TAL
Traitement Automatique des Langues
Synthèse
Analyse du discours

Rémi Cadène, Niki Rohani

3 mai 2015

Table des matières

1 Introduction **3**

2 Extraction non supervisée de relations discursives et sémantiques lexicales (Axe Rémi Cadène) **3**
2.1 Introduction à l'analyse de discours 3
2.2 Framework RST, Corpus, Parser Hilda 4
2.2.1 Framework RST 4
2.2.2 Corpus 5
2.2.3 Hilda basé sur SVM 5
2.3 Identification de relations discursives implicites 6
2.3.1 Présentation de la problématique 6
2.3.2 Données 7
2.3.3 Modèle et jeu de traits 7
2.3.4 Expériences 8
2.4 Identification de relations sémantiques lexicales 9
2.4.1 Présentation de la problématique 9
2.4.2 Détail de la base de connaissances construite 10
2.4.3 Mesures de l'association sémantique des verbes 10
2.4.4 Description des méthodes d'évaluation 11

3 Utilisation d'un RST-DT dans le résumé de texte. (Axe Niki Rohani) **12**
3.1 De RST-DT à DEP-DT. 12
3.1.1 Definition d'un DEP-DT 13
3.1.2 Algorithme 13
3.2 Modélisation du problème de résumé sous forme d'un Tree Knapsack Problem. 14
3.2.1 Définition d'un TKP 14
3.2.2 Optimisation, ILP et formules 14
3.3 Expérience 15
3.3.1 ROUGE, un moyen de noter les résumés 15
3.3.2 Première expérience 15
3.4 Améliorations Possibles 16
3.4.1 Utilisation d'un DEP-DT directement généré. 16
3.4.2 Amélioration de la fonction coût sur TKP. 17

4 Conclusion **19**

1 Introduction

Cette synthèse a pour objectif d'aborder quelques techniques de résumés de textes élaborées par des chercheurs dans le domaine de l'analyse de discours. Dans un premier temps, il s'agira de présenter les principaux objets qui ont servi à l'étude du discours, notamment la présentation des relations discursives et des relations sémantiques lexicales. Nous nous intéresserons ensuite à une modélisation de texte sous la forme d'un arbre de relations rhétoriques Rhetorical Structure Discourse Tree (RST-DT), pour définir ensuite comment représenter une unité de texte dans un arbre de ce type Elementary Discourse Unit (EDU). Une fois les bases expliquées, nous allons voir comment utiliser l'analyse de discours pour le résumé de texte, en étudiant deux méthodes proposées par [6] et [14]. Ces méthodes ont recours aux RST-DT et à d'autres structures comme les Dependency based Discourse Tree (DEP-DT). Historiquement, la méthode [14] se fonde sur [6]. Nous présenterons donc celle de [6] en premier. De plus, nous présenterons une méthode pour évaluer les résumés de textes : la méthode ROUGE.

2 Extraction non supervisée de relations discursives et sémantiques lexicales (Axe Rémi Cadène)

2.1 Introduction à l'analyse de discours

Les discours, documents ou textes, ne sont pas seulement des collections de phrases, mais possèdent une structure. L'identification des relations de discours, parfois appelées relations rhétoriques, est fondamentale dans beaucoup d'applications TAL comme les questions-réponses (Chai and Jin, 2004) [3] ou les générations de dialogues (Prendinger et al., 2007) [12]. Ces relations représentent de riches liens sémantiques, comme la causalité, le contraste ou la continuation, entre des segments de textes de tailles variables appelés Unités Elémentaires de Discours (EDUs). Leur taille est communément déterminée par l'utilisation d'un groupe verbal. C'est en effet pour cela que le second article que nous avons étudié ne s'intéresse uniquement qu'à identifier les relations entre les verbes des groupes verbaux des différents EDUs à l'aide de connecteurs discursif (mais, car, etc.). Les chercheurs ont appelé cette approche "extraction non supervisée de relations sémantiques lexicales" [1].

Outre la difficulté de trouver des méthodes d'analyse de discours adaptées à la langue française, les chercheurs rencontrent les problèmes de l'ambiguïté de certaines relations en fonction des connecteurs utilisés, mais aussi plus généralement l'absence de ces connecteurs. En effet, les relations rhétoriques sont fréquemment implicites, c'est à dire sans connecteur, et donc difficilement identifiables par un humain ce qui rend l'annotation de corpus fastidieuse. Ainsi le premier article étudié [4] tente de résoudre ce dernier problème en identifiant le type des relations discursives implicites en utilisant divers trais sur les EDUs. Ces derniers sont issus des corpus annotés manuellement suivant le framework RST, décrit

dans le premier axe, de la ressource française ANNODIS.

Même si ces deux articles partagent le même thème et souvent les mêmes difficultés, leur approche et leur problématique diffèrent fortement ce qui nous empêche de les comparer avec précision. Ainsi, nous avons présenté dans un premier axe les corpus généralement utilisés par les articles du domaine tout en expliquant la méthode de parsing du programme Hilda. Puis, dans nos deux derniers axes, nous avons décris pour chacun des articles lus les méthodes d'analyse utilisées, les difficultés rencontrées et les méthodes d'évaluations utilisées.

