relations entre les mots afin d'obtenir un sous-arbre enraciné représentant le résumé final.

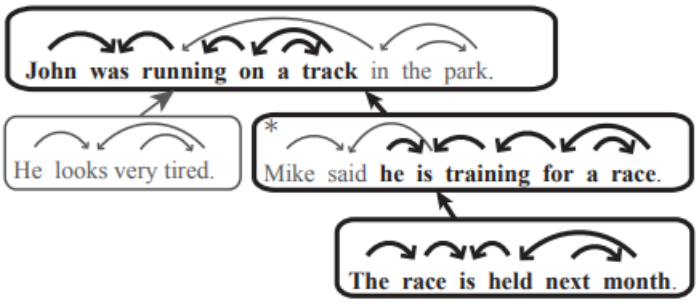

FIGURE 5.7 – Exemple de l'arbre RST imbriquant les arbres de dépendances (Kikuchi et al., 2014)

5.4 Application de la RST pour le résumé mis-à-jour

Dans les travaux de l'état de l'art, les résultats de l'analyse RST ont été utilisés dans la plupart des cas comme un critère principal de repérage des informations pertinentes dans le texte (approche abstractive) ou comme le support d' application d'un algorithme pour déterminer les EDUs du résumé (approche extractive). Notre idée consiste à utiliser la RST comme un critère parmi d'autres dans une

approche de résumé par optimisation globale. À cet effet, nous avons intégré l'information discursive issue de l'analyse RST dans l'approche ILP proposée par (Gillick and Favre, 2009). La figure A.2 de l'annexe A donne une vue d'ensemble en termes systémiques de cette intégration que nous détaillons dans ce qui suit.

5.4.1 Analyseurs RST

Les travaux visant la création d'analyseurs RST sont anciens mais la mise à disposition large d'outils offrant un certain niveau de performance est plus récente. Nous passons rapidement en revue quelques uns de ces analyseurs récents. HILDA est un analyseur RST supervisé fondé sur plusieurs classifieurs SVM : un pour la segmentation en EDUs (*Seg*), un pour la détection des structures (*Struct*) et un pour l'attribution des types de relations et des nucléarités (*Label*) aux structures trouvées (Hernault et al., 2010). Pour la segmentation, le classifieur apprend à déterminer si un mot constitue la frontière d'un EDU ou non. Pour l'attribution des types de relations, les classes sont les relations rhétoriques possibles entre deux EDUs (18 relations). La classification s'appuie dans ce cas sur des critères organisationnels (p.ex. si deux EDUs appartiennent ou pas à la même phrase), des critères syntaxiques et sur la présence d'indicateurs de discours (p. ex. "Cependant" ou "En effet"). Ces différents classifieurs, entraînés sur le RST Discourse Treebank [2] (RST-DT), sont finalement utilisés pour construire l'arbre de façon ascendante. En partant d'une liste de tous les EDUs, le classifieur *Struct* est appliqué à toutes les paires d'EDU pour déterminer la paire d'EDU consécutifs ayant la plus forte probabilité de former une structure (sous-arbre). Ensuite, le classifieur *Label* est utilisé pour déterminer la relation entre les deux EDUs ainsi que leur nucléarité. La paire d'EDUs dans la liste de départ est ensuite remplacée par la nouvelle structure formée. Le processus est répété jusqu'à ce que toutes les structures soient fusionnées dans un seul arbre. Cette approche a atteint un F-score de 93,8% sur la segmentation en EDU.

(Feng and Hirst, 2012) a proposé une amélioration de HILDA en se focalisant spécifiquement sur la construction de l'arbre et l'attribution des relations, HILDA

2. https://catalog.ldc.upenn.edu/ldc2002t07

étant déjà assez performant en ce qui concerne la segmentation en EDUs et la nucléarité. L'amélioration proposée consiste à enrichir l'ensemble des critères utilisés dans HILDA par des critères linguistiques plus profonds en s'inspirant des travaux de (Lin et al., 2009) sur le Penn Discourse Treebank [3] (PDTB). Parmi ces critères, on peut citer la prise en compte du contexte pour déterminer la relation entre deux éléments d'un segment en se servant des relations rhétoriques liant les segments qui le précèdent et le succèdent. La structure de l'arbre est aussi prise en compte pour l'attribution des relations, les relations attribuées aux segments de plus bas niveau dans l'arbre RST pouvant aider à prédire la relation liant les parents directs. La similarité sémantique est aussi utilisée pour détecter des relations pouvant échapper aux filtres syntaxiques ou lexicaux comme la comparaison.

(Joty et al., 2013) ont proposé un système différent appelé TSP. Contrairement aux systèmes précédents qui traitent séparément les composantes structure, relation et nucléarité, (Joty et al., 2013) proposent des modèles probabilistes discriminants, en l'occurrence des champs conditionnels aléatoires (CRF), permettant de déduire les probabilités des éléments de l'arbre RST. Ces modèles sont capables de capturer conjointement la structure et les relations entre les constituants de l'arbre ainsi que leurs dépendances séquentielles. Ce système distingue entre l'analyse discursive inter-phrase et intra-phrase en utilisant une méthode pour chaque niveau. Finalement, il combine les deux analyseurs (inter- et intra- phrase) de deux façons différentes.

DPLP (Ji and Eisenstein, 2014) est un analyseur RST adoptant pour sa part une approche fondée sur le *Representation Learning*, ce que l'on pourrait traduire par apprentissage de caractéristiques ou de représentation. Ce système commence par projeter l'ensemble des critères attachés à une représentation de type sac de mots dans un espace latent de faible dimension dans lequel les relations discursives sont capturées plus facilement. Chaque EDU est ainsi représentée par un vecteur qui reflète sa sémantique ainsi que ses caractéristiques lexico-sémantiques. En apprenant ces vecteurs, le modèle apprend en même temps à prédire la structure discursive en se fondant sur cette représentation vectorielle.

3. https://www.seas.upenn.edu/~pdtb/

Les trois premiers analyseurs utilisent comme critères des représentations locales des EDUs comme les relations syntaxiques, l'appartenance des EDUs aux mêmes phrases, la similarité entre les EDUs, etc. DPLP se distingue par le fait que, en plus des critères locaux, il apprend des représentations distribuées des EDUs, approche que l'on retrouve aussi dans des analyseurs RST récents fondés les réseaux de neurones tels que (Braud et al., 2016), (Li et al., 2016) et (Braud et al., 2017).

Approche	Structure	Nucléarité	Relation
HILDA	83.0	68,4	54,8
TSP	82.74	68.40	55.71
DPLP	82.08	71.13	61.63

TABLE 5.2 – Comparaison des performances des analyseurs RST sur le corpus RST-DT

Les performances des différents analyseurs RST sont assez proches comme le montre le tableau 5.2. Cependant, HILDA dépasse légèrement les autres systèmes sur la Structure et DPLP l'emporte pour la nucléarité et les relations. Étant donné, que DPLP est publiquement disponible [4] et qu'il était le plus simple à utiliser, notre choix s'est porté sur ce système. De plus, il fournit un modèle pré-entraîné sur les données d'entraînement du corpus de référence RST-DT.

5.4.2 Intégration de la RST dans l'ILP

Les évaluations présentées dans le tableau 5.2 montrent que les analyseurs RST sont à l'heure actuelle plus performants dans la segmentation en EDUs et la détermination de nucléarité que dans l'attribution des relations. Nous avons pu le constater aussi en appliquant l'outil DPLP sur nos données. Les arbres RST produits présentent majoritairement des relations d'élaboration, une relation pas assez distinctive pour la tâche de résumé automatique. Les figures 5.8 et 5.9

4. https://github.com/jiyfeng/DPLP

European airplane maker Airbus "is likely to discuss before the end of the year" a possible increase in production capacity of its new super-jumbo A380 aircraft, Airbus' production chief Gustav Humbert said in a magazine interview "We're already sold out until 2010," Humbert told the weekly Focus Money in comments released ahead of publication on Thursday. "There is room" for an increase in capacity, he added. Airbus has so far received orders for 149 of the new giant aircraft with the first delivery scheduled for 2006. Production is expected to be running at full pelt from 2008.

FIGURE 5.8 – Exemple d'un article de journal

montrent respectivement un exemple des documents à résumer ainsi que son arbre RST généré par DPLP.

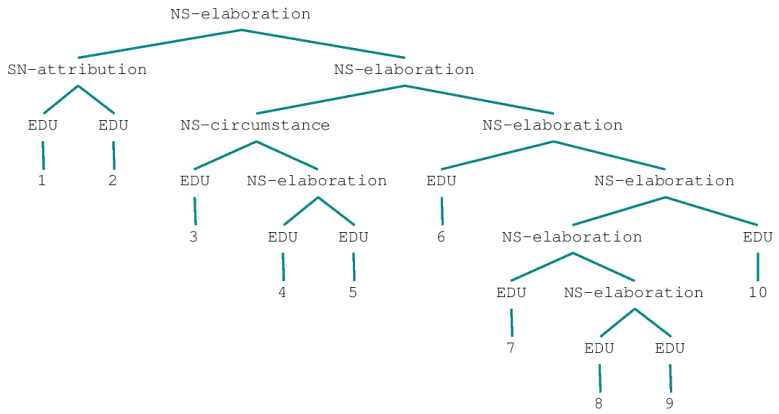

FIGURE 5.9 – Arbre RST généré par DPLP

Pour cette raison, nous avons choisi d'exploiter seulement la structuration hiérarchique et la nucléarité de la RST. Ce choix est aussi validé par une étude comparative des indicateurs discursifs pour le résumé automatique (Louis et al., 2010a). Cette étude montre que les critères structurels du discours sont plus indicatifs pour le RA que les critères sémantiques relationnels.

5.4.3 Méthode de pondération des EDUs

Pour le résumé de mise-à-jour, nous partons du même système de base ICSISummn, que pour notre première contribution (cf. chapitre 3). Notre objectif

consiste à exploiter l'analyse RST pour aider le modèle ILP à sélectionner les phrases les plus saillantes de l'ensemble des documents B (*i.e.* ensemble des documents récents). Le constat que la meilleure façon d'injecter de nouvelles informations dans un modèle ILP passe par le poids des concepts à couvrir reste valable dans ce contexte. Par conséquent, nous avons modifié les poids des bigrammes de mots dans notre modèle ILP de façon à favoriser les éléments saillants du point de vue de la RST. Pour cela, nous nous sommes inspiré principalement des méthodes par classement des EDUs pour déduire les poids de ces dernières. Nous avons choisi la pondération par la méthode la plus simple, celle de (Ono et al., 1994). En effet, malgré sa simplicité, cette méthode a pu dépasser les autres méthodes décrites dans le paragraphe 5.3.1 selon une étude comparative des méthodes de résumé fondées sur la RST (Uzêda et al., 2010). Pour différentes langues et pour différents analyseurs RST, la méthode de Ono y a toujours obtenu les meilleurs scores bien que les différences entre les méthodes n'étaient pas significatives. De ce fait, et comme le recommandent (Uzêda et al., 2010), il vaut mieux choisir la méthode de Ono, qui représente le meilleur compromis entre les critères de coût et de performance.

