RECITAL 2015

Table des matières

fr2sql : Interrogation de bases de données en français

Benoît Couderc[1] Jérémy Ferrero[2, 3]
(1) Aix Marseille Université, Marseille, France
(2) Compilatio, 276 rue du Mont Blanc, 74540 Saint-Félix, France
(3) LIG-GETALP, Université Grenoble Alpes, France
benoit.couderc@etu.univ-amu.fr, jeremy.ferrero@imag.fr

Résumé. Les bases de données sont de plus en plus courantes et prennent de plus en plus d'ampleur au sein des applications et sites Web actuels. Elles sont souvent amenées à être utilisées par des personnes n'ayant pas une grande compétence en la matière et ne connaissant pas rigoureusement leur structure. C'est pour cette raison que des traducteurs du langage naturel aux requêtes SQL sont développés. Malheureusement, la plupart de ces traducteurs se cantonnent à une seule base du fait de la spécificité de l'architecture de celle-ci. Dans cet article, nous proposons une méthode visant à pouvoir interroger n'importe quelle base de données à partir de questions en français. Nous évaluons notre application sur deux bases à la structure différente et nous montrons également qu'elle supporte plus d'opérations que la plupart des autres traducteurs.

Abstract.
fr2sql : database query in French
Databases are increasingly common and are becoming increasingly important in actual applications and Web sites. They often used by people who do not have great competence in this domain and who do not know exactly their structure. This is why translators from natural language to SQL queries are developed. Unfortunately, most of these translators is confined to a single database due to the specificity of the base architecture. In this paper, we propose a method to query any database from french. We evaluate our application on two different databases and we also show that it supports more operations than most other translators.

Mots-clés : Base de données (BdD), Requêtes SQL, Traducteur français SQL, Interface Homme-BdD.

Keywords: Databases (DB), Structured Query Language (SQL), French to SQL translator, Natural language interfaces to databases (NLIDB).

1 Introduction

Depuis plusieurs années, les bases de données (BdD) deviennent inévitables pour tous les sites Web ou applications gérant d'importantes masses d'informations, comme des comptes utilisateurs (banques, agences de transport, réseaux sociaux, jeux vidéo, etc.). Internet s'est peu à peu démocratisé et vulgarisé mais les bases de données quant à elles restent encore abstraites pour beaucoup de personnes. Certains postes ne requérant aucune formation en informatique ou en administration de données nécessitent tout de même de travailler étroitement avec des bases de données, comme en comptabilité ou en secrétariat par exemple. C'est dans le but qu'une personne, n'ayant aucune compétence dans le domaine de gestion des BdD, puisse non pas directement en administrer une, mais tout du moins comprendre son fonctionnement, interagir avec elle et effectuer dessus des tâches simples (interrogation, ajout, suppression), que les traducteurs du langage naturel au langage d'interrogation de bases de données ont vu le jour.

Depuis une cinquantaine d'années (Green *et al.*, 1961), le problème d'interrogation d'une base de données en langage naturel est récurrent et fait l'objet de nombreuses recherches. La majorité des outils développés sont très performants mais ne sont malheureusement pour la plupart compatibles qu'avec une seule langue naturelle source et/ou une seule base de données cible. Ils sont développés uniquement dans le but d'être l'interface d'une base de données et sont donc exclusivement compatibles avec celle-ci. En raison de structure, vocabulaire et convention de nommage extrêmement différents d'une base à l'autre, le portage d'un outil non compatible multi-bases sur une base différente de celle prévue

initialement est difficile et le rendrait de toute façon inefficace. C'est en partant de ce constat que cet article a pour objectif de concevoir un traducteur permettant l'interrogation de n'importe quelle base de données à partir du français.

Après avoir défini quelques notions et présenté l'état de l'art, on décrira comment extraire les informations relatives à une base de données cible afin d'en connaître sa structure et son vocabulaire, pour ensuite croiser ces informations avec les mots clefs de la question posée, et ainsi générer en sortie la requête SQL équivalente la plus probable. *La requête sera donc générée en fonction de la présence, du nombre et de l'ordre des mots clefs identifiés dans la phrase entrée par l'utilisateur.* Pour finir, on présentera l'évaluation de notre approche, en comparant les capacités de notre application à celles des applications déjà existantes et en évaluant ses performances sur un jeu de requêtes tests.

2 La traduction du langage naturel aux requêtes SQL

2.1 Les requêtes SQL

Une base de données est un dispositif informatique où est enregistré un ensemble d'informations. Dans une base de données relationnelle, les informations sont stockées sous forme de matrices, appelées tables. Une base de données relationnelle peut comporter une ou plusieurs tables, reliées ou non entre elles. Les entrées (informations) sont réunies dans ce que l'on appelle des colonnes (ou des champs). Un groupe de colonnes ayant trait à une même entité (objet) forme une table.

Un schéma, aussi appelé modèle de données est un diagramme (comme par exemple la figure 4) ou un descriptif textuel décrivant la distribution et l'organisation des données au sein d'une base. Il renseigne les caractéristiques de chaque type de données et les relations entre elles. Un schéma relationnel est le moyen le plus répandu de décrire une base de données relationnelle.

Le SQL (Structured Query Language) est un langage normalisé permettant d'effectuer des opérations (interrogations, modifications, suppression, etc.) sur les bases de données relationnelle.

L'objectif de cet article est de proposer une application permettant d'interroger une base de données relationnelle au moyen d'une question en français. Nous traitons donc seulement l'interrogation d'une base qui se fait à l'aide de la commande SELECT, dont la syntaxe est la suivante :

```
SELECT column_list
FROM table_list
[JOIN jointure_expression]
[WHERE conditional_expression]
[GROUP BY group_by_column_list]
[HAVING conditional_expression]
[ORDER BY order_by_column_list]
```

La syntaxe d'une commande SELECT est toujours construite de la même façon. Après le mot clef *SELECT* on énumère la liste des colonnes, qui contiennent les informations que l'on souhaite récupérer. Après le *FROM* on indique dans quelle(s) table(s) se trouvent ces informations. Toutes les lignes suivantes, marquées entre crochets, sont facultatives. Elle permettent respectivement et dans l'ordre, de préciser des tables supplémentaires si des jointures sont nécessaires, d'ajouter des contraintes à l'interrogation et de regrouper ou ordonner les valeurs de retour.

En sachant cela, lorsque l'utilisateur entre une demande du type :

Quel est l'**âge** des **élèves** qui ont pour **prénom Jean** ?

L'application doit exécuter une requête comme celle-ci :

SELECT **age** FROM **eleve** WHERE **prenom** = '**JEAN**'

Le passage de cette première phrase à la seconde représente toute la problématique du projet.

On constate, ci-dessus en gras, des *éléments clefs* communs entre la demande entrée et la requête à produire. C'est sur la correspondance entre ces éléments que la plupart des outils, y compris le nôtre, se positionnent.

2.2 État de l'art

Le lecteur est invité à se reporter à (Androutsopoulos *et al.*, 1995) et (Cimiano & Minock, 2009) pour un état de l'art complet.

L'un des premiers problèmes que soulève l'interrogation d'une base de données par un utilisateur n'ayant aucune connaissance dans ce domaine est qu'il ne connaît ni la structure ni le vocabulaire employé au sein de la base qu'il cherche à interroger. Les solutions les plus triviales consistent soit à limiter le vocabulaire qu'il peut utiliser, soit à employer une grammaire stricte, qui à l'aide de règles, limitera les phrases qu'il pourra construire. Les travaux de (Rao *et al.*, 2010) suivent la première méthode, ils contraignent l'utilisateur à rentrer une question formatée selon un dictionnaire de mots bien précis connus par l'application, se limitant donc à une base en particulier. Les applications françaises MONDE-2000 (Pasero, 1997) et DISQUE (Pasero & Sabatier, 1998) privilégient la seconde méthode. Ces dernières fonctionnent seulement sur une base de données prédéfinie, leur base de données respective, car elles possèdent le vocabulaire et l'ensemble des règles de grammaire adéquates à leur fonctionnement seulement sur ces bases ci. Le problème de cette catégorie de méthodes, d'après l'étude de (Androutsopoulos *et al.*, 1995), est qu'elles ne conviennent pas à l'utilisateur, qui se sent alors « piégé » . C'est pour cela que des méthodes moins contraignantes sont également employées. Les travaux de (Popescu *et al.*, 2003) qui, n'utilisant pas de dictionnaire de synonymes, obligent l'utilisateur à entrer une phrase libre mais contenant un lexique de façon exact ou racinalisé par rapport à celui de la BdD, sont déjà plus permissifs. Néanmoins, cette méthode oblige encore l'utilisateur à avoir une connaissance parfaite de la structure de la base, et plus particulièrement de ses noms de colonnes et tables. Des recherches plus récentes règlent ce problème en croisant les mots clefs de la base et ceux de la phrase entrée par l'utilisateur à un dictionnaire de mots spécifiques à la base (Deshpande & Devale, 2012) ou de façon plus générale à un dictionnaire de synonymes (Chaudhari, 2013).