2.2 Framework RST, Corpus, Parser Hilda

2.2.1 Framework RST

Le framework Rhetorical Structure Theory (RST), proposé par Mann & Thompson (1988) [10], est l'une des théories de l'organisation du discours les plus utilisées pour créer des outils segmentant les éléments de discours. Par exemple, la phrase suivante, issues du célèbre corpus anglais RST Discourse Treebank (RST-DT par Carlson, Marcu and al., 2003) [2], peut être segmentée en EDUs comme suit dans la Figure 1.

Farm lending was enacted to correct this problem by providing a reliable flow of lendable funds.
1er EDU. Farm lending was enacted
2nd EDU. to correct this problem
3ème EDU. by providing a reliable flow of lendable funds.

FIGURE 1: Segmentation d'une phrase en EDUs.

Puis, les EDUs obtenus sont mis en relation en utilisant des relations de discours pré-définies. Il existe deux types de relations de discours dans RST : hypotactic (mononucléaire) et paratactic (multi-nucléaire). Par exemple, les relations de Condition, Background, Circumstance, Elaboration et Purpose sont hypotactic, tandis que les relations de Contrast, Disjunction, Sequence, et Topic-Comment, sont paratactic. La figure 2 montre deux types de relations de discours avec RST.

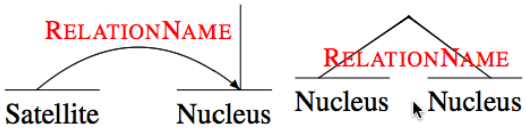

FIGURE 2: Les deux types de relations dans RST. A gauche : mononucléaire. A droite : multi-nucléaire.

Le but final d'un parser est de produire une structure arborescente pour représenter la façon dont les EDUs sont liés. Ainsi la figure 3 montre la représentation arborescente du précédent exemple dans le framework RST.

2.2.2 Corpus

RST-DT

Comme mentionné précédemment, RST-DT est un large corpus de documents annoté selon le framework RST. Il a été publié en 2002 et contient 385 articles de longueur variable du Wall Street Journal. Tandis que le framework RST original proposait 24 types de relations de discours, cette liste a évolué afin de s'adapter aux différents types d'application. De ce fait, dans RST-DT, les relations entre EDUs utilisent un ensemble de 78 relations rhétoriques (53 mononucléaires et 25 multi-nucléaires) qui augmente pouvoir expressif des arbres RST. Celui ci a notamment été utilisé pour entraîné et évalué différents algorithmes de segmentation, dont la parser Hilda.

ANNODIS

En ce qui concerne la langue française, le corpus ANNODIS est souvent utilisé pour l'analyse de discours. Il est issu de quatre sources : Est Républicain (39 articles, 10000 mots), Wikipédia (30 articles + 30 extraits, 242000 mots), Actes du Congrès Mondial de Linguistiques Française 2008 (25 articles, 169000 mots), Rapports de l'Institut Français de Relations Internationales (32 rapports, 266000 mots). De la même façon que RST-DT, il introduit de nouveaux concepts et des relations rhétoriques différentes adaptées à la langue française. Dans le contexte du framework RST, ce corpus est composé de 3188 EDUs et de 1395 Unités Complexes de Discours (CDUs) reliées part 3355 relations de discours typés (e.g. contraste, élaboration, résultat, attribution, etc.). Les CDUs sont en fait des racines de sous arbres ayant pour noeuds des EDUs ou CDUs comme illustré dans la Figure 3.

[Principes de la sélection naturelle.]_1 [La théorie de la sélection naturelle [telle qu'elle a été initialement décrite par Charles Darwin,]_2 repose sur trois principes :]_3 [1. le principe de variation]_4 [2. le principe d'adaptation]_5 [3. le principe d'hérédité]_6

FIGURE 3: Exemple de graphe discursif. Les nœuds correspondent aux unités discursives : les EDU représentées par leur numérotation et les CDU par un nœud étiqueté π_n. Les arêtes avec flèches représentent les relations rhétoriques, les arêtes en pointillé sans flèches représentent l'inclusion d'EDUs dans un CDU. elab=. = Élaboration, e-elab = Élaboration d'entité, C. = Continuation

2.2.3 Hilda basé sur SVM

Le parser HILDA applique à un document la séquence de traitements de la figure 4 :

1. Tout d'abord, le texte est segmenté en EDUs.

2. Ensuite, l'étape de labélisation des relations évalue l'appartenance d'un certain type de relations entre des EDUs consécutifs avec un maximum de vraisemblance. Les deux EDUs les plus plus probablement connectés par une relation rhétorique sont alors fusionnés en une structure rhétorique arborescente de deux EDUs.
3. Puis, de façon itérative, l'étape de labélisation est à nouveau appliquée afin de re-évalue quelles relations sont les plus vraisemblables entre deux structures rhétorique arborescente de n'importe quelle taille en incluant une structure atomique composée d'un seul EDU.
4. Cette procédure est répétée jusqu'à ce que toutes les structures réthorique arborescente aient été fusionnées.

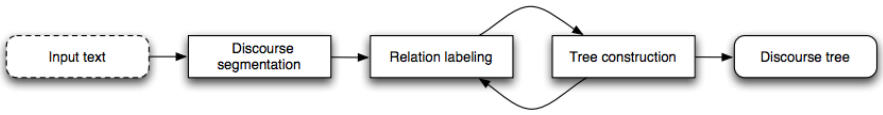

FIGURE 4: Les différentes séquences du parser Hilda.