5.4.4 Méthode de pondération des bigrammes

Nous situant dans le contexte du résumé multi-document, nous commençons par construire l'arbre RST de chaque document de la collection à résumer. Ensuite, nous procédons à la pondération des EDUs de chaque document par la méthode de Ono. Pour déduire le $poids_{RST}$ d'un bigramme, les EDUs contenant ce bigramme au moins une fois sont repérés dans tous les documents. Le $poids_{RST}$ est ensuite calculé comme étant la moyenne des poids des EDUs où le bigramme apparaît. Vu que le poids de Ono est proportionnel à la hauteur de l'arbre RST qui est, à son tour, proportionnel à longueur du document, nous normalisons les poids des EDUs par la hauteur de l'arbre RST dont elles sont issues. Par conséquent, tous les poids des EDUs seront compris entre 0 et 1. Comme les poids des bigrammes ne sont qu'une moyenne des poids des EDUs, le $poids_{RST}$ d'un bigramme sera aussi compris entre 0 et 1. D'après la méthode de Ono, plus le poids de l'EDU est proche de la hauteur de l'arbre et plus l'EDU est important. En projetant ceci sur

Système	ROUGE-2		
	DUC 2007	TAC 2008	TAC 2009
ISumm	7,59	11,19	10,37
MCL-W2V-ISumm	**8,14**	**11,49**	**10,48**
RST-ISumm	7,89	11,40	10,41
MCL-RST-ISumm	8,14	11,49	10,48

TABLE 5.3 – Évaluation de l'intégration de la RST dans le modèle ILP

les bigrammes, on déduit que plus le $poids_{RST}$ est proche de 1 et plus ce bigramme est important. Comme les poids initiaux w_i des bigrammes sont des fréquences en documents allant de 1 à 10 (de 3 à 10 après l'élagage), pour injecter les $poids_{RST}$, nous utilisons ces derniers comme un facteur de pénalisation qui laisse inchangés les poids des bigrammes les plus importants mais qui pénalise les bigrammes les moins importants. L'équation 5.1 résume la modification de poids effectuée pour les bigrammes.

$$poids_{RST}(c_i) \leftarrow w_i \times \frac{\sum_{\{edu|c_i \in edu\}} \frac{poids(edu)}{hauteur(arbre_{edu})}}{card|\{edu|c_i \in edu\}|} \tag{5.1}$$

où c_i est le bigramme dont nous modifions le poids et arb_edu fait référence à l'arbre RST d'où l'élément edu est issu.

5.4.5 Évaluation du système avec les nouveaux poids

Nous présentons dans cette partie les résultats de l'évaluation de la modification des poids des bigrammes par un critère fondé sur la RST. En premier lieu, nous effectuons la modification des poids sur le modèle de base ISumm. Nous appelons cette variante RST-ISumm. Ensuite, nous présentons les résultats de la variante MCL-RST-ISumm dans laquelle les poids des bigrammes sont modifiés d'une part selon la méthode fondée sur la similarité sémantique et d'autre part selon la méthode fondée sur la RST. Le tableau 5.3 présente ces résultats sur les données d'évaluation de DUC 2007, TAC 2008 et TAC 2009.

Nous remarquons d'abord que l'approche fondée sur la similarité inter-phrase devance la variante fondée sur la RST présentée dans ce chapitre. RST-ISumm réus-

sit tout de même à améliorer les résultats du modèle de départ sur les trois jeux de données, bien que cette amélioration reste minime sur TAC 2009. Cependant, l'amélioration sur DUC 2007 et TAC 2008 est non négligeable, ce qui confirme l'utilité de l'analyse RST dans ce type d'approche.
D'autre part, l'utilisation simultanée des critères sémantique et discursif représentée par la variante MCL-RST-ISumm, ce qui représente une forme de fusion précoce des deux critères, n'arrive pas à améliorer les résultats par rapport à chacune des approches. De plus, cette version obtient les mêmes scores que MCL-W2V-ISumm. Ceci suggère que le critère fondé sur la RST n'apporte rien par rapport au critère sémantique. Cependant, en comparant les scores ROUGE des deux approches sur chaque échantillon des jeux de données, nous avons remarqué que pour plusieurs échantillons, RST-ISumm fait mieux que MCL-W2V-ISumm. Il existe donc des cas que RST-ISumm gère mieux que MCL-W2V-ISumm. Mais cette contribution n'a pu être observée dans le système utilisant les deux critères simultanément. On doit donc faire l'hypothèse que la pénalisation de certains bigrammes par rapport à d'autres réalisée par RST-ISUMM est indirectement remise en cause par la pénalisation appliquée par MCL-W2V-ISUMM. Les deux modifications de pondération des bigrammes se neutralisentau moins partiellement.
Afin de bénéficier des points forts de chacune de ces approches, nous étudions dans la section suivante la piste d'une fusion des résultats de ces systèmes, c'est-à-dire une fusion tardive des deux critères.

5.5 Fusion de systèmes de résumé

L'intégration de la similarité sémantique inter-phrase pour minimiser la redondance a permis d'améliorer la performance du modèle initial. Parallèlement, l'exploitation de l'analyse discursive pour maximiser la saillance a aussi amélioré le même modèle initial. Néanmoins, nous avons constaté que la prise en compte simultanée des deux aspects sémantique et discursif dans le même modèle n'arrive pas à faire mieux que chacune des deux approches. Nous supposons donc que la modification des poids des bigrammes par les deux méthodes conjointement, ce qui correspond à une fusion dite précoce, n'est pas la meilleure façon pour fusionner les deux approches. De ce fait, nous avons exploré la piste d'une fusion dite tar-

dive au travers de l'agrégation de systèmes de résumé. Ces méthodes proposent de fusionner les sorties des différents systèmes de résumé au lieu de combiner les méthodes dans un seul système. Cette approche a été appliquée dans plusieurs tâches du TAL comme l'amélioration des thésaurus distributionnels (Ferret, 2015) ou la traduction automatique (Rosti et al., 2007) et surtout en Recherche d'Information (RI) pour ré-ordonnancer les listes des documents répondant à une requête. En effet, quand les méthodes sont assez hétérogènes ou nombreuses, il devient difficile de les combiner au sein d'une même approche. Mais ce type de fusion ne peut être intéressant que si les résultat à fusionner ne sont pas trop proches. Dans le cas qui nous occupe, bien que les performances des différents systèmes de résumé de l'état de l'art soient très similaires, le contenu des résumés produits est assez différent. Autrement dit, chacun de ces systèmes arrive à prédire une partie différente de la bonne réponse. Plus précisément, le score ROUGE d'un système sur un jeu de données est la moyenne des scores ROUGE de ce système sur chaque échantillon du jeu de données. Deux systèmes différents peuvent avoir le même score ROUGE, donc une même performance, mais l'un peut dépasser l'autre sur un ou plusieurs échantillons du jeu de données. L'idée de la combinaison de système est de retenir la meilleure partie de la réponse (i.e. sortie) de chacun des systèmes à combiner.

5.5.1 Travaux précédents

Malgré l'intérêt porté au résumé automatique de texte depuis une vingtaine d'années, les travaux ayant étudié la fusion tardive des résultats des systèmes de résumé sont rares.
(Wang and Li, 2012) furent les premiers à s'intéresser à ce sujet en essayant de fusionner les listes de classement des phrases fournies par chacun des systèmes de résumé en une liste optimale représentant le consensus pondéré des différents systèmes. Le point de départ est un ensemble de listes de classements $\{r_1, r_2, ...r_k\}$ de k systèmes de résumé. Le j^{eme} élément d'une liste r_i indique le classement attribué par le système i à la phrase en position j. L'objectif est d'obtenir le classement optimal r^* par un ensemble de poids $\begin{bmatrix} w_1 & w_2 & ... & w_k \end{bmatrix}^T$ minimisant la distance entre r^* et tous les r_i. En trouvant les poids minimisant cette distance, le classement optimal est :

$$r^* = \sum_{i=1}^{k} w_i \cdot r_i$$

(Pei et al., 2012) s'inscrivent dans le même cadre que (Wang and Li, 2012) en cherchant le poids à attribuer à chaque système mais le font de façon supervisée en utilisant *SVM Ranking* (Joachims, 2002). Le classement final est calculé de la même façon. Un peu plus tard, (Hong et al., 2015) ont proposé une méthode de fusion des systèmes de résumé fondée sur les scores plutôt que les rangs. En commençant par générer toutes les combinaisons de phrases candidates au résumé final, un modèle de régression est entraîné pour estimer le score ROUGE-1 du résumé candidat. Pour cela, le modèle repose sur différents critères comme la similarité avec les textes source, la position des phrases, la redondance, la fréquence des mots dans de larges corpus, etc.
Enfin, (Mehta and Majumder, 2018) étudient l'agrégation de systèmes de résumé en utilisant trois systèmes de fusion de l'état de l'art mais en fusionnant à chaque fois les sorties soit de systèmes différents soit du même système mais en faisant varier à chaque exécution les critères de sélection (p. ex. la similarité sémantique ou le modèle de représentation textuelle). Ils montrent ainsi que toutes les techniques d'agrégation utilisées fonctionnent beaucoup mieux lorsque les listes de classement sont générées par le même système mais en utilisant des mesures de similarité différentes par rapport à la combinaison de listes de classement provenant de systèmes très différents.

5.5.2 Limite supérieure de la fusion de systèmes

Dans un premier temps, nous examinons la limite supérieure de la fusion des résultats des différents systèmes de résumé afin de définir si une telle fusion peut théoriquement aboutir à une amélioration. Pour cela, nous choisissons d'agréger les résumés issus de ISumm, MCL-W2V-ISumm et RST-ISumm. Pour chaque instance du jeu de données, et sur la base des résumés issus des trois systèmes, nous extrayons toutes les combinaisons de 3, 4 à 5 phrases dont la taille est inférieure à 100 mots. Le résumé retenu est celui qui maximise la métrique ROUGE. Nous avons effectué ce test sur TAC 2008 en optimisant à chaque fois soit la F-mesure (F-Oracle), la précision (Prec-Oracle) ou le rappel (Rec-Oracle). Quelle que soit lea mesure que nous optimisons, l'oracle réussit à dépasser chacun des systèmes

Système	ROUGE-2		
	rappel	précision	F-mesure
ISumm	11,15	10,78	10,95
MCL-W2V-ISumm	11,45	11,10	11,26
RST-ISumm	11,40	11,05	11,22
F-Oracle	11,22	13,54	12,21
Rec-Oracle	11,65	11,92	11,75
Prec-Oracle	10,19	14,11	11,73

TABLE 5.4 – Évaluation des résumés Oracle agrégés

fusionnés en termes de précision et de F-mesure. Il faut noter en premier lieu que la précision est plus facile à améliorer que le rappel et atteint des scores importants. Ce constat n'est pas surprenant dans la mesure où les méthodes de fusion mettent intrinsèquement l'accent sur les éléments communs entre les différents systèmes, ce qui favorise la précision. Néanmoins, on observe qu'une amélioration du rappel est possible, entraînant de concert une amélioration de la précision aboutissant au final à un rappel et une précision supérieurs à ceux des systèmes fusionnés. Ces résultats, ainsi que ceux reportés dans (Hong et al., 2015), montrent que la fusion de systèmes peut produire de meilleurs résumés que ceux produits par chaque système. Notre objectif est alors de capturer une partie de cette amélioration afin de combiner les points forts de chacune des approches que nous avons proposées.