Chandra (Chandra, 2006) rapporte également des problèmes de linguistique et d'ambiguïté. Il constate que le vocabulaire employé dans la question de l'utilisateur n'est souvent pas le même que dans la base de données et cela entraîne des problèmes de correspondances. Les travaux de (Mohite & Bhojane, 2014) mettent en avant le même phénomène, qu'ils appellent *le problème d'épellation* (spelling mistake). Il s'agit du fait que si l'utilisateur se trompe sur l'orthographe d'un mot clef, mot représentant une table ou une colonne par exemple, dans la demande entrée, il fausse tout le système empêchant une correspondance d'être trouvée. C'est donc pour cette raison que les systèmes utilisant des dictionnaires de synonymes (Chaudhari, 2013) ou des aspects sémantiques (Djahantighi *et al.*, 2008) sont de plus en plus nombreux. *Les ressources lexicales externes jouent un rôle essentiel pour la portabilité des traducteurs.*

Chaudhari (Chaudhari, 2013) développe un traducteur relativement simple mais déjà plus ambivalent. Il se contente d'identifier le type de demande (select ou delete), de transformer les nombres écrits en lettres en chiffres, d'enlever les apostrophes et la ponctuation, d'extraire les mots clefs et de construire la requête en conséquence. Il utilise un dictionnaire de synonymes à compléter à la main pour élargir le vocabulaire accepté par son système. L'inconvénient de cette méthode est qu'à chaque fois que l'on souhaite porter cet outil sur une nouvelle base, il faut effectuer des entrées manuelles dans le dictionnaire.

Pour régler définitivement les problèmes dus à la restriction de vocabulaire d'une base, certaines recherches font intervenir, en plus de la gestion de la synonymie, un aspect sémantique (Djahantighi *et al.*, 2008) afin de trouver des équivalences de sens. Le plus souvent par le biais de méthodes d'apprentissage afin d'établir des correspondances entre une question et une réponse attendue (Giordani & Moschitti, 2012) ou une question et une requête (Giordani & Moschitti, 2009).

Pour la génération des requêtes aussi plusieurs techniques font référence. Certains utilisent des grammaires avec des règles prédéfinies (Alexander *et al.*, 2013), d'autres des grammaires probabilistes (Deshpande & Devale, 2012) ou des automates (Kaur *et al.*, 2013). Certains même utilisent un système d'apprentissage (Giordani & Moschitti, 2009; Minock, 2010; Giordani & Moschitti, 2012). Ils exploitent ensuite la structure de la BdD pour générer des requêtes SQL candidates qui sont ordonnées de la plus plausible à la moins plausible grâce à un SVM-ranker basé sur un système de noyau d'arbres (Giordani & Moschitti, 2012).

Le défaut de la plupart de ces méthodes (Rao *et al.*, 2010; Pasero, 1997; Pasero & Sabatier, 1998) est qu'elles sont efficaces seulement sur une base de données particulière, celle pour laquelle leur grammaire ou leur dictionnaire est fourni, comme le travail de (Safari & Patrick, 2014) qui est uniquement fonctionnel sur sa base de données gérant une clinique, ou les recherches de (Chen, 2014) qui gère une base de données géographique, ou encore l'application de (Alexander *et al.*, 2013) qui dispose uniquement des tableaux des règles d'équivalences et des clefs primaires et liaisons de la base sur laquelle elle travaille.

Côté industriel en revanche, on remarque une plus grande portabilité multi-bases. L'approche commune dans la plupart

des produits opérationnels vise à offrir une interface de création associée à une grammaire sémantique (Minock, 2010) qui interprète les requêtes des utilisateurs sur leur base de données tel que dans l'outil EnglishQuery (Microsoft, 2000) (décrit dans (Popescu *et al.*, 2003)). English2SQL (Hurricane Electric, 2012) quant à lui, analyse la phrase et détermine sa signification à l'aide d'un algorithme non divulgué qui ne tient pas compte d'une grammaire. Cela permet, à la fois de déterminer le sens des phrases qui sont longues et complexes, et de traiter également certaines particularités, comme le problème des *contraintes muettes*. Ce problème récurrent dans les outils procédant par recherche de correspondances entre la phrase entrée par l'utilisateur et les entités de la BdD se rencontre lorsque le nom de la colonne sur laquelle doit s'effectuer la contrainte n'est pas énoncé dans la phrase. La phrase suivante illustre ce problème.

Quel est l'**âge** des **élèves** s'appellant **Jean** ?

Il est alors plus complexe d'opérer une correspondance afin de savoir dans quelle colonne chercher le mot *Jean*, à moins de fouiller toutes les colonnes de la table *eleve* à la recherche d'une valeur *Jean*, ce que fait l'outil English2SQL qui considère que chaque mot de la phrase entrée par l'utilisateur qui n'est pas un nom de table ou de colonne, peut être une valeur de colonne.

Pour finir, on peut citer des travaux issus de la communauté des bases de données (Pound *et al.*, 2010; Patil & Chen, 2012) qui tentent de rapprocher l'interrogation par mots clés de l'interrogation en langue naturelle.

Nous nous positionnons donc de façon originale vis à vis de l'état de l'art pour les raisons suivantes :

— le langage naturel source est le français ;
— la méthode présentée est portable (opérationnelle instantanément sur n'importe quelle base de données SQL) ;
— la méthode présentée possède une grammaire suffisamment permissive pour que l'utilisateur ne ressente aucune restriction lexicale ou syntaxique.

3 Notre approche

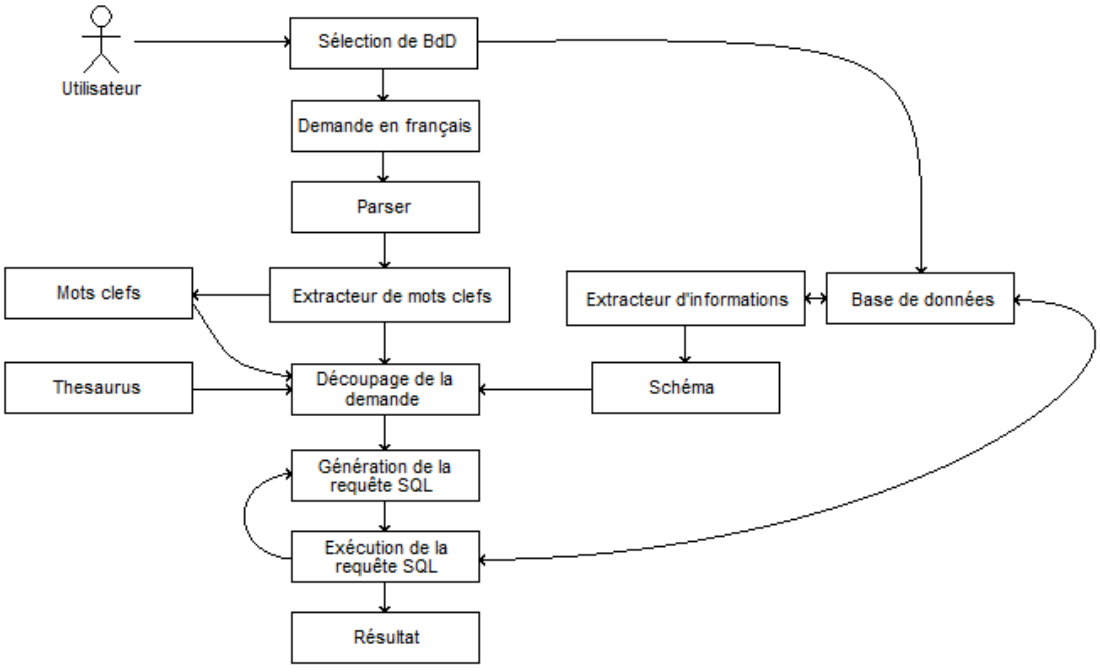

FIGURE 1 – Schéma représentant l'architecture synthétisée du fonctionnement du système fr2sql.