Nous notons tout de même que dans RST, les relations multi-nucléaires peuvent prendre un nombre d'arguments arbitraire, comme par exemple la relation LIST. Dans un but de compatibilité avec l'approche de classification utilisant SVM, ces arbres de relations à n-arités sont transformées en arbre binaires. Cette conversion peut être réalisée trivialement en combinant les arguments des relations multi-nucléaires de façon consécutive, comme dans la figure 4.

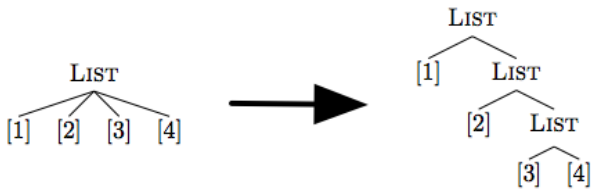

FIGURE 5: Binarisation des relations réthoriques multi-nucléaires.

2.3 Identification de relations discursives implicites

2.3.1 Présentation de la problématique

Il existe deux types de relations discursives différenciées par leur difficulté d'identification et non par le type de relations. Les relations dites explicites sont facilement identifiables grâce à des mots clés comme mais, car ou cela dit. Les relations dites implicites sont, quant à elles, plus difficilement identifiables, car elles ne comportent pas de mots clés. En effet, les chercheurs rapportent une

exactitude de 94% de bonnes classifications des relations explicites en utilisant les méthodes de l'état de l'art, et 40% seulement pour les relations implicites. L'article que nous avons étudié se restreint à l'identification de quatre relations : *contrast*, *result*, *continuation* et *explanation*.

{[Juliette est tombée,] car [Marion l'a poussée]}_{*explanation*}
{[Juliette est tombée.] [Marion l'a poussée]}_{*explanation*}

FIGURE 6: La première relation discursive de type *explanation* est explicite, tandis que la deuxième est implicite.

2.3.2 Données

Deux corpus ont été utilisé lors de cette étude. Le premier est celui d'Annodis qui a été traité en utilisant le LexConn, lexique des connecteurs discursifs du français développé par (Roze, 2010) [13], de manière à identifier tout connecteur, donc de ne récupérer que des exemples implicites. La distribution des exemples par relation est résumée dans le tableau ci dessous.

Relation	Exemples explicites	Exemples implicites	Total
contrast	100	42	142
result	52	110	162
continuation	404	70272	676
explanation	58	70	128
all	614	494	1108

TABLE 1: Corpus Annodis : nombre d'exemples explicites et implicites par relation

Le deuxième est un corpus artificiel qui a été généré par les chercheurs afin de tenter de résoudre le problème du faible nombre de relations implicites annotées. Ces dernières sont obtenues à partir d'articles de l'Est Républicain (9M de phrases) en utilisant les mêmes traitements que pour le corpus Annodis, c'est à dire en supprimant certains mots clés permettant d'identifier des relations discursives explicites afin d'obtenir uniquement des relations implicites. Cette méthode simple permet de générer rapidement de gros volumes de données : au total, les chercheurs ont pu extraire 392 260 exemples (voir tableau 2).

2.3.3 Modèle et jeu de traits

Le modèle utilisé pour classifier les relations implicites est un modèle discriminant par régression logistique (ou maximum d'entropie), car il est beaucoup utilisé pour différents problèmes de TAL, a été implanté dans différentes librairies librement disponibles et donne de bonnes performances. L'intérêt de ce genre de modèles, par rapport aux génératifs, est de permettre l'ajout de nombreux

Relation	Disponible	Entraînement	Développement	Test	Total
contrast	252 793	23 409	2 926	2 926	29 261
result	50 297	23 409	2 926	2 926	29 261
continuation	29 261	23 409	2 926	2 926	29 261
explanation	59 909	23 409	2 926	2 926	29 261
all	392 260	93 636	11 704	11 704	117 044

TABLE 2: Corpus artificiel : nombre d'exemples par relation

descripteurs potentiellement redondants sans faire d'hypothèses d'indépendance.

Certains traits sont calculés pour chaque argument :

1. Indice de complexité syntaxique : nombre de syntagmes nominaux, verbaux, prépositionnels, adjectivaux, adverbiaux (valeur continue)
2. Information sur la tête d'un argument : lemme d'éléments négatifs (valeur nominale), information temporelle/aspectuelle (valeurs continue et nominale), informations sur les dépendants de la tête (valeurs booléenne et nominale) et informations morphologiques (valeur nominale)

D'autres traits portent sur la paire d'arguments :

1. Trait de position : si l'exemple est inter ou intra-phrastique (valeur booléenne)
2. Indice de continuité thématique (valeur continue)
3. Information sur les têtes des arguments (valeurs booléennes)

2.3.4 Expériences

Modèles de base

Les expériences ont été réalisées avec l'implémentation de l'algorithme de la librairie MegaM en multiclasse avec un au maximum 100 itérations. Une validation croisée en 10 sous-ensembles sur le corpus des données manuelles équilibré à 70 exemples maximum par relation. Les performances sont données en termes d'exactitude globale sur l'ensemble des relations, des scores ventilés de F1 par relation sont également fournis.
Dans un premier temps deux modèles ont été construit par les chercheurs, l'un à partir des seules données manuelles (ManOnly, 252 exemples), l'autre des seules données artificielles (AutoOnly, 93 636 exemples d'entraînement). Le ManOnly obtient une exactitude de 39.7, tandis que l'AutoOnly obtient une exactitude de 47.8 sur les données artificielles, mais de 23.0 sur les données manuelles.