5.5.3 Méthodes de fusion utilisées

Nous implémentons 5 méthodes de fusion de l'état de l'art, dont l'une est dédiée au résumé automatique.

1. Fusion de Borda. La méthode de Borda est une méthode de vote très ancienne qui a été formalisée par Jean-Charles de Borda en 1770 (de Borda, 1781). Cette méthode a été largement utilisée en TAL par exemple pour l'annotation en rôles sémantiques (Robles et al., 2010), l'analyse de sentiments (Grandi et al., 2016), etc. Son principe est assez simple. Disposant de plusieurs listes de classement, dans chaque liste de taille n on attribue au premier élément le score n, au deuxième le

score $n-1$ et ainsi de suite. Par exemple, le $2^{\text{ème}}$ item d'une liste de 6 éléments obtient le score $6-1=5$. Ensuite, pour chaque élément on calcule la somme de ses scores issus des différentes listes. Finalement, pour obtenir l'ordonnancement final, on classe les éléments par ordre décroissant de leur score. Plus le score d'un élément est grand, meilleur est son classement.

2. Fusion de Condorcet. Le vote de Condorcet est une méthode très ancienne aussi (marquis de Condorcet, 1785) utilisée jusqu'à aujourd'hui dans la fusion de listes d'ordonnancement. Elle est aussi répandue que la méthode de Borda dans le domaine du TAL et de la RI. Elle a ainsi été appliquée, par exemple, pour les systèmes de question réponse (Agarwal et al., 2012) et pour la sélection d'attributs (*feature selection*) pour le texte (Wu et al., 2009). Alors que la méthode de Borda est fondée sur le rang des éléments dans les listes de classement, la méthode de Condorcet s'appuie sur le choix majoritaire en opérant des séries de comparaisons entre les paires d'éléments. Si nous devons classer 3 éléments x_i, alors pour chaque paire d'éléments (x1,x2), on examine à travers les listes de classement, si x1 est préféré par rapport à x2 ou l'inverse. Finalement, le gagnant (classé premier) est celui qui devance tous les autres éléments le plus fréquemment dans ces comparaisons binaires.
Prenons l'exemple de 4 systèmes de résumé S1, S2, S3 et S4 classant 4 phrases a, b, c et d. Les classements sont les suivants :

S1 : a > b > c > d
S2 : b > c > a > d
S3 : c > b > a > d
S4 : c > a > b > d

Selon la méthode de Borda les scores des différents éléments sont :
Borda(a)=4+2+2+3=11
Borda(b)=3+4+3+2=12
Borda(c)=2+3+4+4=13
Borda(d)=1+1+1+1=4
Par conséquent, le classement final est c>b>a>d.
En revanche, selon la méthode de Condorcet, les comparaisons par paires donnent :

a>b selon 2 systèmes et b>a selon 2 systèmes
a>c selon 1 système, et c>a selon 3 systèmes
b>c selon 2 systèmes et c>b selon 2 systèmes
a>d selon 4 systèmes et d>a selon 0 système
b>d selon 4 systèmes et d>b selon 0 système
c>d selon 4 systèmes et d>c selon 0 système
Au final, la première phrase ainsi sélectionnée par la méthode de Condorcet est "b". En cherchant de la même façon les phrases suivantes, le classement global est : b>c>a>d, ce qui est différent du classement trouvé par la méthode de Borda.

3. Comb-SUM. Cette méthode se fonde sur les scores plutôt que les rangs (Fox and Shaw, 1994). Le score agrégé d'un élément est tout simplement la somme de ses scores issus des différents systèmes. Une normalisation des scores des différents systèmes est appliquée lorsque ces scores ne sont pas directement comparables.

4. RRF (Fusion des Rangs Réciproques). Cette méthode calcule les nouveaux rangs des éléments à partir de n systèmes s_i de la façon suivante :

$$RRF(x) = \sum_{i=0}^{n} \frac{1}{k + rang(x))} \tag{5.2}$$

où k est une constante destinée à atténuer les rangs très élevés. Le but est de donner la chance aux éléments classés très bas de participer au résultat final. Malgré sa simplicité, cette méthode a montré son efficacité particulièrement vis-à-vis de la méthode de Condorcet et des méthodes supervisées comme RankSVM (Cormack et al., 2009).

4. WCS. est le système que nous avons décrit dans la section 5.5.1.

5.5.4 Mise en oeuvre de la fusion de systèmes et évaluation des résultats

Comme le modèle ILP que nous utilisons pour le résumé est un modèle de maximisation de couverture, il ne produit pas un classement des phrases. Il sélectionne les phrases qui maximisent la somme des poids des bigrammes sélectionnés

tout en respectant les contraintes définies. Afin d'utiliser les méthodes de fusion des listes de classement, nous avons recalculé *a posteriori* les scores des phrases des résumés produits par chacun de nos systèmes comme étant la somme des poids des bigrammes qu'ils contiennent. Nous ordonnons ensuite les phrases de chaque résumé en fonction de ce score pour obtenir une liste de classement. Le tableau 5.5 présente les résultats de l'évaluation des différentes méthodes de fusion appliquées à ces listes.

Système	ROUGE-2		
	rappel	précision	F-mesure
ISumm	11,15	10,78	10,95
MCL-W2V-ISumm	11,45	**11,10**	11,26
RST-ISumm	11,40	11,05	11,22
Borda	11,48	11,01	11,23
Condorcet	11,41	10,05	11,22
CombSUM	11,32	10,86	11,08
RRF	**11,51**	11,05	**11,27**
WCS	11,12	10,76	10,93

TABLE 5.5 – Évaluation de la fusion de résumés

La méthode de Borda a réussi à composer des résumés dont le rappel dépasse légèrement celui de chacun des trois systèmes : ISumm, MCL-W2V-ISumm et RST-ISumm. Cependant, les méthodes de Condorcet et CombSUM n'arrivent pas à atteindre cet objectif. La méthode RRF [5], qui est l'une des plus simples, a pu améliorer le rappel du score ROUGE-2 par rapport aux 3 modèles de départ de façon notable. De façon surprenante, la méthode WCS fondée sur l'optimisation globale a obtenu des scores plus faibles que ceux de toutes les autres méthodes d'agrégation. Dans l'article proposant WCS, ce dernier a dépassé sur deux jeux de données 8 baselines dont la méthode de Borda. Cette contre-performance peut être due à la nature des données fusionnées. En effet, les résumés que nous voulons fusionner sont issus de modèles assez similaires. On peut alors supposer que les

5. avec le paramètre k fixé à 60

méthodes de fusion de listes de classement qui ont montré leur efficacité sur des données hétérogènes ne sont pas forcément aussi performantes sur des listes de classement proches.
En revenant au repère défini par les résumés de fusion Oracle, on peut constater que la méthode RRF n'a réussi à capter qu'une partie de l'amélioration possible. Pour aller plus loin, on pourrait penser à ajouter d'autres résumés à l'entrée des systèmes de fusion afin de diversifier les listes de classement et profiter ainsi de la performance attendue de méthodes sophistiquées telles que WCS.

5.6 Conclusion

Dans ce chapitre, nous avons présenté la théorie de la structure rhétorique ainsi que les travaux précédents sur le RA l'ayant exploitée. Après avoir fait un tour d'horizon des analyseurs RST disponibles actuellement, nous avons présenté la méthode que nous avons définie pour intégrer l'information rhétorique dans notre modèle ILP de base. Bien que globalement inférieurs aux résultats obtenus par la prise en compte explicite de la redondance sémantique au niveau phrastique, les résultats issus de cette intégration permettent eux aussi d'améliorer le modèle ILP initial, montrant ainsi l'intérêt de l'information rhétorique pour le résumé de mise à jour. Finalement, afin de combiner l'aspect sémantique étudié dans les chapitres 3 et 4 et l'aspect rhétorique étudié dans le chapitre courant, nous avons exploré la problématique de la fusion tardive de systèmes par fusion des listes de classement. De façon assez surprenante, les gains observés concernent plus spécifiquement le rappel et restent limités en comparaison des gains que nous avons montrés comme possibles en théorie. Le chapitre suivant conclut cette thèse en synthétisant les constatations les plus notables et en proposant les pistes d'amélioration jugées les plus prometteuses.

Chapitre 6

Bilan et perspectives

Notre objectif initial durant cette thèse était d'améliorer la qualité des résumés mis-à-jour. Détecter la nouveauté dans les informations, réduire la redondance et identifier les éléments les plus pertinents dans le texte furent les problématiques principales auxquelles nous avons essayé de répondre.
Ce document ne couvre pas la totalité des travaux de recherche menés mais en cite les plus pertinents et les plus fructueux. En effet, le choix des approches présentées dans cette thèse n'a pas été direct et nous avons effectué un grand nombre d'expérimentations et de tests que nous n'avons pas explicités dans ce manuscrit. Nous commençons, dans ce dernier chapitre, par présenter un bilan récapitulatif des contributions proposées. Ensuite, nous discutons les orientations de recherche futures et les pistes d'amélioration dans notre contexte.

6.1 Bilan

Comme nous l'avons souligné lors de la présentation de notre problématique en introduction, ce travail s'articule autour de deux axes principaux : la détection de la redondance et la maximisation de la saillance pour le résumé mis-à-jour. Pour le premier axe, nous avons présenté dans le chapitre 3 la façon dont nous avons pris en compte la similarité inter-phrases dans le cadre du résumé mis-à-jour par optimisation sous contraintes pour détecter la redondance avec les anciennes informations. À ce niveau, nous avons constaté d'après les expérimentations que dans les modèles de maximisation de couverture, ajouter des composantes à optimiser dans l'objectif n'est pas la meilleure façon d'intégrer des critères dans le modèle. Il est préférable de modifier les poids des composantes à optimiser pour prendre en compte des critères externes.
Le chapitre 4 présente les résultats de l'évaluation de notre approche et valide son

intérêt. De plus, il évalue et étudie l'influence des paramètres et des traitements linguistiques utilisés. Il est important de noter que ces évaluations valident non seulement l'utilité de la similarité sémantique pour réduire la redondance informationnelle mais aussi l'utilité du clustering sémantique. Nous soulignons aussi que, bien que nous optimisions une couverture en bigrammes de mots, la prise en compte de la similarité au niveau des bigrammes n'est pas aussi efficace que la similarité entre les phrases. Ceci s'explique par le fait que comparer des phrases sémantiquement permet de contourner partiellement le problème de polysémie des mots en considérant leur contexte représenté par la phrase contenant le mot en question.