La figure 1 représente le fonctionnement global de l'outil fr2sql. Dans un premier temps, l'utilisateur sélectionne la base de données qu'il souhaite interroger et rentre une demande dans le champ prévu à cet effet. L'application va alors extraire

les mots porteurs de sens de la demande entrée par l'utilisateur (section 3.1). Ensuite, elle va récupérer la structure et les informations nécessaires à son usage dans la base de données qui vient d'être sélectionnée (section 3.2). À l'aide d'un dictionnaire de synonymes (section 3.3), une correspondance est recherchée entre les mots clefs extraits de la demande utilisateur et les entités de la base. Un découpage de la demande entrée est effectué en fonction des correspondances trouvées (section 3.4). Une seconde recherche est opérée afin de trouver des mots pouvant indiquer des opérations supplémentaires lors de la sélection (section 3.5). Une fois que l'application a déterminé quel type de requête était demandée et sur quels éléments, en fonction du découpage opéré et des correspondances trouvées, elle la génère (section 3.6). La requête ainsi générée est ensuite exécutée sur la base de données, si le résultat de l'interrogation ne renvoie aucune erreur (renvoyer une erreur est différent de ne renvoyer aucun résultat), il est affiché à l'utilisateur.

3.1 Extraction de mots porteurs de sens

En premier lieu, l'idée est de récupérer seulement les mots porteurs de sens dans la phrase entrée par l'utilisateur. En effet, il est important de pouvoir par la suite opérer une correspondance (un matching) entre certains concepts entrés par l'utilisateur et des éléments de la base de données. Pour cela on conserve donc plus particulièrement les noms communs, pouvant être le nom d'une table ou d'une colonne, mais aussi les noms propres, chiffres, adjectifs, etc. pouvant s'agir quant à eux d'une valeur de colonne recherchée.

Pour faire ceci, on utilise l'outil TreeTagger (Schmid, 1994) afin de filtrer les mots vides en fonction de leur classe grammaticale (prépositions, pronoms, déterminants, etc.) et d'effectuer une racinisation (stemming) des mots restant. En considérant la phrase :

Quel est l'**âge** des **élèves** qui ont pour **prénom Jean** ?

Le filtre doit retourner les éléments : « *âge, élève, prénom, Jean* ». L'ordre des mots est conservé et a son importance lors des étapes suivantes.

3.2 Récupération de l'architecture de la base de données

La seconde étape du processus consiste à récupérer l'architecture (la structure) de la base de données sur laquelle on va vouloir effectuer les requêtes et ce afin d'en connaître les entités (colonnes, tables, clefs primaires et secondaires, etc.) afin de permettre dans un second temps une correspondance avec les mots extraits de la demande utilisateur lors de la section 3.1.

Deux méthodes ont été implémentées pour parvenir à ce résultat. La première méthode consiste à collecter les informations nécessaires en interrogeant la base de données à l'aide de requêtes SQL du type « *SHOW TABLES, SHOW COLUMNS, DESCRIBE, etc.* ».

La seconde méthode, quant à elle, consiste à analyser le fichier de sauvegarde ou de création de la base de données. Avec cette méthode, une connexion en pré-processing à la base de données n'est plus requis mais un schéma SQL universel est nécessaire (certaines commandes étant syntaxiquement différentes sous MySQL ou Oracle par exemple).

À noter que fr2sql est donc seulement compatible avec une base de données SQL.

3.3 Couplage à un thésaurus

Comme le souligne l'étude de (Mohite & Bhojane, 2014), si l'utilisateur n'entre pas une phrase correctement orthographiée ou dont le vocabulaire employé n'est pas rigoureusement le même que celui de la base de données, aucune correspondance entre sa phrase et des éléments de la base ne sera trouvée et aucun résultat pertinent ne sera retourné. Il est donc important de maximiser le nombre de mots qui donneront lieu à une correspondance pertinente entre un mot clef de la demande entrée et un élément de la BdD. Pour ce faire, parallèlement au processus évoqué dans les sections précédentes, un dictionnaire de synonymes est chargé. Pour chaque mot, on a donc accès à un tableau de concepts contenant tous les mots de la langue par lesquels il peut être substitué. Par exemple, les mots « élèves » et « étudiants » représentent le même concept. Un concept est une idée, un sens représenté par un mot ou un groupe de mots. Dès lors, un concept est représenté par un mot porteur de sens lexical ainsi que tous ses mots de substitution possibles contenus dans son tableau. Le but de ce traducteur est de rendre accessible l'interrogation d'une base de données à une personne n'en connaissant ni la structure ni les mots clefs (noms de tables et de colonnes) et étant donc susceptible d'utiliser un synonyme d'un mot

employé dans la base à la place du mot lui même. Il est dès lors plus judicieux de représenter un mot par un concept, un tableau de tous les mots par lesquels il peut être substitué (un tableau de ses synonymes, lui compris) plutôt que seulement par lui-même. De cette façon pour interroger la table *eleve*, l'utilisateur pourra rentrer le mot *étudiant*.

La table 1 représente une partie des mots de substitution correspondant aux mots porteurs de sens extraits sur la phrase exemple lors de la section 3.1.

Mots porteur de sens	Mots de substitution
âge	ancienneté, ère, période, génération, ...
élève	écolier, étudiant, apprenti, collégien, lycéen, ...
prénom	nom de baptême, surnom, ...
Jean	-

TABLE 1 – Tableau d'une partie des mots de substitution disponibles pour la phrase étudiée.

Lors de cette étude, le dictionnaire utilisé est le thesaurus v.2.3 en date du 20 décembre 2011 de LibreOffice v.3.4. Cette ressource se trouve en accès libre sur internet.

Dans un souci de permettre toutes nomenclatures de nommage des colonnes et tables des bases de données, une interface d'administration du dictionnaire des synonymes a également été développée, permettant ainsi à n'importe quel utilisateur sans connaissance particulière d'ajouter, supprimer ou modifier à sa guise les synonymes de chacun des mots. Ainsi si la table regroupant les informations des étudiants a par exemple pour nom *ETUD_UNIV_01*, et qu'aucun synonyme n'a donc été automatiquement ajouté à ce nom de table, l'utilisateur pourra rentrer manuellement le mot *étudiant* comme synonyme et l'équivalence se mettra automatiquement à jour en ajoutant également tous les synonymes du mot *étudiant* contenu dans le dictionnaire des synonymes.

3.4 Découpage de la demande

À ce stade là du processus, chaque mot clef de la demande entrée par l'utilisateur est donc extrait. L'application a également à sa disposition une liste de synonymes pour chacun de ces mots clefs. L'idée est maintenant de rechercher une correspondance entre les mots clefs de la demande (ou leurs synonymes) et les entités de la base afin d'effectuer une segmentation de la demande en fonction des correspondances trouvées et ainsi de connaître au mieux la structure de la requête à générer. Lors du matching, tous les mots sont mis en minuscule et tous les caractères diacritiques (accents, cédilles, etc.) sont normalisés. Chaque mot clef retrouvé est taggé en fonction de s'il s'agit d'une colonne ou d'une table de la base de données interrogée, ou bien encore d'autre chose toujours inconnue pour le moment.

Tout d'abord, un découpage de la phrase entrée, illustré par la figure 2, est effectué en fonction des mots clefs taggés « table » et « colonne » trouvés dans la phrase, afin de savoir quel segment de la phrase correspond à quelle partie de la requête à construire. La présence d'un segment SELECT et FROM est obligatoire dans la phrase, le premier désigne quel va être le type de sélection et sur quel(s) élément(s) exactement, le second spécifie où chercher, dans quelle(s) table(s), l'élément de la sélection. Les segments TIER et WHERE sont, quant à eux, facultatifs. Le segment TIER sert lors des jointures explicites (section 3.5) et le WHERE à préciser, s'il y en a, les contraintes exercées sur la sélection. *En fonction du nombre et de la position des mots clefs dans la phrase, le découpage n'est pas le même et ne donnera donc pas lieu à la même structure de requête en sortie.* On peut notamment remarquer, que si une demande ne contient aucun mot s'apparentant à une table, elle sera forcément invalide et générera donc une erreur.