Modèles avec combinaisons de données

De nombreuses méthodes de combinaisons des données ont été testé, comme l'union simple des corpus d'entraînement manuels et artificiels (Union), ou comme l'ajout de sous-ensembles aléatoires des données artificielles (AutoSub)

	ManOnly	AutoOnly	
Données de test	Manuelles	Manuelles	Artificielles
Exactitude	39.7	23.0	47.8
contrast	13.3	23.2	38.3
result	49.0	15.7	57.4
continuation	39.7	32.1	54.3
explanation	43.8	22.4	37.5

TABLE 3: Modèles de base, exactitude du système et f-mesure par relaiton

et pondération des exemples manuels (ManW). Les expériences où la prise en compte des données artificielles passe par l'ajout de traits donnent les meilleurs résultats avec une exactitude de 42.9 pour le modèle qui intègre les prédictions du modèle artificiel comme descripteur (AddPred). Le second modèle, qui exploite en plus les probabilités (AddProb) mène quant à lui à une légère diminution ce qui suggère que les traits de probabilité dégradent les performances.

Modèles avec sélection automatique d'exemples

Les expériences précédentes ont montré, selon les chercheurs, que l'ajout des données artificielles donne le plus souvent lieu à des gains de performance qui restent modestes, voire non significatifs, car de nombreux exemples artificiels amènent du bruit dans le modèle. Le but de cette nouvelle méthode est de sélectionner les exemples artificiels les plus informatifs et qui complémentent le mieux les données manuelles. L'idée est de conserver les exemples prédits avec une probabilité supérieur à un seuil. Ainsi, les meilleurs systèmes sont obtenus entre les seuils 60 et 70 et atteignent des performances de 45.6 (ManW, paramètre 900, seuil 65) et 44.0 (AddProb, seuil 65).

2.4 Identification de relations sémantiques lexicales

2.4.1 Présentation de la problématique

Cette article propose une méthode automatique pour l'analyse de discours, permettant de segmenter un texte français en relation intraphrastique, c'est à dire en triplet constitué d'une paire de verbes associés par une relation dite sémantique/discursive, en s'appuyant sur la présence d'un connecteur discursif. Par exemple, pour la phrase "J'ai apprécié l'engagement mais le jeu m'a contrarié", la relation intraphrastique détectée est la suivante : {'apprécié', {'contraste', 'mais'}, 'contrarié'}.
Nous noterons que les méthodes par apprentissage ont ici un impact réduit, car elles nécessitent un grand nombre de données. En effet, les relations rhétorique sont fréquemment implicites et donc difficilement identifiable par un humain, ce qui rend l'annotation de corpus fastidieuse et explique ainsi le faible nombre de textes annotés.

2.4.2 Détail de la base de connaissances construite

La base de connaissances de relations verbales a été construite à partir de pages Web dans le domaine .fr et contient environ 1.6 milliards de mots. Celle-ci a été étiquetée morpho-syntaxiquement, puis analysée syntaxiquement en dépendances par des méthodes automatiques. C'est à dire qu'une phrase est transformée en graphe ayant pour sommets des mots et pour arcs des relations de dépendances (sujet, objet, modal, etc.). L'objectif est de rechercher des paires de verbes liés par une relation marquée explicitement par un connecteur dans le corpus. Il existe quatre groupes de relations : causales, temporelles, de comparaison, d'expansion. Un lexique français construit manuellement est utilisé afin de repérer les relations discursives marquées explicitement.

Suite à l'application de la méthode automatique expliquée dans la prochaine section, un tableau associant relation et distribution est créé et comparé avec la ressource Annodis, annotée manuellement. Tout d'abord les relations de contraste (mais) et de cause (parce que, ainsi) sont largement majoritaires, ce qui signifie qu'elles sont le plus fréquemment marquées par les connecteurs considérés, et ensuite, les relations de continuation (et, encore) et d'élaboration (en particulier) sont bien plus fréquentes dans les annotations manuelles, ce qui signifie que ces relations reposent probablement moins sur la présence d'une marque explicite telle qu'un connecteur.

2.4.3 Mesures de l'association sémantique des verbes

Les mesures d'association lexicale considérées ont pour but de classer la force d'association des paires de verbes. Elles sont utilisées dans la recherche de coocurrences et tiennent compte de la fréquence des items reliés. Plusieurs mesures spécifiques ont été retenues. Certaines dépendent des liens de causalité entre verbes.

La PMI (pointwise mutual information) est la mesure la plus simple retenue. Son principe est d'estimer si l'apparition simultanée de deux items est supérieure à la probabilité d'apparition a priori des deux items indépendamment. Ici, des triplets constitués d'une paire de verbes avec une relation sémantique/discursive lui ont été appliqués. Après normalisation, cette mesure est comprise entre -1 et 1, approchant -1 lorsque les items n'apparaissent jamais ensemble, prenant la valeur 0 en cas d'indépendance, et la valeur 1 en cas de cooccurence complète.

D'autres mesures ont été retenues, comme la PMI pondérée, censée pallier le biais de la PMI pour les triplets peu fréquents, la PMI locale, prenant en compte la fréquence absolue d'occurence du triplet, la mesure de spécificité, permettant de mesurer la précision des cadres de sous-catégorisation, la mesure complexe Do, concernant l'apport de deux prédicats qui supportent une relation causale, et enfin une mesure combinée, évaluant l'apport de chaque composant du triplet à son informativité.