Concernant notre second axe, qui s'intéresse à la maximisation de la saillance des phrases sélectionnées, nous avons étudié dans le chapitre 5 l'impact de la prise en compte de l'information discursive au sein du même modèle de résumé pour favoriser les informations pertinentes. Nous avons réussi à améliorer le modèle de base en exploitant la hiérarchie attribuée au texte et la nucléarité attribuée aux segments de phrases. Pour cela, nous avons procédé à la pénalisation des éléments jugés peu pertinents selon l'analyse discursive. Dans ce contexte, nous avons pu conclure que pour les données journalistiques, supposer que les nouvelles informations figurent aux premières positions est une baseline forte. Quels que soient les critères utilisés pour détecter la nouveauté, il faut veiller à ne pas pénaliser les éléments favorisés par le critère de position.

Afin d'intégrer les critères sémantique et discursif, nous avons utilisé un système de fusion de listes de classement, ce qui nous a permis d'améliorer encore un peu plus les résultats et de montrer que ces deux dimensions sont complémentaires.

À la suite de ces travaux, nous faisons le constat qu'il est actuellement difficile d'améliorer les performances du résumé extractif multi-document générique. Pour la tâche du résumé mis-à-jour, la marge d'amélioration est cependant plus importante, sans doute parce que le sujet est plus récent. Cette difficulté a plusieurs origines. L'une d'entre elles est méthodologique. Le résumé par extraction est intrinsèquement limité par la contrainte de réutiliser le matériau des documents source. Néanmoins, nos expériences sur les résumés oracle prouvent qu'une marge importante de progression existe encore dans ce cadre. L'autre difficulté tient à la rareté des données disponibles. Les approches par apprentissage sont

performantes dans beaucoup de domaines mais paradoxalement assez peu présentes dans le champ du résumé par extraction à cause de la faiblesse des données d'entraînement. L'autre difficulté principale tient à l'évaluation. Pour des raisons de moyens, celle-ci est principalement réalisée de manière automatique, à l'aide des mesures ROUGE fondées sur le recouvrement de ngrammes. Or les approches les plus performantes, comme les modèles de type ILP, fondent leur optimisation sur le recouvrement de ngrammes et l'on voit maintenant des approches neuronales utilisant explicitement la mesure ROUGE comme critère d'optimisation. On ne peut s'empêcher de penser qu'il y a là une forme de biais qui complique la mise en évidence de l'intérêt potentiel de critères différents.
Dans le même ordre d'idées, nous notons aussi que des améliorations minimes en termes d'évaluation standard ne changent pas forcément le jugement humain sur le résumé produit. Enfin, la plupart des systèmes de l'état de l'art sont dans la même plage de scores même si leurs sorties sont assez différentes. Nous estimons que le manque de diversité des données d'évaluation freine l'avancement sur le résumé automatique. En effet, la majorité des systèmes actuels sont conçus et testés dans le cadre restreint des données de DUC et TAC avec les mêmes paramètres comme la taille du résumé final et la taille des données à résumer. Par ailleurs, le fait d'évaluer des résumés extractifs en les comparant à des résumés abstractifs est un point problématique.

6.2 Perspectives

À partir des problèmes et des limites que nous avons rencontrés au cours de nos travaux, nous avons identifié plusieurs pistes d'amélioration possibles, soit de nos approches, soit du résumé automatique en général.
Premièrement, améliorer le rappel de notre mesure de similarité sémantique permettrait de détecter plus de paires de phrases similaires et par conséquent, plus de redondance informationnelle qui nous échappe à l'heure actuelle.
Deuxièmement, travailler sur la performance des analyseurs RST en termes d'attributions de rôles rhétoriques peut s'avérer très utile dans plusieurs tâches du TAL et particulièrement pour le RA. En effet, les rôles peuvent être exploités afin de simplifier des phrases ou encore être utilisés comme critère de sélection dans

des modèles supervisés.

D'autre part, il y a probablement des améliorations à faire concernant la fusion des résultats des modèles. En effet, parmi les différentes méthodes de fusion de modèles, une seule nous a permis d'améliorer la performance par rapport à chacun des modèles de départ. Étant donné que cette méthode est très basique, nous estimons que travailler sur des méthodes de fusion pourrait améliorer encore la performance.

Sur un autre plan, on ne peut pas négliger la tendance actuelle au résumé abstractif fondé sur les modèles seq2seq. Cependant, ces modèles ont aussi leurs limites, en particulier la nécessité de s'appuyer sur de larges ensembles d'apprentissage, obtenus parfois de façon un peu artificielle. De plus, dans ces modèles, le traitement de textes au-delà de quelques phrases n'est pas vraiment géré. En revanche, une coopération plus étroite entre l'approche extractive et l'approche abstractive est clairement une perspective intéressante.

Plus globalement, il serait très instructif de proposer des jeux de données et des méthodes d'évaluation plus flexibles. Par exemple, il est possible d'annoter les segments des textes source avec des étiquettes indiquant à quel point ils sont qualifiés pour apparaître dans le résumé. De cette façon, il serait plus facile de faire la distinction entre les systèmes en termes de performance dans le sens où si nous disposons de deux résumés très différents des résumés manuels, les méthodes d'évaluation actuelles (ROUGE, PYRAMID, etc) vont pénaliser de la même façon les deux systèmes. Néanmoins, bien que les deux résumés soient loin du résumé idéal, l'un des deux peut être plus proche de l'objectif que l'autre. Comparer le chevauchement du résumé produit avec le résumé idéal est une méthode binaire et peu sensible aux variations de performance quand les textes à comparer sont différents lexicalement. De plus, évaluer des résumés en se fondant sur l'annotation des textes source est plus adapté aux méthodes extractives. Certes, les méthodes abstractives permettent de contourner des problèmes comme l'extraction des phrases longues ou les variations lexicales. Mais il est toujours nécessaire, comme pour les méthodes extractives, d'identifier les éléments importants et de pouvoir évaluer l'aptitude du système à repérer l'information saillante. Évaluer les résumés par rapport à des textes annotés est donc une piste à explorer, susceptible en outre de permettre la production à moindre coût d'un ensemble plus large de données de

référence. Si elle s'avère efficace, elle le sera non seulement pour le résumé extractif mais aussi pour le résumé abstractif.

Annexe A

Annexe : Conception et implémentation

A.1 Logiciels utilisés

Tâche	Bibliothèque/logiciel
Modèle ILP de base Gillick and Favre (2009)	https://code.google.com/archive/p/icsisumm/
Solveur ILP	GLPK https://www.gnu.org/software/glpk/
Algorithme Word2Vec	https://code.google.com/archive/p/word2vec/
Algorithme GloVe	https://nlp.stanford.edu/projects/glove/
Similarité des phrases	gensim.word2vec.n_similarity
Tokenisation en mots	nltk.tokenize.word_tokenize
Tokenisation en phrases	nltk.tokenize.punkt
Clustering de Markov	https://micans.org/mcl/
Évaluation ROUGE	Implémentation originale en Perl https://github.com/kylehg/summarizer/blob/master/rouge/ROUGE-1.5.5.pl commande utilisée : ./ROUGE-1.5.5.pl -e data -a -n 2 -2 4 -m -l 100 -u -c 95 -p 0 -r 1000
Préparation des données pour l'évaluation ROUGE	prepare4rouge.pl http://kavita-ganesan.com/prepare4rouge-script-prepare-rouge-evaluation/#.W7SiixMzYdU
Stemming	nltk.stem.porter.PorterStemmer

Analyseur RST	DPLP https://github.com/jiyfeng/DPLP

A.2 Ressources et données

Ressource	Source
Liste des mots vides	nltk.corpus.stopwords.words('english')
Données TAC	https://tac.nist.gov/tracks/index.html
Plongements lexicaux pré-entraînés	Word2Vec : GoogleNews Word Vectors https://code.google.com/archive/p/word2vec/ GloVe : Common Crawl Word Vectros https://nlp.stanford.edu/projects/glove/ ConceptNet Words Ensemble[1] https://github.com/commonsense/conceptnet-numberbatch

A.3 Temps d'exécution

Les temps d'exécution présentés ci-dessous sont approximatifs pour donner une idée de l'ordre de grandeur des coûts de calcul.

Processus	Temps d'exécution
Similarité sémantique	30 minutes pour le calcul des similarités de 5 millions de paires de phrases
Clustering de phrases	10 secondes pour regrouper 1000 éléments
ILP + pré-traitement	10 secondes pour 48 résumés de 10 documents chacun
Analyse RST	20 minutes pour 480 documents

A.4 Diagrammes de flux de données

Nous présentons à la figure A.1 le diagramme de flux de données de la première contribution décrite dans le chapitre 3.