3.5 Détermination de la structure de la requête

Ensuite, dans chacun des segments obtenus lors du découpage (section 3.4), on analyse les mots clefs taggés jusqu'à présent *inconnu*. Ces mots peuvent être des facteurs de présence d'une requête dite de comptage, de calculs algébriques, d'une négation, etc., ou bien encore d'une valeur sur laquelle devra s'effectuer une contrainte. De cette façon, si un mot faisant référence au comptage comme par exemple « combien » est trouvé dans le premier segment de la phrase, celui correspondant au SELECT, le système identifie la requête à générer comme étant une requête de comptage, c'est-à-dire un SELECT COUNT(*), première branche du premier segment sur la figure 2. L'application fonctionne de la même façon, avec un système de reconnaissance de mots clefs dans les segments SELECT et/ou WHERE, pour de nombreux autres

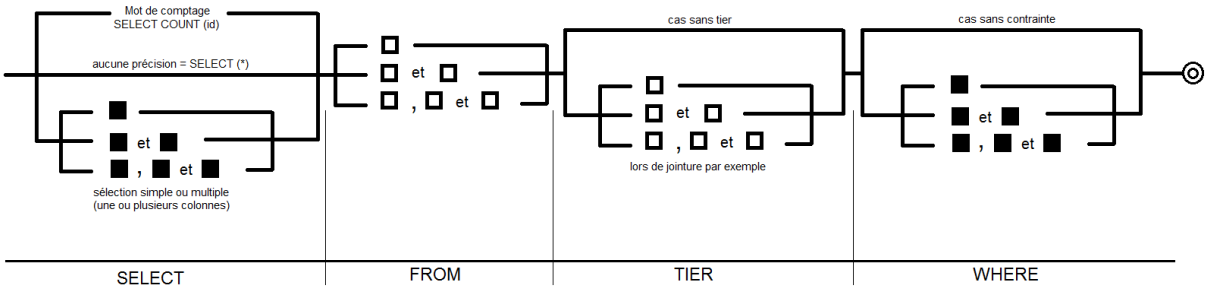

FIGURE 2 – Schéma représentant le découpage opéré sur la phrase entrée par l'utilisateur afin de savoir quel sera le type de requête à générer en sortie, les carrés blancs représentant les tables et les carrés noirs, les colonnes.

types d'opérations. La table 2 énumère de façon volontairement non exhaustive certains mots clefs donnant lieu à ces différentes opérations.

Mots clefs	Opérations
« quel est le nombre », « combien y a-t-il », ...	comptage
« ne [...] pas », « n'[...] pas », ...	négation
« plus grand que », « supérieur à », ...	supériorité
« plus petit que », « inférieur à », ...	infériorité
« quel est la somme », « additionne », ...	agrégat
« quel est la moyenne », ...	moyenne

TABLE 2 – Équivalences entre les mots clefs identifiés et les opérations à générer lors de la sélection.

Nous traitons dans cet article seulement les jointures internes (INNER JOIN). Deux types de jointures internes existent, les implicites et les explicites. Lorsqu'il y a une sélection ou une contrainte sur une colonne qui ne fait pas partie de la table du FROM, autrement dit lorsque la table à laquelle appartient la colonne ciblée n'est pas mentionnée dans la phrase entrée, c'est une jointure implicite. Il faut dans ce cas là, faire une jointure entre la table de la colonne cible et la table du FROM qui est spécifiée dans la phrase. Dans le cas d'une jointure explicite, la table sur laquelle on doit effectuer la jointure est précisée directement dans la phrase, comme dans l'exemple : « Quels sont les élèves ayant un professeur dont le prénom est Jean ? ». C'est dans ce cas là où le segment TIER, apparaissant dans la figure 2, existe et contient une ou plusieurs tables. Ici, il faut faire une jointure entre la table *eleve* et *professeur*, afin de pouvoir sélectionner des élèves tout en faisant une contrainte sur les prénoms des professeurs. Si la colonne de la sélection ou de la contrainte ne se trouve ni dans la table du FROM, ni dans une table accessible par jointure depuis la table du FROM, alors la requête est impossible à construire. La construction des jointures par l'application est rendue possible par la section 3.2, car fr2sql connaissant les clefs primaires et étrangères de chaque table, peut en déduire de façon implicite les liaisons effectives entre les tables et sait donc si une table peut être reliée à une autre et si oui, par quelle(s) table(s) passer. En effet, fr2sql peut créer des jointures passant par plusieurs tables si cela est requis (comme sur la figure 4, pour accéder à la table *professeur* depuis la table *eleve* en passant par la table *enseigner* et *classe*).

3.6 Génération de la requête à l'aide d'une grammaire « laxiste »

La permissivité des applications déjà existantes est due à leur module de matching trop « tolérant », étant donné qu'il recherche un ensemble fini de données dans un espace assez large. C'est ensuite le rôle des règles de leurs grammaires très strictes de réduire à nouveau l'espace des requêtes possibles jusqu'à proposer la plus plausible. Dans fr2sql, c'est en quelque sorte l'inverse qui est opéré. Le matching bidirectionnel réduit en premier lieu l'espace des requêtes possibles, étant donné qu'il effectue une intersection entre un faible ensemble de données et un autre faible ensemble de données. C'est ensuite une grammaire laxiste qui génère la requête en sortie, les règles de cette grammaire ne servant pas à discriminer ou ordonner les requêtes possibles mais seulement à générer la requête déterminée préalablement par le matching. Bien que cette technique rajoute un temps de traitement et des opérations en pré-processing non négligeable, cela amé-

liore sensiblement la discrimination des correspondances (matches) et permet ainsi par la suite de simplement utiliser une grammaire, qui n'a pas pour objectif d'avoir des règles les plus discriminantes possible, étant donné que les étapes précédentes auront déjà filtré la majorité des sources d'erreurs (des correspondances faux positifs), mais qui a uniquement pour objectif de générer une requête en sortie.

En raison du fait qu'un grand nombre de règles, représentant chacune la construction d'une requête possible, existe, nous avons pris la décision de les indexer à l'aide d'une table de hachage pour permettre ainsi leur acquisition plus rapidement (la recherche dans un tableau indexé par des entiers se faisant plus rapidement que la recherche d'une clef constituée de chaîne de caractères). Pour ce faire, un entier est attribué à chaque élément clef de la demande, le 1 pour les éléments de type *table*, le 2 pour les *colonnes*, le 3 pour les *valeurs*, le 4 pour un mot de comptage, etc. La demande entrée donne donc lieu à une série de nombres que l'on concatène, formant ainsi un entier unique. Le 0 correspond au fait que l'élément précédent dans la chaîne peut être présent 1 à N fois au sein de la règle. Ainsi on obtient un tableau de règles indexées par des entiers. Pour retrouver la règle équivalente à une demande entrée, il suffit de rechercher l'entier correspondant à la structure de la demande, qui se trouve également être la clef de la valeur, représentant la structure de la requête à produire en sortie, dans la table.

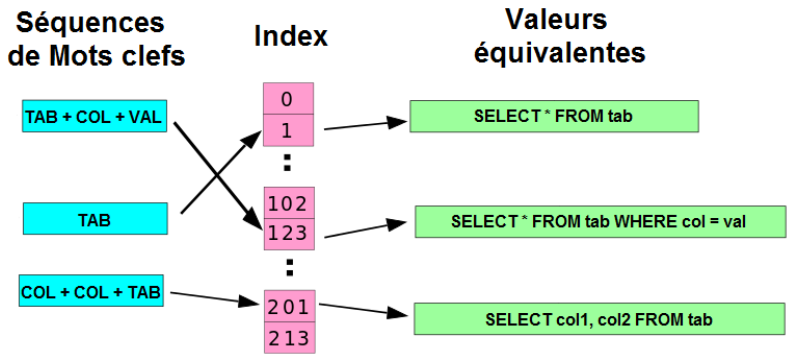

FIGURE 3 – Table de hachage indexant les structures des requêtes à générer en fonction de la structure de la phrase entrée.