2.4.4 Description des méthodes d'évaluation

Les auteurs proposent d'effectuer une évaluation intrinsèque du lien entre les verbes, dans la perspective de valider la base comme une ressource sémantique, et une évaluation extrinsèque, en étudiant la prédiction de relations discursives en l'absence de marque explicite.

Pour la première évaluation, la possibilité d'attribuer de façon fiable un lien sémantique à une paire de verbes hors de tout contexte a été étudiée dans un premier temps. Par exemple pour la cause, y a-t-il une causalité entre les verbes pousser et tomber ? Dans un second temps, quelques paires de verbes et une centaine de contextes dans lesquels ces paires apparaissent ensemble dans le corpus d'origine ont été sélectionnées afin de juger du lien sémantique en contexte.

Pour le jugement sur des liens hors contexte, un auteur a choisi 100 paires de verbes avec des proportions similaires de paires présentant des bons et mauvais scores pour la relation choisie et selon les mesures choisies. Puis, d'autres auteurs ont dû juger pour chacune des 300 paires si elle pouvait ou non être reliée avec la relation considérée sans connaître l'origine des paires.
Une méthode plus complexe a été utilisée pour évaluer la précision des liens sémantiques en considérant des jugements d'association en contexte.

3 Utilisation d'un RST-DT dans le résumé de texte. (Axe Niki Rohani)

Maintenant que nous avons vu comment représenter un texte sous forme d'un arbre de structures rhétoriques, nous allons décrire les étapes nécessaires à l'utilisation de cette structure d'analyse de discours, dans le résumé de texte. Nous étudierons la méthode décrite dans l'article (Hirao et al. 2013) [6] . La méthode s'applique à une strucure de graphe de dépendances, DEP-DT. Nous allons donc montrer ce qu'est un DEP-DT, et la manière d'en obtenir un.

3.1 De RST-DT à DEP-DT.

Un RST-DT ne définit pas toujours la relation parent/enfant entre deux éléments de texte. Si nous avons besoin d'élaguer un arbre, il pourra être parfois difficile de déterminer quel élement d'une relation il faudra exclure. De là vient la nécessité de modéliser les relations entre deux éléments de texte, que nous appellerons EDU, pour les expliciter. Etant donné que l'algorithme de résolution pour le résumé se fonde sur un problème, connu sous le nom de Tree Knapsack Problem (TKP), il est nécessaire de définir une structure permettant l'explicitation, afin de faciliter l'élagage.
Nous allons considérer que le RST-DT d'un document est donné. De ce RST-DT, nous voulons obtenir un modèle connu sous le nom de DEP-DT.

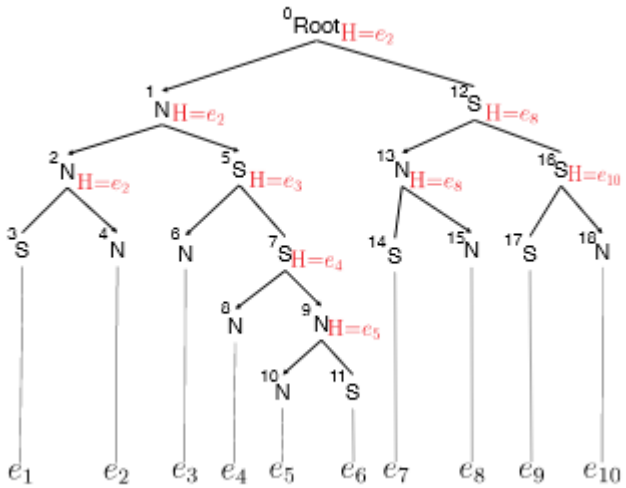

FIGURE 7: RST-DT, les head des noeuds sont nommés par H.

3.1.1 Definition d'un DEP-DT

Un DEP-DT est donc une structure représentant des relations entre EDU. Ces relations sont composées de deux EDU : le premier représente le parent et est appelé "head", et le second représente le dépendant. Le DEP-DT résulte du passage d'un algorithme sur un RST-DT. Les EDU de l'arbre sont analysés et les relations sont transformées. Nous allons voir comment s'effectue l'opération de transformation.

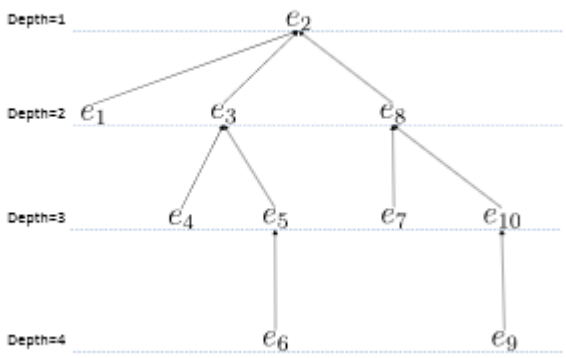

FIGURE 8: DEP-DT, Exemple de graphe généré à partir de l'arbre RST de la figure 7.

3.1.2 Algorithme

Voici l'algorithme :

1. Pour tous les noeuds qui ne sont pas terminaux, on crée un head. Le head de chaque noeud non terminal est le descendant EDU le plus à gauche dont le parent est un nucleus. Pour mieux illustrer le propos voici un exemple sur la figure 7.
2. Pour les EDU dont le parent est nucleus, nous prenons le plus proche S avec un head et l'EDU est ajouté au DEP comme un enfant du head du parent du noeud S. Dans la figure 7, e4 est un EDU de parent N, le plus proche S est le parent de N, et le head de son parent est e3. Le parent de e4 est e3.
3. Pour chaque EDU dont le parent est S, nous prenons le plus proche ancêtre avec un head et l'EDU est ajouté au DEP comme un enfant de ce head.