1. Désormais ConceptNet Numberbatch.

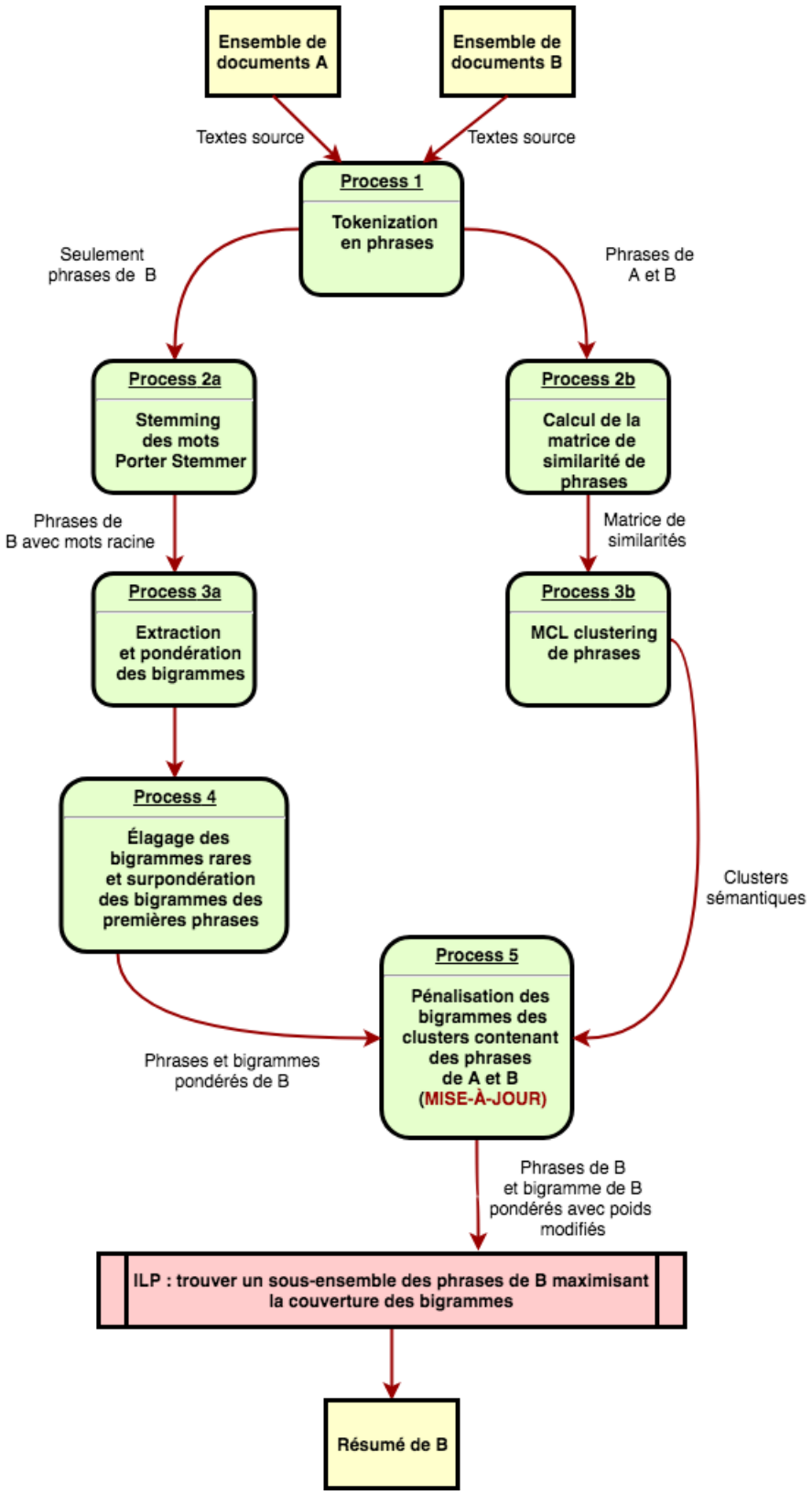

FIGURE A.1 – Diagramme de flux de données de l'intégration de la similarité sémantique dans le modèle ILP

La figure A.2 détaille quant à elle les flux de données et les processus liés à l'intégration de critères discursifs pour le résumé mis-à-jour présentée dans le chapitre 5.

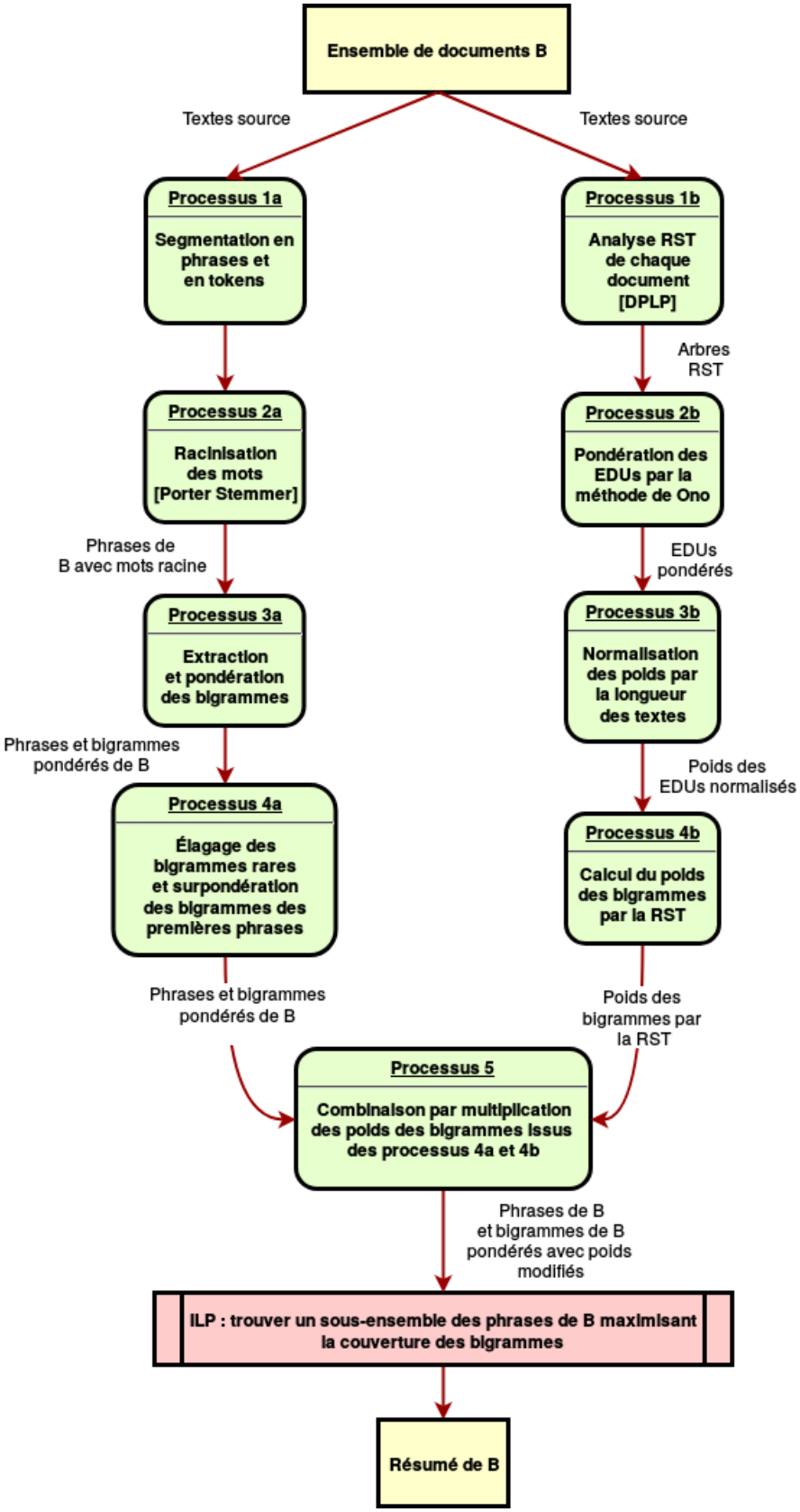

FIGURE A.2 – Diagramme de flux de données de la prise en compte de l'analyse du discours RST dans le modèle ILP

Bibliographie

Arvind Agarwal, Hema Raghavan, Karthik Subbian, Prem Melville, Richard D Lawrence, David C Gondek, and James Fan. Learning to rank for robust question answering. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 833–842. ACM, 2012. (Cité en page 105.)

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. Semeval-2016 task 1 : Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California, June 2016. (Cité en page 80.)

Miguel B Almeida and Andre FT Martins. Fast and robust compressive summarization with dual decomposition and multi-task learning. In *ACL (1)*, pages 196–206, 2013. (Cité en page 36.)

Roxana Angheluta, Rik De Busser, and Marie-Francine Moens. The use of topic segmentation for automatic summarization. In *Proceedings of the ACL-2002 Workshop on Automatic Summarization*, pages 11–12, New Brunswick, NJ, 2002. (Cité en page 23.)

Rachit Arora and Balaraman Ravindran. Latent dirichlet allocation based multi-document summarization. In *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data*, AND'08, pages 91–97, New York, NY, USA, 2008. ISBN 978-1-60558-196-5. doi : 10.1145/1390749.1390764. URL `http://doi.acm.org/10.1145/1390749.1390764`. (Cité en page 23.)

Daniel Bär, Torsten Zesch, and Iryna Gurevych. Composing Measures for Computing Text Similarity. Technical Report TUD-CS-2015-0017, TU Darmstadt, Allemagne, 2015. (Cité en page 13.)

Pinaki Bhaskar. Multi-document summarization using automatic key-phrase extraction. In *RANLP*, pages 22–29, 2013. (Cité en page 36.)

Pinaki Bhaskar and Sivaji Bandyopadhyay. A query focused multi document automatic summarization. In *PACLIC*, pages 545–554, 2010. (Cité en page 31.)

Aurélien Bossard and Emilie Guimier De Neef. Etude de l'impact du regroupement automatique de phrases sur un système de résumé multi-documents. In *huitième Conférence en Recherche d'Information et Applications*, page 8, 2011. (Cité en page 47.)

Florian Boudin and Juan Manuel Torres-Moreno. A maximization-minimization approach for update text summarization. *Recent Advances in Natural Language Processing V : Selected Papers from RANLP 2007*, 309 :143, 2009a. (Cité en page 39.)

Florian Boudin and Juan-Manuel Torres-Moreno. Résumé automatique multi-document et indéndance de la langue : une première évaluation en français. In *Actes de la 16ème conférence sur le Traitement Automatique des Langues Naturelles*, Senlis, France, June 2009b. (Cité en page 19.)

Florian Boudin, Juan-Manuel Torres-Moreno, and Marc El-Béze. Improving update summarization by revisiting the mmr criterion. *arXiv preprint arXiv :1004.3371*, 2010. (Cité en page 40.)

Florian Boudin, Hugo Mougard, and Benoît Favre. Concept-based summarization using integer linear programming : From concept pruning to multiple optimal solutions. In *2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 1914–1918, Lisbon, Portugal, 2015. (Cité en pages 20 et 62.)

Chloé Braud, Barbara Plank, and Anders Søgaard. Multi-view and multi-task training of rst discourse parsers. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*, pages 1903–1913, 2016. (Cité en page 97.)

Chloé Braud, Maximin Coavoux, and Anders Søgaard. Cross-lingual rst discourse parsing. *arXiv preprint arXiv :1701.02946*, 2017. (Cité en page 97.)

Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1) :13–47, 2006. (Cité en page 47.)

Praveen Bysani. Detecting novelty in the context of progressive summarization. In *Proceedings of the NAACL HLT 2010 Student Research Workshop*, pages 13–18. Association for Computational Linguistics, 2010. (Cité en page 42.)

Praveen Bysani, Vijay Bharath Reddy, and Vasudeva Varma. Modeling novelty and feature combination using support vector regression for update summarization. In *Proceedings of ICON-2009 : 7th International Conference on Natural Language Processing*, 2009. (Cité en page 41.)

Luis Adrián Cabrera-Diego and Juan-Manuel Torres-Moreno. Summtriver : A new trivergent model to evaluate summaries automatically without human references. *Data & Knowledge Engineering*, 113 :184–197, 2018. (Cité en page 28.)

Luis Adrián Cabrera-Diego, Juan-Manuel Torres-Moreno, and Barthélémy Durette. Evaluating multiple summaries without human models : A first experiment with a trivergent model. In *International conference on applications of natural language to information systems*, pages 91–101. Springer, 2016. (Cité en page 28.)

Ziqiang Cao, Furu Wei, Sujian Li, Wenjie Li, Ming Zhou, and WANG Houfeng. Learning summary prior representation for extractive summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers)*, volume 2, pages 829–833, 2015. (Cité en page 17.)

Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998. (Cité en page 17.)