Quelques exemples d'équivalences sont présenté dans la table 3. Ces exemples sont les plus triviaux, ils ont pour seul objectif ici d'exposer les équivalences formulées par les suites de mots clefs et les types de requêtes en découlant.

Règles	Opérations notables
NB_KW + TABLE	sélection avec comptage
TABLE	sélection de toutes les colonnes de la table
TABLE + (, TABLE)* + *et* + TABLE	une même demande posée sur plusieurs tables
COLONNE + TABLE	sélection d'une seule colonne
COLONNE + (, COLONNE)* + *et* + COLONNE + TABLE	sélection de plusieurs colonnes
TABLE + COLONNE + VAL	sélection avec contrainte

TABLE 3 – Équivalences entre les mots clefs identifiés et les requêtes à générer.

Maintenant que l'on connaît la structure de la requête à générer en sortie, il suffit de remplacer les tags *variable*, *table* et *colonne* par leur véritable valeur ou nom (les correspondances obtenues dans la base).

La requête est donc générée en fonction de la présence, du nombre et de l'ordre des mots clefs identifiés dans la phrase entrée par l'utilisateur. Nous appelons cette grammaire « laxiste » car elle possède suffisamment de règles pour donner l'impression à l'utilisateur qu'elle accepte toutes formes de demande. De plus, ne vérifiant que la présence et l'ordre des mots clefs, à peu près tous les mots de liaisons ou formes d'écriture sont possibles. Ceci dans le but d'être suffisamment permissif pour que l'utilisateur ne ressente aucune restriction lexicale ou syntaxique.

À noter que le *problème des contraintes muettes* n'est pas supporté par fr2sql. Les demandes telles que « Quels sont les élèves s'appelant Jean ? » ou « Quels sont les élèves de 18 ans ? » ne seront pas traitées correctement par l'application. Ici, la colonne sur laquelle doit s'effectuer la contrainte est implicite, elle n'est pas clairement spécifiée, il est donc impossible pour l'application de la retrouver. Un être humain peut comprendre que dans le premier cas c'est le nom de l'élève que

l'on souhaite et dans le second cas, son âge, mais le système, tel qu'il est conçu, n'a aucun moyen d'y parvenir. On notera également que si une même interrogation est posée sur plusieurs tables dans la même demande, comme par exemple dans la phrase « Quels sont les élèves et les professeurs ayant plus de 25 ans ? », alors le système est capable de produire plusieurs requêtes en sortie, ici deux, pour répondre à la demande ($3^{\text{ème}}$ ligne de la table 3 et $2^{\text{ème}}$ et $3^{\text{ème}}$ branche du segment FROM dans la figure 2).

Une fois une première requête SQL candidate obtenue, on tente de l'exécuter sur la base de données. Si la requête est invalide, c'est qu'elle est mal construite (nom de colonne à la place de table, oubli d'une valeur, oubli d'une colonne, etc.). Le système tente alors de la construire différemment, si en s'exécutant, elle retourne encore une erreur, le système renvoie un message identifiant précisément le type de l'erreur. Ce système permet certes de réduire les erreurs en sorties mais surtout de connaître la catégorie des erreurs renvoyées. D'après les travaux de (Androutsopoulos *et al.*, 1995), l'un des défauts les plus reprochés aux traducteurs du langage naturel à une autre langue, est le manque de clarté. Grâce à ce système, ce problème est en partie résolu.

4 Évaluation et tests

4.1 Base de tests

Aucune base de tests pour un outil en français et supportant plusieurs tables n'ayant été mise à disposition à ce jour, c'est plus de 200 requêtes reprenant tous les éléments standards d'une interrogation classique (sélection simple ou multiple, comptage et calculs algébriques, conjonctions, disjonctions, jointures, conditions, négations, limites et ordonnancement), qui ont été testées sur deux bases différentes, afin de bien illustrer la portabilité multi-bases de notre approche.

Les requêtes ont toutes été écrites manuellement, dont 100 par des personnes ayant des connaissances en BdD et 100 par des personnes n'en ayant aucune dans ce domaine. Toutes ces personnes avaient reçu des informations grossières sur les tables des bases tests mais aucune n'avait pris connaissance de leur structure.

Les deux bases de données tests présentent des structures ainsi que des conventions de nommage différentes. La figure 4 représentent les schémas relationnels de ces bases, elles sont reproductibles et exploitables pour de futurs travaux.

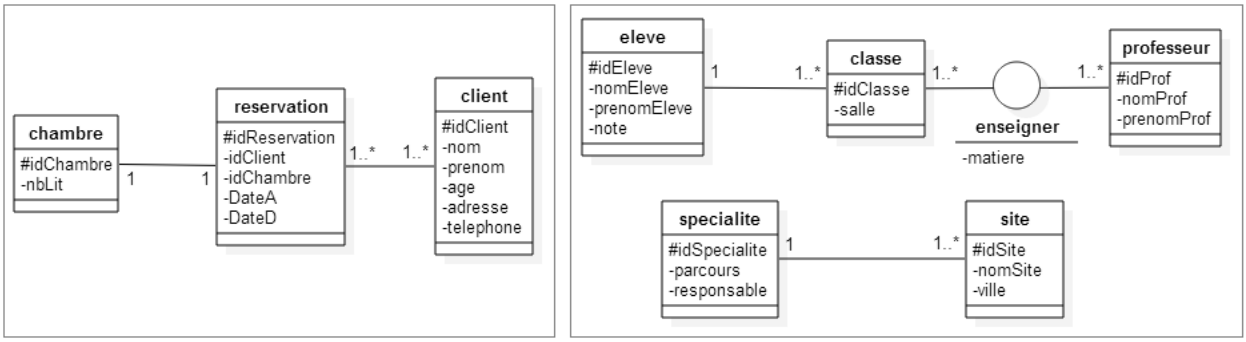

FIGURE 4 – Schémas relationnels des bases de données tests.

4.2 Résultats

Nous adoptons la définition de *précision* et *rappel* décrites dans (Minock *et al.*, 2008) et (Popescu *et al.*, 2003). Une phrase entrée par l'utilisateur peut demander la génération de plusieurs requêtes ou peut ne produire aucune requête.

$$\text{précision} = \frac{card(\text{réponses correctes retournées})}{card(\text{réponses retournées})}$$

$$\text{rappel} = \frac{card(\text{réponses correctes retournées})}{card(\text{phrases entrées})}$$

On constate dans la table 4 que notre méthode supporte autant si ce n'est plus d'opérations que les outils déjà existants, tout en étant compatible instantanément sur n'importe quelle base de données. On constate également que seule fr2sql est compatible sur toutes bases et gère la synonymie de façon à ne pas restreindre les demandes au seul vocabulaire employé dans la base.

	MONDE	SQL-HAL	English2SQL	fr2sql
Sélection sur colonne	OUI	OUI	OUI	OUI
Sélection sur table	OUI	OUI	OUI	OUI
Sélection multiple	NON	OUI	OUI	OUI
Comptage	OUI	OUI	OUI	OUI
Comptage multiple	NON	OUI	OUI	OUI
Contrainte simple	OUI	OUI	OUI	OUI
Contrainte muette	OUI	NON	OUI	NON
Disjonction	OUI	OUI	OUI	OUI
Conjonction	OUI	OUI	OUI	OUI
Contrainte croisée	OUI	OUI	OUI	OUI
Gestion des dates	NON	OUI	OUI	NON
Rangement	NON	NON	NON	OUI
Comparaison	OUI	OUI	OUI	OUI
Calcul algébrique	NON	OUI	OUI	OUI
Négation	OUI	NON	OUI	OUI
Synonymie	NON	NON	NON	OUI
Jointure pour sélection	OUI	NON	OUI	OUI
Jointure pour condition	OUI	NON	OUI	OUI
Requêtes imbriquées	OUI	NON	OUI	NON
Compatibilité sur plusieurs base de données	NON	OUI	OUI	OUI
Restriction de vocabulaire ou de grammaire	OUI	OUI	OUI	NON

TABLE 4 – Opérations supportées par les différentes applications de traductions.

Les résultats dans la table 5 illustrent les performances de notre méthode en fonction du type de demande. Les questions entraînant une sélection sans jointure montrent de bien meilleurs résultats (0.957 de F-score en moyenne) qu'une demande donnant lieu à une jointure (0.761 de F-score). À noter également que les phrases donnant lieu à des requêtes imbriquées ne sont pas correctement traduites par l'application qui ne les gère tout simplement pas encore à l'heure actuelle.