La figure 8 montre le résultat du passage du RST-DT de la figure 7 à un DEP-DT.

3.2 Modélisation du problème de résumé sous forme d'un Tree Knapsack Problem.

Maintenant que nous avons défini la méthode de génération d'un DEP-DT, il nous faut expliciter celle de l'algorithme résolvant le problème de résumé. Le problème peut être modélisé comme un TKP.

3.2.1 Définition d'un TKP

Notons T, l'ensemble des sous-arbres depuis racine d'un DEP-DT. Notons F(t) le score d'un seul sous arbre tel que $t \in T$. Notons L le nombre maximum de mots pouvant être dans le résumé final. Le but de TKP est de trouver le sous-arbre d'un DEP-DT maximisant une fonction $F(t)$. L'arbre optimal est obtenu comme suit :

$$t^* = argmax_{t \in T} F(t) \tag{1}$$

$$Length(t) <= L \tag{2}$$

$$F(t) = \sum_{e \in E(t)} \frac{W(e)}{Profondeur(e)} \tag{3}$$

E(t) est l'ensemble des EDU que contient t. Profondeur (e) est la profondeur de l'EDU e dans le DEP-DT. W(e) est définis comme suit :

$$\sum_{w \in W(e)} tf(w, D) \tag{4}$$

W(e) est l'ensemble de mots constituant e, et tf (w, D) est le nombre d'occurences de chaque mot dans D.
Le problème formulé, voyons comment optimiser la fonction.

3.2.2 Optimisation, ILP et formules

D'après [14], nous pouvons trouver le t qui maximise F en résolvant un problème Integer Linear Programming (ILP) [1]

$$maximise_x \sum_{i=1}^{N} \frac{W(e_i)}{Profondeur(e_i)} \tag{5}$$

$$\sum_{i=1}^{N} l_i x_i <= L \tag{6}$$

$$\forall i : x_{parent(i)} >= x_i \tag{7}$$

$$\forall i : x_i \in \{0, 1\} \tag{8}$$

1. Un problème ILP est un algorithme d'optimisation où on minimise ou maximise une fonction linéaire.

x représente un vecteur de dimension N. $x_i = 1$ signifie que le i-ème EDU est inclus dans le résumé. N est le nombre d'EDU dans le document, et l_i la longueur en nombre de mots du i-ème EDU dans le DEP-DT. La contrainte (7) assure qu'un résumé est un sous-arbre partant de la racine de DEP-DT.
Normalement, TKP est un problème NP-complet, mais avec un solveur ILP l'optimisation peut être faite en un temps raisonnable.

3.3 Expérience

Maintenant que nous avons défini le protocole utilisé dans [14], nous allons explorer quelques unes des expériences faites sur différents articles.

3.3.1 ROUGE, un moyen de noter les résumés

Afin de mesurer la performance des résumés automatiques, nous avons besoin au préalable d'une fonction efficace. Pour cela, les chercheurs ont utilisé Recall-Oriented Under study for Gisting Evaluatio (ROUGE) dans les différents articles étudiés. Pour plus de détails sur cette métrique, se référer à l'article [9].

3.3.2 Première expérience

La première expérience que nous allons présenter a été faite dans l'article [6]

1. Paramètres :
   L'évaluation expérimentale a été faite sur l'ensemble de test, pour le résumé de texte contenu dans le RST Discourse Treebank (Carlson et al., 2001) distribué par Linguistic Data Consortium. Ce Treebank regroupe 385 articles du Wall Street Journal avec des annotations RST ; 30 de ces articles ont des résumés fait par un humain.

   L'article compare la méthode TKP avec celle de Marcu (Marcu) (Marcu, 1998), un simple modèle knapsack (KP), maximum coverage model (MCP) et une méthode lead (LEAD). MCP est connu pour avoir été une méthode de l'état de l'art à l'époque de la rédaction de cet article, 2013. LEAD est aussi une méthode de résumé utilisée.
   Deux types de DEP-DT sont examinés par l'algorithme, le premier est obtenue par le gold RST-DT, l'autre est obtenu en analysant le document avec un analyseur connu nommé HILDA (du Verle and Prendinger, 2009 ; Hernault et al., 2010). La version ROUGE est 1.5.5.
2. Résultat :
   Sur la figure 9, TKP(G) et TKP(H) représentent la méthode utilisant le DEP-DT généré avec un RST-DT gold ou HILDA. Marcu (G) et Marcu (H) représentent la méthode Marcu décrite dans l'article [11] avec un RST-DT gold et HILDA.
   D'après les résultats, TKP(G) et Marcu(G) ont eu un meilleur résultat que MCP, KP et LEAD. TKP(G) a également surpassé Marcu(G).
   Les résultats ont montré l'efficacité du modèle présenté dans l'article.

	ROUGE-1		ROUGE-2	
	F	R	F	R
TKP(G)	**.310**^{H,K,L}	**.321**^{G,H,K,L}	**.108**	**.112**^{H}
TKP(H)	.281^{H}	.284^{H}	.092	.093
Marcu(G)	.291^{H}	.272^{H}	.101	.093
Marcu(H)	.236	.219	.073	.068
MCP	.279	.295^{H}	.073	.077
KP	.251	.266^{H}	.071	.075
LEAD	.255	.240	.092	.086

FIGURE 9: ROUGE de RST.