Yllias Chali and Mohsin Uddin. Multi-document summarization based on atomic semantic events and their temporal relationships. In *European Conference on Information Retrieval*, pages 366–377. Springer, 2016. (Cité en page 41.)

Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. *arXiv preprint arXiv :1603.07252*, 2016. (Cité en page 17.)

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. Probabilistic frame induction. *arXiv preprint arXiv :1302.4813*, 2013. (Cité en page 11.)

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv :1406.1078*, 2014. (Cité en page 10.)

C. Ravindranath Chowdary and P. Sreenivasa Kumar. An incremental summary generation system. In *Proceedings of the 14th International Conference on Management of Data, December 17-19, 2008, IIT Bombay, Mumbai, India*, pages 83–92, 2008. URL `http://www.cse.iitb.ac.in/~comad/2008/PDFs/18.pdf`. (Cité en page 24.)

John M Conroy, Judith D Schlesinger, Peter A Rankel, and Dianne P O'Leary. Guiding classy toward more responsive summaries. In *TAC*, 2010. (Cité en page 29.)

John M. Conroy, Judith D. Schlesinger, and Jeff Kubina. CLASSY 2011 at TAC : Guided and Multi-lingual Summaries and Evaluation Metrics. In *Proccedings of the Fourth Text Analysis Conference (TAC 2011)*, 2011. (Cité en page 16.)

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759. ACM, 2009. (Cité en page 106.)

C. Dalal and L.G Malik. A survey of extractive and abstractive text summarization techniques. pages 109–110, 2013. (Cité en page 9.)

Arpita Das, Harish Yenala, Manoj Chinnakotla, and Manish Shrivastava. Together we stand : Siamese networks for similar question retrieval. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 378–387, Berlin, Germany, August 2016. (Cité en page 56.)

Jean C de Borda. Mémoire sur les élections au scrutin. 1781. (Cité en page 104.)

Gaël de Chalendar. The LIMA Multilingual Analyzer Made Free : FLOSS Resources Adaptation and Correction. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland*, pages 2932–2937, 2014. URL http://www.lrec-conf.org/proceedings/lrec2014/summaries/362.html. (Cité en page 9.)

Jean-Yves Delort and Enrique Alfonseca. Dualsum : a topic-model based approach for update summarization. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 214–223. Association for Computational Linguistics, 2012. (Cité en pages 43 et 45.)

Anjali R Deshpande and LMRJ Lobo. Text summarization using clustering technique. *International Journal of Engineering Trends and Technology*, 4(8), 2013. (Cité en page 47.)

Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora : exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics (COLING 2004)*, pages 350–356, 2004. (Cité en page 80.)

H. P. Edmundson. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16(2) :264–285, 1969. (Cité en pages 10, 14 et 15.)

Günes Erkan and Dragomir R Radev. Lexpagerank : Prestige in multi-document text summarization. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004a. (Cité en page 37.)

Günes Erkan and Dragomir R Radev. Lexrank : Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22 : 457–479, 2004b. (Cité en page 37.)

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter*

of the Association for Computational Linguistics : Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015, pages 1606–1615, 2015. URL http://aclweb.org/anthology/N/N15/N15-1184.pdf. (Cité en pages 53 et 54.)

Vanessa Wei Feng and Graeme Hirst. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Long Papers-Volume 1*, pages 60–68. Association for Computational Linguistics, 2012. (Cité en page 95.)

Vanessa Wei Feng and Graeme Hirst. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, volume 1, pages 511–521, 2014. (Cité en page 15.)

Olivier Ferret. Early and late combinations of criteria for reranking distributional thesauri. In *53rd Annual Meeting of the Association for Computational Linguistics and 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015), short p aper session*, pages 470–476, Beijing, China, July 2015. URL http://www.aclweb.org/anthology/P15-2077. (Cité en page 102.)

Olivier Ferret, Brigitte Grau, Jean-Luc Minel, and Sylvie Porhiel. Repérage de structures thématiques dans des textes. In *TALN 2001*, pages 163–172, 2001. (Cité en page 15.)

Olivier Ferret, Sana Leila Châar, and Christian Fluhr. Filtrage pour la construction de résumés multi-documents guidée par un profil. *Traitement Automatique des Langues*, 45(1) :65–93, 2004. URL https://hal-cea.archives-ouvertes.fr/cea-00189182. (Cité en page 23.)

Katja Filippova. *Dependency Graph-Based Sentence Fusion and Compression*. PhD thesis, TU Darmstadt, Allemagne, 2010. (Cité en page 11.)

Edward A Fox and Joseph A Shaw. Combination of multiple searches. *NIST special publication SP*, 243, 1994. (Cité en page 106.)

Jorge García Flores, Olivier Ferret, and Gaël de Chalendar. Summarizing through sense concentration and contextual exploration rules : the CHORAL system at TAC 2009. In *Proceedings of the Second Text Analysis Conference (TAC 2009)*, Gaithersburg, Maryland, USA, 2009. (Cité en page 9.)

Johanna Geiss. Creating a gold standard for sentence clustering in multi-document summarization. In *47th Annual Meeting of the Association for Computational Linguistics and 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP 2009), Student Research Workshop*, pages 96–104, Singapore, 2009. (Cité en page 80.)

Pierre-Etienne Genest and Guy Lapalme. Framework for Abstractive Summarization using Text-to-Text Generation. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 64–73, Portland, Oregon, June 2011. (Cité en page 11.)

Pierre-Etienne Genest and Guy Lapalme. Fully Abstractive Approach to Guided Summarization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 354–358, 2012. URL `http://dl.acm.org/citation.cfm?id=2390665.2390745`. (Cité en pages 9 et 11.)

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1602–1613, 2014. (Cité en page 93.)

S.J. Gershman and J.B. Tenenbaum. Phrase similarity in humans and machines. In *Proceedings of the 37th Annual Conference of the Cognitive Science Society*, 2015. (Cité en page 56.)

Dan Gillick and Benoit Favre. A Scalable Global Model for Summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 10–18, Boulder, Colorado, 2009. ISBN 978-1-932432-35-0. URL `http://dl.acm.org/citation.cfm?id=1611638.1611640`. (Cité en pages 4, 19, 33, 45, 46, 59, 95 et 115.)

Daniel Gillick, Benoit Favre, Dilek Hakkani-Tür, Bernd Bohnet, Yang Liu, and Shasha Xie. The icsi/utd summarization system at tac 2009. In *Text Analysis Conference*, 2009. (Cité en pages 59, 61 et 62.)

Umberto Grandi, Andrea Loreggia, Francesca Rossi, and Vijay Saraswat. A borda count for collective sentiment analysis. *Annals of Mathematics and Artificial Intelligence*, 77(3-4) :281–302, 2016. (Cité en page 104.)

Martin Hassel. Exploitation of Named Entities in Automatic Text Summarization for Swedish. In *In Proceedings of NODALIDA 03 - 14 th Nordic Conference on Computational Linguistics*, Reykjavik, Iceland, 2003. (Cité en page 13.)

Marti A. Hearst. TextTiling : Segmenting Text into Multi-paragraph Subtopic Passages. *Computational Linguistics*, 23(1) :33–64, March 1997. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972684.972687. (Cité en page 23.)

Hugo Hernault, Helmut Prendinger, Mitsuru Ishizuka, et al. Hilda : A discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3), 2010. (Cité en page 95.)

Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (HLT-NAACL 2016)*, pages 1367–1377, San Diego, California, 2016. (Cité en pages 49 et 56.)

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. Single-document summarization as a tree knapsack problem. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1515–1520, 2013. (Cité en pages vii, viii, 92, 93 et 94.)

Tsutomu Hirao, Masaaki Nishino, Jun Suzuki, and Masaaki Nagata. Enumeration of Extractive Oracle Summaries. In *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 386–396, Valencia, Spain, 2017. (Cité en pages 10 et 69.)

Kai Hong, John Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. A Repository of State of the Art and Competitive Baseline Summaries for Generic News Summarization. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1608–1616, Reykjavik, Iceland, May 2014. ELRA. ISBN 978-2-9517408-8-4. (Cité en pages 3, 26 et 71.)

Kai Hong, Mitchell Marcus, and Ani Nenkova. System combination for multi-document summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL http://aclweb.org/anthology/D15-1011. (Cité en pages 34, 44, 103 et 104.)

Lei Huang and Yanxiang He. Corrrank : update summarization based on topic correlation analysis. *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 641–648, 2010. (Cité en page 43.)

Kokil Jaidka, Christopher Khoo, and Jin-Cheon Na. Deconstructing human literature reviews – a framework for multi-document summarization. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 125–135, Sofia, Bulgaria, August 2013. URL http://www.aclweb.org/anthology/W13-2116. (Cité en page 14.)

Aditya Jain, Divij Bhatia, and Manish K Thakur. Extractive text summarization using word vector embedding. In *Machine Learning and Data Science (MLDS), 2017 International Conference on*, pages 51–55. IEEE, 2017. (Cité en page 18.)

Yangfeng Ji and Jacob Eisenstein. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, volume 1, pages 13–24, 2014. (Cité en page 96.)

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002. (Cité en page 103.)

Shafiq Joty, Giuseppe Carenini, Raymond Ng, and Yashar Mehdad. Combining Intra- and Multi-sentential Rhetorical Parsing for Document-level Discourse Analysis. In *Proceedings of ACL*, 2013. (Cité en pages 15 et 96.)

K. Kaikhah. Automatic Text Summarization with Neural Networks. In *Proceedings of IEEE International Conference on Intelligent Systems*, volume 1, pages 40–44, June 2004. doi : 10.1109/IS.2004.1344634. (Cité en page 17.)

Rahul Katragadda, Prasad Pingali, and Vasudeva Varma. Sentence position revisited : a robust light-weight update summarization'baseline'algorithm. In *Proceedings of the Third International Workshop on Cross Lingual Information Access : Addressing the Information Need of Multilingual Societies*, pages 46–52. Association for Computational Linguistics, 2009. (Cité en page 37.)

Tom Kenter, Alexey Borisov, and Maarten de Rijke. Siamese cbow : Optimizing word embeddings for sentence representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 941–951, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1089. (Cité en page 56.)

Atif Khan and Naomie Salim. A Review on Abstractive Summarization Methods. *Journal of Theoretical and Applied Information Tech nology*, 59(1) :64–72, 2014. (Cité en page 11.)

Yuta Kikuchi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. Single document summarization based on nested tree structure. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)*, volume 2, pages 315–320, 2014. (Cité en pages viii, 93 et 94.)

Walter Kintsch and Teun A. van Dijk. Toward a model of text comprehension and production. *Psychological Review*, 85(5) :363–394, 1978. (Cité en page 10.)

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015. (Cité en page 56.)

Thomas K Landauer and Susan T Dumais. A solution to plato's problem : The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2) :211, 1997. (Cité en page 52.)

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014. (Cité en pages 55 et 56.)