	Précision	Rappel	F-mesure
Tout type de requêtes	0.939	0.850	0.892
Sélections seulement	1	0.969	0.984
Avec jointures	0.832	0.702	0.761
Avec conditions	0.987	0.880	0.930

TABLE 5 – Performances par catégorie de requête de l'application fr2sql.

5 Conclusions

Bien que nous n'avons pas pu effectuer clairement de comparatif, d'après l'état de l'art, notre méthode montre des résultats globalement équivalents à la plupart des applications actuelles (une précision supérieur à 0.90 pour un rappel supérieur à 0.85) avec néanmoins une faiblesse au niveau des jointures (0.761 de F-mesure). On notera également l'impossibilité de celle-ci à gérer les contraintes muettes et à générer des requêtes imbriquées.

Dans de futurs travaux, nous prévoyons de traiter les contraintes muettes, en conservant les verbes comme mots clefs et en ajoutant quelques règles dans la grammaire. Cela permettra par exemple de définir que « l'élève s'appellant Jean » signifie la même chose que « l'élève dont le nom est Jean » ou bien encore que « l'élève de 18 ans » équivaut à « l'élève

ayant pour âge 18 ans ». De plus, il est prévu de détecter la langue de la demande entrée par l'utilisateur afin d'utiliser un dictionnaire de synonymes ayant trait à cette langue et d'ajuster les règles en fonction de la langue pour ainsi rendre le système robuste à d'autres langues que le français.

Pour conclure, bien que perfectible, cette approche permet bien d'interroger n'importe quelle base de données SQL, répondant ainsi aux objectifs de portabilité fixés, tout en gardant des performances dans la moyenne des applications déjà existantes et en couvrant une large palette d'opérations de sélection.

Références

ALEXANDER R., RUKSHAN P. & MAHESAN S. (2013). Natural Language Web Interface for Database (NLWIDB). *In CoRR*.

ANDROUTSOPOULOS I., RITCHIE G. & THANISCH P. (1995). Natural Language Interfaces to Databases - An Introduction. *In Journal of Natural Language Engineering*, **1**, 29–81.

CHANDRA Y. (2006). *Natural Language Interfaces to Databases*. PhD Thesis. University of North Texas, USA.

CHAUDHARI P. P. (2013). Natural Language Statement to SQL Query Translator. *In International Journal of Computer Applications*, **82**(5), 18–22.

CHEN W. (2014). Parameterized Spatial SQL Translation for Geographic Question Answering. *In Semantic Computing (ICSC), 2014 IEEE International*, p. 23–27.

CIMIANO P. & MINOCK M. (2009). Natural Language Interfaces : What is the Problem ? - A data-driven quantitative analysis. *In 14th International Conference on Applications of Natural Language to Information System (NLDB)*, **5723**.

DESHPANDE A. K. & DEVALE P. R. (2012). Natural Language Query Processing Using Probabilistic Context Free Grammar. *In International Journal of Advances in Engineering and Technology*, **3**, 568–573.

DJAHANTIGHI F. S., NOROUZIFARD M., DAVARPANAHAND S. & SHENASSA M. H. (2008). Using Natural Language Processing in Order to Create SQL Queries. *In Proceedings of the International Conference on Computer and Communication Engineering*, p. 600–604.

GIORDANI A. & MOSCHITTI A. (2009). Semantic Mapping Between Natural Language Questions and SQL Queries via Syntactic Pairing. *In Natural Language Processing and Information Systems*, **5723**, 207–221.

GIORDANI A. & MOSCHITTI A. (2012). Translating Questions to SQL Queries with Generative Parsers Discriminatively Reranked. *In COLING 2012 : Posters*, p. 401–410.

GREEN B. F., WOLF A. K., CHOMSKY C. & LAUGHERY K. (1961). Baseball : An automatic question-answerer. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '61 (Western), p. 219–224, New York, NY, USA : ACM.

HURRICANE ELECTRIC I. S. (2012). English2SQL powered by he.net. Logiciel.

KAUR J., CHAUHAN B. & KOREPAL J. K. (2013). Implementation of Query Processor Using Automata and Natural Language Processing. *In International Journal of Scientific and Research Publications*, **3**.

MICROSOFT (2000). TechNet : Developing with English Query. Logiciel.

MINOCK M. (2010). C-Phrase : A system for building robust natural language interfaces to databases. *In Data and Knowledge Engineering*, **69**(3), 290–302.

MINOCK M., OLOFSSON P. & NÄSLUND A. (2008). Towards Building Robust Natural Language Interfaces to Databases. *In 13th International Conference on Applications of Natural Language to Information System (NLDB)*, **5039**, 187–198.

MOHITE A. & BHOJANE V. (2014). Challenges and Implementation Steps of Natural Language Interface for Information Extraction from Database. *In International Journal of Recent Technology and Engineering (IJRTE)*, **3**.

PASERO R. (1997). Une interface en français à une base de données discographiques. Logiciel.

PASERO R. & SABATIER P. (1998). Une interface en français à une base de données sur les États du monde. Logiciel.

PATIL R. & CHEN Z. (2012). Struct : incorporating contextual information for english query search on relational databases. In T. W. LING, G. YU, J. LU & W. W. 0011, Eds., *KEYS*, p. 11–22 : ACM.

POPESCU A. M., ETZIONI O. & KAUTZ H. (2003). Towards a Theory of Natural Language Interfaces to Databases. *In Proceedings of the 8th International Conference on Intelligent User Interfaces*, p. 149–157.

POUND J., ILYAS I. F. & WEDDELL G. (2010). Expressive and flexible access to web-extracted data : A keyword-based structured query language. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, p. 423–434, New York, NY, USA : ACM.

RAO G., AGARWAL C., CHAUDHRY S., KULKARNI N. & PATIL D. S. (2010). Natural Language Query Processing using Semantic Grammar. *In International Journal on Computer Science and Engineering*, **2**, 219–223.

SAFARI L. & PATRICK J. D. (2014). Restricted natural language based querying of clinical databases. *In Journal of Biomedical Informatics*.

SCHMID H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

Désambiguïsation lexicale à base de connaissances par sélection distributionnelle et traits sémantiques

Mokhtar Boumedyen Billami[1]
(1) Aix-Marseille Université, LIF UMR 7279, 163 avenue de Luminy, 13288 Marseille Cedex 9
mokhtar.billami@lif.univ-mrs.fr

Résumé. La désambiguïsation lexicale permet d'améliorer de nombreuses applications en traitement automatique des langues (TAL) comme la recherche d'information, l'extraction d'information, la traduction automatique, ou la simplification lexicale de textes. Schématiquement, il s'agit de choisir quel est le sens le plus approprié pour chaque mot d'un texte. Une des approches classiques consiste à estimer la similarité sémantique qui existe entre les sens de deux mots puis de l'étendre à l'ensemble des mots du texte. La méthode la plus directe donne un score de similarité à toutes les paires de sens de mots puis choisit la chaîne de sens qui retourne le meilleur score (on imagine la complexité exponentielle liée à cette approche exhaustive). Dans cet article, nous proposons d'utiliser une méta-heuristique d'optimisation combinatoire qui consiste à choisir les voisins les plus proches par sélection distributionnelle autour du mot à désambiguïser. Le test et l'évaluation de notre méthode portent sur un corpus écrit en langue française en se servant du réseau sémantique BabelNet. Le taux d'exactitude obtenu est de 78% sur l'ensemble des noms et des verbes choisis pour l'évaluation.

Abstract.

A Knowledge-Based Approach to Word Sense Disambiguation by distributional selection and semantic features.
Word sense disambiguation improves many Natural Language Processing (NLP) applications such as Information Retrieval, Information Extraction, Machine Translation, or Lexical Simplification. Roughly speaking, the aim is to choose for each word in a text its best sense. One of the most popular method estimates local semantic similarity relatedness between two word senses and then extends it to all words from text. The most direct method computes a rough score for every pair of word senses and chooses the lexical chain that has the best score (we can imagine the exponential complexity that returns this comprehensive approach). In this paper, we propose to use a combinatorial optimization metaheuristic for choosing the nearest neighbors obtained by distributional selection around the word to disambiguate. The test and the evaluation of our method concern a corpus written in French by means of the semantic network BabelNet. The obtained accuracy rate is 78 % on all names and verbs chosen for the evaluation.