3.4 Améliorations Possibles

Maintenant que nous avons présenté la méthode basique et avons montré que l'article [1] a réussi à obtenir de meilleurs résultats que les méthodes de l'état de l'art, voyons quelles sont les améliorations qui ont été proposées, ces dernières se fondant sur la méthode.

3.4.1 Utilisation d'un DEP-DT directement généré.

Nous allons nous intéresser au document [14]. Ce document s'appuie directement sur les recherches de [6]. Dans leur document, ils proposent une méthode de génération d'un DEP-DT directement à partir d'un texte, et non plus à partir d'un RST-DT. Selon leur expertise, cela devrait améliorer le score ROUGE. Pour rappel, la méthode de Hirao, et al. 2013 [6] s'appuie sur une génération de DEP-DT à partir d'un RST-DT, ce qui rend les résultats dépendants de la qualité de la génération de RST-DT. Sur la figure 10 on voit le processus d'apprentissage, où un parser DEP-DT est entraîné depuis un RST-DT. Dans la phase de résumé, le parser analyse le document directement sans passer par un arbre RST-DT.
. L'analyseur DEP-DT, basé sur l'algorithme Maximum Spanning Tree (McDonald et al., 2005) (MST), extrait les descripteurs des EDU ei (head) et des EDU ej (dependant). Les descripteurs utilisés sont principalement ceux de (Hernault et al., 2010) et sont : - utilisation de N-gram, distance entre ei et ej, des descripteurs afin de savoir si ei et ej sont de la même phrase.
Afin d'entraîner l'analyseur les auteurs on utilisé l'algorithme Margin Infused Relaxed Algorithm (MIRA). Voici son fonctionnement :
Le score est défini ainsi $s(w, y) = w^T f_y$ pour w un vecteur de poids et un DEP-DT y. La fonction de coût $L(y, y*)$ est défini par le nombre d'EDU qui on un parent incorrect dans le DEP-DT généré. Il suffit maintenant d'optimiser le

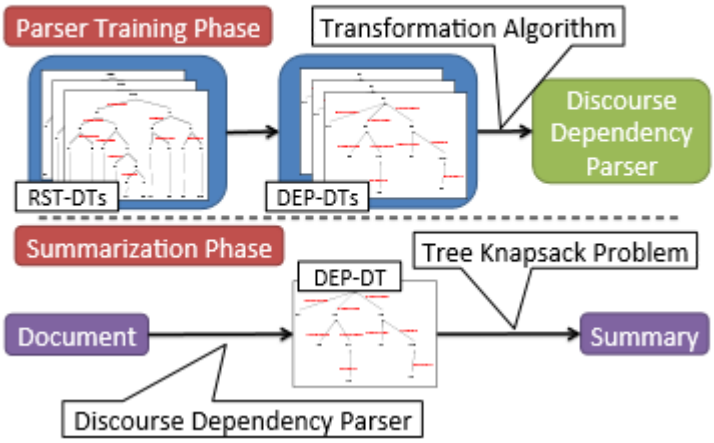

FIGURE 10: Méthode proposé par [14] pour la génération d'un résumé de texte basé sur un DEP-DT.

score en minimisant l'erreur. :

$$min_w||w - w^t|| \tag{9}$$

avec obligatoirement

$$s(w, y) - s(w, y*) >= L(y, y*) \tag{10}$$

$$y* = argmax_y s(w, y) \tag{11}$$

Les expériences ont étés menées sur un corpus RST-DT. Le corpus contient 385 articles du Wall Street Journal, et 30 de ces articles ont étés résumés à la main. Ces 30 articles ont étés utilisés comme ensemble de test pour le résumé. Nous ne parlerons uniquement que des performances du résumé avec et sans DEP-DT généré à travers le document. Nous notons **TKP-DIS-DEP** l'expérience avec un DEP-DT généré à partir de l'analyseur entraîné, et **TKP-HILDA** un DEP-DT obtenu à l'aide d'un RST-DT généré avec l'analyseur HILDA. La figure 11 montre les résultats d'expériences en comparant la méthode avec un DEP-DT généré à partir du document, et celle proposée par [6]. On voit que dans les deux expériences différentes, le résumé est plus efficace avec un DEP-DT généré directement à partir d'un document.

3.4.2 Amélioration de la fonction coût sur TKP.

Une autre optimisation proposée par [14] est celle de l'optimisation de la fonction de coût sur le problème TKP. Les auteurs ont mentionné le fait que

	ROUGE-1	ROUGE-2
TKP-DIS-DEP	0.319	0.109
TKP-HILDA	0.284	0.093

FIGURE 11: Résultats Rouge de l'expérience sur l'analyseur DEP.