Huang Lei and He Yanxiang. CorrRank : Update Summarization Based on Topic Correlation Analysis. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence. 6th International Conference on Intelligent Computing (ICIC 2010)*, pages 641–648, Changsha, China, 2010. (Cité en page 71.)

Chen Li, Xian Qian, and Yang Liu. Using supervised bigram-based ILP for extractive summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1004–1013, Sofia, Bulgaria, 2013. URL `http://aclweb.org/anthology/P/P13/P13-1099.pdf`. (Cité en pages 20 et 33.)

Chen Li, Yang Liu, and Lin Zhao. Improving update summarization via supervised ilp and sentence reranking. In *HLT-NAACL*, pages 1317–1322, 2015a. (Cité en pages 39, 45, 61 et 71.)

Chen Li, Yang Liu, and Lin Zhao. Using external resources and joint learning for bigram weighting in ilp-based multi-document summarization. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 778–787, 2015b. (Cité en page 20.)

Jiwei Li, Sujian Li, Xun Wang, Ye Tian, and Baobao Chang. Update summarization using a multi-level hierarchical dirichlet process model. In *COLING 2012*, pages 1603–1618, Mumbai, India, 2012. (Cité en page 45.)

Peng Li, Yinglin Wang, Wei Gao, and Jing Jiang. Generating aspect-oriented multi-document summarization with event-aspect model. In *Proceedings of the*

2011 Conference on Empirical Methods in Natural Language Processing, pages 1137–1146, Edinburgh, Scotland, UK., July 2011a. URL http://www.aclweb.org/anthology/D11-1105. (Cité en pages 20 et 23.)

Qi Li, Tianshi Li, and Baobao Chang. Discourse parsing with attention-based hierarchical neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 362–371, 2016. (Cité en page 97.)

Xuan Li, Liang Du, and Yi-Dong Shen. Graph-based marginal ranking for update summarization. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 486–497. SIAM, 2011b. (Cité en page 38.)

Chin-Yew Lin. Rouge : A package for automatic evaluation of summaries. In *Text Summarization Branches Out : Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. (Cité en pages 26 et 66.)

Chin-Yew Lin and Eduard Hovy. Identifying Topics by Position. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 283–290, Washington, DC, USA, March 1997. doi : 10.3115/974557.974599. URL http://www.aclweb.org/anthology/A97-1042. (Cité en page 14.)

Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing : Volume 1-Volume 1*, pages 343–351. Association for Computational Linguistics, 2009. (Cité en page 96.)

Annie Louis and Ani Nenkova. Automatic summary evaluation without human models. In *Proceedings of the First Text Analysis Conference (TAC 2008)*, 2008. (Cité en page 28.)

Annie Louis and Ani Nenkova. Automatically assessing machine summary content without a gold standard. *Computational. Linguistics*, 39(2) :267–300, June 2013. ISSN 0891-2017. doi : 10.1162/COLIa00123. URL http://dx.doi.org/10.1162/COLIa00123. (Cité en page 28.)

Annie Louis, Aravind Joshi, and Ani Nenkova. Discourse indicators for content selection in summarization. In *Proceedings of the 11th Annual Meeting of the*

Special Interest Group on Discourse and Dialogue, pages 147–156. Association for Computational Linguistics, 2010a. (Cité en page 98.)

Annie Louis, Aravind Joshi, and Ani Nenkova. Discourse indicators for content selection in summarization. In *Proceedings of the SIGDIAL 2010 Conference*, pages 147–156, Tokyo, Japan, September 2010b. URL http://www.aclweb.org/anthology/W/W10/W10-4327. (Cité en page 15.)

Annie P Louis. A bayesian method to incorporate background knowledge during automatic text summarization. Association for Computational Linguistics, 2014. (Cité en pages 35 et 41.)

H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research Development*, 2(2) :159–165, 1958. URL http://courses.ischool.berkeley.edu/i256/f06/papers/luhn58.pdf. (Cité en pages 9 et 12.)

Nitin Madnani and Bonnie J. Dorr. Generating phrasal and sentential paraphrases : A survey of data-driven methods. *Computational Linguistics*, 36(3) : 341–387, 2010. (Cité en page 11.)

Inderjeet Mani. *Automatic Summarization*. John Benjamins Publishing, 2001. (Cité en page 15.)

William C. Mann and Sandra A. Thompson. Rhetorical Structure Theory : Toward a functional theory of text organization. *Text*, 8(3) :243–281, 1988. (Cité en page 15.)

Daniel Marcu. The rhetorical parsing, summarization, and generation of natural language texts. Technical Report CSRG-371, Computer Systems Research Group, University of Toronto, 1997. (Cité en pages 16 et 91.)

Daniel Marcu. Improving summarization through rhetorical parsing tuning. In *Proceedings of The Sixth Workshop on Very Large Corpora*, pages 206–215, Montreal, Canada, August 1998a. (Cité en pages 4 et 16.)

Daniel Marcu. To build text summaries of high quality, nuclearity is not sufficient. In *Working Notes of the AAAI-98 Spring Symposium on Intelligent Text Summarization*, pages 1–8, 1998b. (Cité en pages 90 et 92.)

Marie Jean Antoine marquis de Condorcet. *Essai sur l'application de l'analyse a la probabilite des decisions : rendues a la pluralite de voix*. De l'Imprimerie royale, 1785. (Cité en page 105.)

André FT Martins and Noah A Smith. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 1–9. Association for Computational Linguistics, 2009. (Cité en page 34.)

Paul Mc Kevitt, Derek Partridge, and Yorick Wilks. Approaches to natural language discourse processing. *Artificial Intelligence Review*, 6(4) :333–364, 1992. (Cité en page 15.)

Richard McCreadie, Craig Macdonald, and Iadh Ounis. Incremental update summarization : Adaptive sentence selection based on prevalence and novelty. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 301–310, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2598-1. doi : 10.1145/2661829.2661951. URL http://doi.acm.org/10.1145/2661829.2661951. (Cité en page 24.)

Ryan McDonald. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 557–564, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-71494-1. URL http://dl.acm.org/citation.cfm?id=1763653.1763720. (Cité en page 19.)

Kathleen R. McKeown, Judith L. Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. Towards Multidocument Summarization by Reformulation : Progress and Prospects. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, pages 453–460, Menlo Park, CA, USA, 1999. ISBN 0-262-51106-1. (Cité en page 14.)

Parth Mehta and Prasenjit Majumder. Effective aggregation of various summari-

zation techniques. *Information Processing & Management*, 54(2) :145–158, 2018. (Cité en page 103.)

Rada Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, 2004. URL http://aclweb.org/anthology/P04-3020. (Cité en pages 13 et 19.)

Rada Mihalcea and Paul Tarau. Textrank : Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004. (Cité en page 37.)

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. (Cité en pages 51 et 72.)

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Counter-fitting word vectors to linguistic constraints. In *HLT-NAACL*, 2016. URL http://arxiv.org/abs/1603.00892. (Cité en page 54.)

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner : A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, pages 3075–3081, 2017. (Cité en page 18.)

Ani Nenkova and Kathleen McKeown. *Mining Text Data*, chapter A Survey of Text Summarization Techniques. Springer, 2012. (Cité en page 8.)

Ani Nenkova and Rebecca J. Passonneau. Evaluating content selection in summarization : The pyramid method. In *HLT-NAACL*, pages 145–152, 2004. URL http://acl.ldc.upenn.edu/hlt-naacl2004/main/pdf/91Paper.pdf. (Cité en page 26.)

Joel Larocca Neto, AlexandreD. Santos, CelsoA.A. Kaestner, and AlexA. Freitas. Generating text summaries through the relative importance of topics. In

MariaCarolina Monard and JaimeSimÃ£o Sichman, editors, *Advances in Artificial Intelligence*, volume 1952 of *Lecture Notes in Computer Science*, pages 300–309. Springer Berlin Heidelberg, 2000. ISBN 978-3-540-41276-2. doi : 10.1007/3-540-44399-131. URL http://dx.doi.org/10.1007/3-540-44399-131. (Cité en page 23.)

Joel Larocca Neto, Alex A. Freitas, and Celso A. A. Kaestner. Automatic text summarization using a machine learning approach. In *Advances in Artificial Intelligence*, pages 205–215. Springer Berlin Heidelberg, 2003. (Cité en pages 13, 15 et 16.)

Fernando Antônio Asevedo Nóbrega and Thiago AS Pardo. Improving content selection for update summarization with subtopic-enriched sentence ranking functions. *Int. J. Comput. Linguistics Appl.*, 7(2) :111–128, 2016. (Cité en page 40.)

Kenji Ono, Kazuo Sumita, and Seiji Miike. Abstract generation based on rhetorical structure extraction. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 344–348. Association for Computational Linguistics, 1994. (Cité en pages 89 et 99.)

Mick O'Donnell. Variable-length on-line document generation. In *Proceedings of the 6th European Workshop on Natural Language Generation*, pages 82–91, 1997. (Cité en page 90.)

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking : Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. URL http://ilpubs.stanford.edu:8090/422/. (Cité en page 18.)

Rebecca J. Passonneau, Emily Chen, Weiwei Guo, and Dolores Perin. Automated pyramid scoring of summaries using distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 143–147, Sofia, Bulgaria, August 2013. URL http://www.aclweb.org/anthology/P13-2026. (Cité en page 27.)

Yulong Pei, Wenpeng Yin, Qifeng Fan, et al. A supervised aggregation framework

for multi-document summarization. *Proceedings of COLING 2012*, pages 2225–2242, 2012. (Cité en page 103.)

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove : Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. (Cité en pages 51 et 72.)

Dragomir R. Radev, Simone Teufel, Horacio Saggion, Wai Lam, John Blitzer, Hong Qi, Arda Çelebi, Danyu Liu, and Elliott Drabek. Evaluation challenges in large-scale document summarization. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 375–382, Sapporo, Japan, July 2003. doi : 10.3115/1075096.1075144. URL http://www.aclweb.org/anthology/P03-1048. (Cité en page 28.)

Dragomir R. Radev, Hongyan Jing, Malgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6) :919–938, November 2004. ISSN 0306-4573. doi : 10.1016/j.ipm.2003.10.006. URL http://dx.doi.org/10.1016/j.ipm.2003.10.006. (Cité en pages 13, 22 et 47.)

Pengjie Ren, Furu Wei, Zhumin Chen, Jun Ma, and Ming Zhou. A redundancy-aware sentence regression framework for extractive summarization. In *COLING*, pages 33–43. ACL, 2016. (Cité en page 34.)

Pengjie Ren, Zhumin Chen, Zhaochun Ren, Furu Wei, Jun Ma, and Maarten de Rijke. Leveraging contextual sentence relations for extractive summarization using a neural attention model. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 95–104. ACM, 2017. (Cité en page 18.)

Cody Rioux, Sadid A. Hasan, and Yllias Chali. Fear the reaper : A system for automatic multi-document summarization with reinforcement learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 681–690, Doha, Qatar, October 2014. URL http://www.aclweb.org/anthology/D14-1075. (Cité en page 35.)