Mots-clés : désambiguïsation lexicale non supervisée, mesure de similarité distributionnelle, mesures de similarité sémantique.
Keywords: unsupervised word sense disambiguation, distributional similarity measure, semantic similarity measures.

1 Introduction

La désambiguïsation des sens de mots est une *« tâche intermédiaire »* (Wilks, Stevenson, 1996), qui ne constitue pas une fin en soi, mais est plutôt nécessaire à un niveau ou à un autre pour accomplir la plupart des tâches de traitement des langues. Ainsi, la désambiguïsation lexicale suscite de l'intérêt depuis les premiers jours du traitement informatique de la langue (Ide, Véronis, 1998). Elle est nécessaire pour plusieurs applications telles que la recherche d'information, l'extraction d'information, la traduction automatique, l'analyse du contenu, la fouille de textes, la lexicographie et le web sémantique. La plupart des systèmes de désambiguïsation lexicale existants s'appuient sur deux grandes étapes (Navigli, 2009) : (1) représentation de l'ensemble des sens d'un mot ; et (2) choix du sens le plus proche du mot par rapport à son contexte. La première étape repose sur l'utilisation de ressources lexicales telles que les dictionnaires ou les réseaux sémantiques. Ide et Véronis (1998) ont montré que la meilleure possibilité d'identifier le sens d'un mot ambigu est de se référer à son contexte.

L'une des approches les plus classiques pour déterminer le sens le plus probable d'un mot polysémique est d'estimer la proximité sémantique entre chaque sens candidat par rapport à chaque sens de chaque mot appartenant au contexte du

mot à désambiguïser[1]. En d'autres termes, il s'agit de donner des scores locaux et de les propager au niveau global. Une application de cette méthode est proposée dans (Pederson *et al.*, 2005). On imagine la complexité exponentielle que retourne cette approche exhaustive. On se retrouve facilement avec un temps de calcul très long alors que le contexte qu'il est possible d'utiliser est petit. Par exemple, pour une phrase de 10 mots avec 10 sens en moyenne, il y aurait 10^{10} combinaisons possibles. Le calcul exhaustif est donc très compliqué à réaliser et, surtout, rend impossible l'utilisation d'un contexte plus important. Pour diminuer le temps de calcul on peut utiliser une fenêtre autour du mot afin de réduire le temps d'exécution d'une combinaison mais le choix d'une fenêtre de taille quelconque peut mener à une perte de cohérence globale de la désambiguïsation.

Le contexte du mot à désambiguïser est délimité par une fenêtre textuelle qui se situe à gauche ou à droite ou des deux côtés et dont la taille peut varier. Les fenêtres peuvent être délimitées soit à l'aide de séparateurs de phrases ou de paragraphes, soit à l'aide de *« n-grammes »* qui permettent d'observer un certain nombre (*n-1*) de mots entourant le mot polysémique dans le texte. La définition de la taille de la fenêtre textuelle est liée à celle de la distance optimale entre les mots ambigus et les indices contextuels pouvant servir à leur désambiguïsation (Audibert, 2007). Selon Yarowsky (1993), une grande fenêtre est nécessaire pour lever l'ambiguïté des noms alors que seulement une petite fenêtre suffit pour le cas des verbes ou des adjectifs. Dans un cadre d'analyse distributionnelle de données, plusieurs recherches sont faites sur la construction automatique de thésaurus à partir de cooccurrences de mots provenant d'un corpus de grande taille. Pour chaque mot cible en entrée, une liste ordonnée de voisins les plus proches (*nearest neighbours*) lui est attribuée. Les voisins sont ordonnés en termes de la similarité distributionnelle qu'ils ont avec le mot cible. Lin (1998) a proposé une méthode pour mesurer la similarité distributionnelle entre deux mots (un mot cible et son voisin). Dans cet article, nous nous intéressons à cette approche d'analyse distributionnelle et nous l'utilisons dans la tâche de la désambiguïsation lexicale.

Dans un premier temps, nous présentons un état de l'art sur les méthodes de désambiguïsation lexicale (section 2), notre méthodologie de désambiguïsation à base de traits sémantiques est présentée dans la section 3. Elle décrit le corpus de travail et d'évaluation, le réseau sémantique BabelNet que nous utilisons pour le choix des sens de mots, la mesure de similarité distributionnelle pour le choix des voisins les plus proches ainsi que les algorithmes permettant de retourner le sens le plus probable d'un mot selon le contexte dans lequel il apparait. Nous présentons par la suite les données des expériences menées (section 4) ainsi que les résultats de l'évaluation des différents algorithmes (section 5). Nous terminons par une conclusion et quelques perspectives (section 6).

2 État de l'art

Nous pouvons distinguer deux grandes catégories de méthodes de désambiguïsation : (1) dirigées par les données, où l'on trouve les méthodes supervisées et non supervisées. Les méthodes supervisées s'appuient sur un corpus d'apprentissage réunissant des exemples d'instances désambiguïsées de mots. Les méthodes non supervisées exploitent les résultats de méthodes automatiques d'acquisition de sens ; (2) basées sur les connaissances, nécessitant une modélisation étendue aux informations lexico-sémantiques ou encyclopédiques. Ces méthodes peuvent être combinées avec les méthodes non supervisées. La désambiguïsation peut être de deux types : (a) désambiguïsation ciblé, seulement sur un mot particulier dans un texte ; (b) désambiguïsation complète pour tous les mots pleins[2] d'un texte. Il y a deux critères importants pour choisir l'algorithme à utiliser. Le premier critère est une mesure de similarité qui dépend des contraintes de la base de connaissances et du contexte applicatif. Le deuxième critère est le temps d'exécution de l'algorithme. Le lecteur pourra consulter (Ide, Véronis, 1998) pour les travaux antérieurs à 1998 et (Navigli, 2009) pour un état de l'art complet.

Une annotation de mots d'un corpus avec des sens désambiguïsés provenant d'un inventaire de sens (ex. WordNet) est extrêmement coûteuse. À l'heure actuelle très peu de corpus annotés sémantiquement sont disponibles pour l'anglais ; à notre connaissance, rien n'existe pour le français. Le consortium de données linguistiques (*Linguistic Data Consortium*[3]) a distribué un corpus contenant approximativement 200 000 phrases en anglais issues du corpus Brown et Wall Street Journal dont toutes les occurrences de 191 lemmes ont été annotées avec WordNet (Ng, Lee, 1996). Le corpus SemCor (Miller *et al.*, 1993) est le plus grand corpus annoté sémantiquement. Il contient 352 textes annotés avec près de 234 000 sens au total. Cependant, ces corpus contiennent peu de données pour être utilisés avec des méthodes statistiques. Ng (1997) estime que, pour obtenir un système de désambiguïsation à large couverture et de haute précision, nous avons probablement besoin d'un corpus d'environ 3,2 millions de mots de sens étiquetés. L'effort humain pour construire un tel

[1] Il peut s'agir d'une phrase, d'un paragraphe ou d'un texte brut.

[2] Mots pleins : noms, verbes, adjectifs et adverbes.

[3] *Linguistic Data Consortium (LDC)*. https://www.ldc.upenn.edu

corpus d'apprentissage peut être estimé à 27 années pour une annotation d'un mot par minute par personne (Edmonds, 2000). Il est clair qu'avec une telle ressource à portée de main, les systèmes supervisés seraient beaucoup plus performants mais ça ne reste qu'une hypothèse.

Des efforts ont été fournis pour annoter sémantiquement des corpus en utilisant des méthodes de boostrapping. Hearst (1991) a proposé un algorithme (*CatchWord*) pour une classification des noms qui comprend une phase d'apprentissage au cours de laquelle plusieurs occurrences de chaque nom sont manuellement annotées. Les informations statistiques extraites du contexte de ces occurrences sont ensuite utilisées pour lever l'ambiguïté d'autres occurrences. Si une autre occurrence peut être désambiguïsée avec certitude, le système acquiert automatiquement des informations statistiques de ces nouvelles occurrences désambiguïsées, améliorant ainsi ses connaissances progressivement. Hearst indique qu'une première série d'au moins 10 occurrences est nécessaire pour la procédure, et que 20 ou 30 occurrences sont nécessaires pour une haute précision.