	ROUGE-1	ROUGE-2
TKP-GOLD	0.321	0.112
TKP-DIS-DEP	0.319	0.109
TKP-DIS-DEP-LOSS	0.323	0.121
TKP-HILDA	0.284	0.093

FIGURE 12: Résultats Rouge de l'expérience sur l'analyseur DEP.

du moment où l'on fait un résumé en utilisant TKP, un DEP-DT où toutes les relations sont correctes n'est pas nécessaire. Le fait que les EDU proches des feuilles de l'arbre ne sont pas sélectionnés et que ceux proche de la racine sont plus importants, font que la fonction de coût peut être ajustée par rapport à ce principe. La formule de coût à été reformulée :

$$L_{Depth}(y, y^*) = \sum_{(i,r,j)\in y} \frac{1 - I(y^*, i, j)}{Depth(e_i)} \tag{12}$$

$I(y^*, i, j)$ est un indicateur qui vaut 1 si l'EDU ej est parent de ei dans le DEP-DT y*, et 0 si ce n'est pas le cas.
L'expérience a été menée en comparant les résultats d'un analyseur utilisant la méthode de [6], vue précédemment, ainsi qu'un analyseur utilisant la méthode de [14], générant un DEP-DT, et la même méthode générant automatiquement le DEP-DT, cette fois avec la fonction de coût ajustée. L'analyseur TKP-DIS-DEP-LOSS, représentant l'analyseur générant automatiquement un DEP-DT à partir d'un document, avec une fonction de coût pour TKP améliorée, sur la figure 12, montre des résultats meilleurs que la méthode sans coût amélioré. Les résultats sont même meilleurs que ceux obtenus par TKP-GOLD, représentant la méthode de résumé sur un RST-DT annoté manuellement.

4 Conclusion

Après avoir présenté le *parser Hilda*, la structure arborescente du *framework* RST, les corpus du domaine les plus utilisés et deux articles scientifiques traitant de l'identification des relations discursives, cette synthèse a montré 3 méthodes de résumés de textes, fondées sur l'analyse de discours. La méthode de [6] propose de formuler le problème sous forme d'un TKP, en utilisant un DEP-DT généré à partir d'un RST-DT, lui-même généré à partir d'un document. Les auteurs ont montré que l'analyseur HILDA, combiné à leur méthode, donnait les meilleurs résultats. Le document [14] propose de revoir la méthode de génération de DEP-DT en générant celui-ci directement à partir d'un document : cette méthode s'est quant à elle montrée plus efficace que la précédente. Les auteurs ont aussi montré qu'en reformulant la fonction de coût de TKP, en prenant en compte la manière dont TKP analyse un DEP-DT, les résultats étaient encore meilleurs.
Actuellement un article scientifique [8] propose d'utiliser des Arbres hiérarchisés (Nested Tree) afin de représenter un document au lieu de RST-DT. Cette structure permet de représenter les relations entre phrases et entre mots ce qui permet de prendre en compte certaines structures que RST-DT ne peut pas analyser. Un approfondissement sur ce sujet est très prometteur et peut permettre d'agrandir la pertinence des résumés produits.

Références

[1] Juliette Conrath Stergos Afantenos Nicholas Asher and Philippe Muller. Extraction non supervisée de relations sémantiques lexicales. *Articles longs*, page 244.

[2] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. *Building a discourse-tagged corpus in the framework of rhetorical structure theory*. Springer, 2003.

[3] Joyce Y Chai and Rong Jin. Discourse structure for context question answering. In *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL*, volume 2004, pages 23–30. Citeseer, 2004.

[4] Juliette Conrath, Stergos Afantenos, Nicholas Asher, and Philippe Muller. Extraction non supervisée de relations sémantiques lexicales. In *Actes de la 21e conférence sur le Traitement Automatique des Langues Naturelles*, pages 244–255, Marseille, France, July 2014. Association pour le Traitement Automatique des Langues.

[5] Hugo Hernault, Helmut Prendinger, Mitsuru Ishizuka, et al. Hilda : a discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3), 2010.

[6] Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. Single-document summarization as a tree knapsack problem. pages 1515–1520, 2013.

[7] Jerry R Hobbs, Mark Stickel, Paul Martin, and Douglas Edwards. Interpretation as abduction. In *Proceedings of the 26th annual meeting on Association for Computational Linguistics*, pages 95–103. Association for Computational Linguistics, 1988.

[8] Uta Kikuchi, Tsutomu Hirao, Hiroya Takamura, Manabu Okumura, and Masaaki Nagata. Single document summarization based on nested tree structure. 2014.

[9] C Y Lin. Rouge : A package for automatic evaluation of summaries. pages 25–26, 2004.

[10] William C Mann and Sandra A Thompson. Rhetorical structure theory : Toward a functional theory of text organization. *Text*, 8(3) :243–281, 1988.

[11] Daniel Marcu. Improving summarization through rhetorical parsing tuning. pages 206–215, 1998.

[12] Paul Piwek, Hugo Hernault, Helmut Prendinger, and Mitsuru Ishizuka. T2d : Generating dialogues between virtual agents automatically from text. In *Intelligent Virtual Agents*, pages 161–174. Springer, 2007.

[13] Charlotte Roze, Laurence Danlos, and Philippe Muller. Lexconn : a french lexicon of discourse connectives.

[14] Yasuhisa Yoshida, Jun Suzuki, Tsutomu Hirao, and Masaaki Nagata. Dependency-based discourse parser for single-document summarization. 2014.

Glossary

DEP-DT Dependency based Discourse Tree. 3, 12

EDU Elementary Discourse Unit. 3, 12

ILP Integer Linear Programming. 14

MCP maximum coverage model. 15

MIRA Margin Infused Relaxed Algorithm. 16

MST Maximum Spanning Tree (McDonald et al., 2005). 16

ROUGE Recall-Oriented Under study for Gisting Evaluatio. 15

RST-DT Rhetorical Structure Discourse Tree. 3

TKP Tree Knapsack Problem. 12, 14