Vladimir Robles, Antonio Molina, and Paolo Rosso. Borda-based voting schemes for semantic role labeling. In *International Conference on Text, Speech and Dialogue*, pages 189–196. Springer, 2010. (Cité en page 104.)

Douglas L. T. Rohde, Laura M. Gonnerman, and David C. Plaut. An improved model of semantic similarity based on lexical co-occurence. *COMMUNICATIONS OF THE ACM*, 8 :627–633, 2006. (Cité en page 52.)

Antti-Veikko Rosti, Spyros Matsoukas, and Richard Schwartz. Improved word-level system combination for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 312–319, 2007. (Cité en page 102.)

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv :1509.00685*, 2015. (Cité en page 36.)

Seonggi Ryang and Takeshi Abekawa. Framework of automatic text summarization using reinforcement learning. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 256–265. Association for Computational Linguistics, 2012. (Cité en page 35.)

Horacio Saggion and Guy Lapalme. Generating Indicative-Informative Summaries with SumUM. *Computational Linguistics*, 28(4) :497–526, jan 2002. (Cité en page 9.)

Horacio Saggion, Juan-Manuel Torres-Moreno, Iria da Cunha, Eric SanJuan, and Patricia Velazquez Morales. Multilingual summarization evaluation without human models. In *23rd COLING International Conference on Computational Linguistics (Posters)*, pages 1059–1067, 2010. (Cité en page 28.)

Gerard Salton, Amit Singhal, Chris Buckley, and Mandar Mitra. Automatic text decomposition using text segments and text themes. In *Proceedings of the the seventh ACM conference on Hypertext*, pages 53–65. ACM, 1996. (Cité en page 11.)

Claude Sammut and Geoffrey I. Webb. *Encyclopedia of Machine Learning*. Springer US, 2010. (Cité en page 13.)

Barry Schiffman, Ani Nenkova, and Kathleen McKeown. Experiments in multidocument summarization. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 52–58, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1289189.1289254. (Cité en page 15.)

Natalie Schluter. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2, Short Papers*, pages 41–45, Valencia, Spain, April 2017. Association for Computational Linguistics. (Cité en pages 10 et 69.)

Chao Shen and Tao Li. Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 984–992. Association for Computational Linguistics, 2010. (Cité en page 31.)

Karen Spärck Jones. Automatic summarising : The state of the art. *Inf. Process. Manage.*, 43(6) :1449–1481, November 2007. ISSN 0306-4573. doi : 10.1016/j.ipm.2007.03.009. URL http://dx.doi.org/10.1016/j.ipm.2007.03.009. (Cité en page 8.)

Robert Speer and Joshua Chin. An ensemble method to produce high-quality word embeddings, 2016. URL http://arxiv.org/abs/1604.01692. cite arxiv :1604.01692Comment : 12 pages, 3 figures. (Cité en pages vii, 54, 55 et 72.)

Sandeep Sripada and Jagadeesh Jagarlamudi. Summarization approaches based on document probability distributions. In *PACLIC*, pages 521–529, 2009. (Cité en page 32.)

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. Dls@cu : Sentence similarity from word alignment. In *8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 241–246, Dublin, Ireland, 2014. (Cité en page 49.)

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. (Cité en page 10.)

Hiroya Takamura and Manabu Okumura. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 781–789. Association for Computational Linguistics, 2009. (Cité en page 32.)

Min-Yen Kan Chew-Lim Tan. Swing : Exploiting category-specific information for guided summarization. 2011. (Cité en page 13.)

Juan-Manuel Torres-Moreno. Trivergence of probability distributions, at glance. *arXiv preprint arXiv :1506.06205*, 2015. (Cité en page 28.)

Juan-Manuel Torres-Moreno, Horacio Saggion, Iria da Cunha, Eric SanJuan, and Patricia Velázquez-Morales. Summary evaluation with and without references. *Polibits*, (42) :13–20, 2010. (Cité en page 28.)

Diana Trandabâţ. Using semantic roles to improve summaries. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 164–169, Nancy, France, September 2011. URL `http://www.aclweb.org/anthology/W11-2822`. (Cité en page 12.)

Diana Trandabăţ. Using semantic roles to improve summaries. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 164–169. Association for Computational Linguistics, 2011. (Cité en page 13.)

Vinícius Rodrigues Uzêda, Thiago Alexandre Salgueiro Pardo, and Maria Das Graças Volpe Nunes. A comprehensive comparative evaluation of rst-based summarization methods. *ACM Transactions on Speech and Language Processing (TSLP)*, 6(4) :4, 2010. (Cité en page 99.)

Stijn van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000. (Cité en page 58.)

Hans Van Halteren and Simone Teufel. Examining the consensus between human summaries : initial experiments with factoid analysis. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5*, pages 57–64. Association for Computational Linguistics, 2003. (Cité en page 27.)

Xiaojun Wan. Update summarization based on co-ranking with constraints. In *Proceedings of COLING 2012 (Posters)*, pages 1291–1300, Mumbai, India, December 2012. URL `http://www.aclweb.org/anthology/C12-2126`. (Cité en page 45.)

Xiaojun Wan and Jianwu Yang. Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306. ACM, 2008. (Cité en page 47.)

Dingding Wang and Tao Li. Weighted consensus multi-document summarization. *Information Processing & Management*, 48(3) :513–523, 2012. (Cité en pages 102 et 103.)

Dingding Wang, Sahar Sohangir, and Tao Li. Update summarization using semi-supervised learning based on hellinger distance. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1907–1910. ACM, 2015. (Cité en page 38.)

Hongling Wang and Guodong Zhou. Toward a unified framework for standard and update multi-document summarization. *ACM Transactions on Asian Language Information Processing (TALIP)*, 11(2) :1–18, June 2012. ISSN 1530-0226. (Cité en page 71.)

Li Wenjie, Wei Furu, Lu Qin, and He Yanxiang. Pnr 2 : ranking sentences with positive and negative reinforcement for query-oriented update summarization. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 489–496. Association for Computational Linguistics, 2008. (Cité en page 37.)

Lyndon White, Roberto Togneri, Wei Liu, and Mohammed Bennamoun. How well sentence embeddings capture meaning. In *Proceedings of the 20th Australasian*

Document Computing Symposium (ADCS'15), pages 1–8, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-4040-3. (Cité en page 56.)

Florian Wolf and Edward Gibson. Representing Discourse Coherence : A Corpus-Based Study. *Computational Linguistics*, 31(2) :249–287, 2005. (Cité en page 15.)

Kristian Woodsend and Mirella Lapata. Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 233–243, Jeju Island, Korea, July 2012. URL `http://www.aclweb.org/anthology/D12-1022`. (Cité en pages 20 et 33.)

Guo-Hua Wu and Yu-Tian Guo. An enhanced lsa-based approach for update summarization. In *Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2015 12th International Computer Conference on*, pages 493–497. IEEE, 2015. (Cité en page 42.)

Ou Wu, Haiqiang Zuo, Mingliang Zhu, Weiming Hu, Jun Gao, and Hanzi Wang. Rank aggregation based text feature selection. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 165–172. IEEE, 2009. (Cité en page 105.)

Tan Xu, Paul McNamee, and Douglas W. Oard. HLTCOE Submission at TREC 2013 : Temporal Summarization. In *The Twenty-Second Text REtrieval Conference Proceedings*, 2013. (Cité en page 24.)

Zhen Yang, Fei YAO, Huayang SUN, Yun ZHAO, Yingxu LAI, and Kefeng FAN. BJUT at TREC 2013 Temporal Summarization Track. In *The Twenty-Second Text REtrieval Conference Proceedings*, 2013. (Cité en page 24.)

Yong Zhang, Meng Joo Er, Rui Zhao, and Mahardhika Pratama. Multiview convolutional neural networks for multidocument extractive summarization. *IEEE transactions on cybernetics*, 47(10) :3230–3242, 2017. (Cité en page 18.)

Denny Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf. Ranking on data manifolds. In *Advances in neural information processing systems*, pages 169–176, 2004. (Cité en page 38.)

Markus Zopf, Eneldo Loza Mencía, and Johannes Fürnkranz. Sequential clustering and contextual importance measures for incremental update summarization. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*, pages 1071–1082, 2016. (Cité en page 46.)

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Résumé automatique multi-document et dynamique

Mots clés : similarité sémantique, regroupement, ILP, analyse discursive

Résumé : Cette thèse s'intéresse au Résumé Automatique de texte et plus particulièrement au résumé mis-à-jour. Cette problématique de recherche vise à produire un résumé différentiel d'un ensemble de nouveaux documents par rapport à un ensemble de documents supposés connus. Elle intègre ainsi dans la problématique du résumé à la fois la question de la dimension temporelle de l'information et celle de l'historique de l'utilisateur. Dans ce contexte, le travail présenté s'inscrit dans les approches par extraction fondées sur une optimisation linéaire en nombres entiers (ILP) et s'articule autour de deux axes principaux : la détection de la redondance des informations sélectionnées et la maximisation de leur saillance. Pour le premier axe, nous nous sommes plus particulièrement intéressés à l'exploitation des similarités inter-phrastiques pour détecter, par la définition d'une méthode de regroupement sémantique de phrases, les redondances entre les informations des nouveaux documents et celles présentes dans les documents déjà connus. Concernant notre second axe, nous avons étudié l'impact de la prise en compte de la structure discursive des documents, dans le cadre de la Théorie de la Structure Rhétorique (RS), pour favoriser la sélection des informations considérées comme les plus importantes. L'intérêt des méthodes ainsi définies a été démontré dans le cadre d'évaluations menées sur les données des campagnes TAC et DUC. Enfin, l'intégration de ces critères sémantique et discursif au travers d'un mécanisme de fusion tardive a permis de montrer dans le même cadre la complémentarité de ces deux axes et le bénéfice de leur combinaison.

Title : Multi-document update-summarization

Keywords : semantic similarity, clustering, ILP, discourse analysis

Abstract : This thesis focuses on text Automatic Summarization and particularly on Update Summarization. This research problem aims to produce a differential summary of a set of new documents with regard to a set of old documents assumed to be known. It thus adds two issues to the task of generic automatic summarization: the temporal dimension of the information and the history of the user. In this context, the work presented here is based on an extractive approach using integer linear programming (ILP) and is organized around two main axes: the redundancy detection between the selected information and the user history and the maximization of their saliency . For the first axis, we were particularly interested in the exploitation of inter-sentence similarities to detect the redundancies between the information of the new documents and those present in the already known ones, by defining a method of semantic clustering of sentences. Concerning our second axis, we studied the impact of taking into account the discursive structure of documents, in the context of the Rhetorical Structure Theory (RST), to favor the selection of information considered as the most important. The benefit of the methods thus defined has been demonstrated in the context of evaluations carried out on the data of TAC and DUC campaigns. Finally, the integration of these semantic and discursive criteria through a delayed fusion mechanism has proved the complementarity of these two axes and the benefit of their combination.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France