Enfin, les méthodes de désambiguïsation lexicale à base de connaissances se composent d'une part de mesures de similarité sémantique locales qui donnent une valeur de proximité entre deux sens de mots et, d'autre part, d'algorithmes globaux qui utilisent ces mesures pour trouver les sens selon le contexte à l'échelle de la phrase ou du texte. Plusieurs solutions, autres que l'algorithme exhaustif, ont été proposées. Par exemple, des approches à base de corpus pour diminuer le nombre de combinaisons à examiner comme la recherche des chaînes lexicales compatibles (Vasilescu *et al*., 2004) ou encore des approches issues de l'intelligence artificielle comme le recuit simulé[4] (Cowie *et al.*, 1992) ou les algorithmes génétiques (Gelbukh *et al.*, 2003). Pour plus de détails, le lecteur pourra consulter (Tchechmedjiev, 2012).

3 Méthodologie

Désambiguïser tous les mots pleins d'un corpus dont le contexte représente un paragraphe est une tâche qui demande beaucoup de temps si on se base sur un algorithme exhaustif simple. La clé de notre approche de désambiguïsation est l'observation des voisins de chaque mot polysémique dans le texte : au lieu de comparer chaque sens d'un mot à désambiguïser avec tous les sens de tous les mots qui se trouvent dans le texte, nous faisons une comparaison uniquement avec les sens des voisins sélectionnés au moyen d'une similarité distributionnelle. D'une part, ces voisins fournissent souvent des indices sur le sens le plus probable d'un mot dans un texte. D'autre part, cela nous permet de diminuer le temps d'exécution de l'algorithme et de ne pas perdre une cohérence au niveau de la désambiguïsation de tous les mots du texte. Il s'agit de garder l'homogénéité des mots afin de retourner le sens le plus spécifique à chaque mot au lieu de retourner le sens le plus général.

3.1 Corpus de données

3.1.1 Corpus de travail

Nous avons à disposition un ensemble de trois corpus de différents genres. Le premier corpus est une collection de l'agence française de presse (*French press agency*[5]). Le deuxième corpus est une collection d'articles d'un journal local français (*l'EST Républicain*[6]). Le troisième corpus est une collection d'articles issue de la ressource encyclopédique libre, *Wikipédia*[7]. L'ensemble des données de ces trois corpus est décrit dans le tableau 1. Ces différents corpus ont été analysés automatiquement par la chaîne de traitement Macaon[8] (Nasr *et al.*, 2011). Nous avons obtenu 2 754 686 triplets[9] différents de dépendances syntaxiques correspondant à 31 774 noms uniques et 5 421 verbes uniques. Ces triplets sont stockés et indexés après extraction de 12 785 450 cooccurrences.

[4] Méthode d'optimisation stochastique classique fondée sur les principes physiques du refroidissement des métaux qui a été appliquée à la désambiguïsation lexicale.

[5] *French press agency* (*AFP*). http://www.afp.com/fr

[6] *L'EST Républicain*. http://www.estrepublicain.fr/

[7] *Wikipédia*, encyclopédie libre sur le web. https://fr.wikipedia.org

[8] Macaon, chaîne de traitement permettant d'effectuer des tâches standard du TAL. http://macaon.lif.univ-mrs.fr

[9] Un triplet de dépendance syntaxique se compose d'une tête, d'un type de dépendance et d'un modificateur. Par exemple, (*recouvrer, suj, regard*) est un triplet extrait de la phrase *« leurs regards recouvraient les eaux du fleuve »*.

Corpus	Phrases	Tokens
AFP	2 041 146	59 914 238
EST REP	2 998 261	53 913 288
WIKI	1 592 035	33 821 460
Total	6 631 442	147 648 986

Tableau 1 : Données du corpus de travail

3.1.2 Corpus d'évaluation

Nous travaillons sur deux corpus différents, corpus IREST[10] contenant 10 textes et un corpus brut[11] contenant 20 textes pour un total de 30 textes. Nous avons 6 235 occurrences de mots (4 139 occurrences de mots pleins) et une moyenne de 208 occurrences (138 occurrences de mots pleins) par texte (cf. section 4, tableau 2). Les textes sont lemmatisés et annotés en parties du discours par Macaon. Le travail de désambiguïsation que nous menons porte sur des unités monolexicales (les expressions polylexicales n'ont pas été prises en compte).

3.2 Ressource lexicale BabelNet

BabelNet[12] (Navigli, Ponzetto, 2012) est un réseau sémantique multilingue permettant de fournir des sens et des entités nommées[13]. BabelNet a été créé en intégrant automatiquement la plus grande encyclopédie multilingue - c'est-à-dire Wikipédia – avec WordNet (Fellbaum, 1998). La construction de cette ressource s'est faite en deux grandes étapes : (1) mapping entre Wikipédia et WordNet ; (2) un système de traduction automatique, basé sur l'application de traduction en ligne de Google, pour recueillir une grande quantité de concepts multilingues et de compléter par les traductions manuellement éditées dans Wikipédia. La construction de BabelNet a permis de couvrir les sens manquants dans WordNet. Le résultat est une ressource multilingue qui fournit des entrées lexicalisées multilingues, reliées entre elles avec une grande quantité de relations sémantiques. De la même façon que WordNet, BabelNet regroupe les mots en différentes langues par groupes de synonymes appelés *Babel synsets*. Pour chaque *Babel synset*, BabelNet fournit des définitions textuelles (appelées gloses) en plusieurs langues, obtenues à partir de WordNet et Wikipédia. A partir de la version 2.0 de BabelNet (octobre, 2013) cette ressource intègre non seulement Wikipédia mais aussi Wiktionary, Wikidata, OmegaWiki et Open Multilingual WordNet, une collection de WordNets disponibles dans différentes langues. A la différence de WordNet qui offre une seule définition par sens, BabelNet permet d'offrir plusieurs définitions pour plusieurs langues.

La version 2.0 de BabelNet couvrait 50 langues, y compris toutes les langues européennes. Actuellement, la version 3.0 couvre 271 langues et contient plus de 13 millions de synsets. Chaque Babel synset contient en moyenne 5,5 synonymes. Le réseau sémantique comprend toutes les relations lexico-sémantiques de WordNet (hyperonymie et hyponymie, méronymie et holonymie, antonymie et synonymie, etc.). Pour la langue française, BabelNet contient actuellement 4 120 733 synsets et 174 591 mots polysémiques. Nous avons choisi d'utiliser BabelNet parce qu'il offre un très grand nombre de synsets et couvre plusieurs mots polysémiques par rapport à d'autres ressources lexicales pour le français comme Wolf[14] (Hanoka, Sagot, 2012) qui offre dans sa version bêta 1.0 un ensemble de 59 091 synsets décrits avec des synonymes et des définitions.

[10] Textes standards pour des tests de vitesse de lecture. http://vision-research.eu

[11] Textes de lecture pour enfants en école primaire.

[12] BabelNet, ressource lexicale. http://babelnet.org

[13] Nous nous intéressons à la désambiguïsation des sens sans tenir compte de la présence des entités nommées. Nous considérons un sens comme étant un concept dans le réseau sémantique.

[14] Wolf est inspiré de WordNet et présente un réseau sémantique libre pour le français. http://alpage.inria.fr/~sagot/wolf.html

BabelNet dans sa version 2.5.1 a été utilisé pour la réalisation d'un système de désambiguïsation et de détection d'entités nommées, Babelfy[15] (Moro *et al.*, 2014). Babelfy obtient de bonnes performances grâce à la structure de BabelNet qui permet l'intégration des sens lexicographiques et d'entités encyclopédiques en un seul réseau sémantique. Nous utilisons la version 2.5.1 pour l'évaluation de nos expériences afin de comparer nos résultats avec ceux retournés par Babelfy.

3.3 Similarité distributionnelle

La similarité distributionnelle est une mesure indiquant le degré de cooccurrence entre un mot cible et son voisin apparaissant dans des contextes similaires. Par exemple, dans un premier texte les voisins de *fleuve* peuvent être *rivière, eau, affluent*. Le sens le plus probable pour *fleuve* est décrit dans BabelNet par trois définitions.

Sens 1

(1) *Cours d'eau naturel* ;
(2) *En hydrographie, une rivière est un cours d'eau qui s'écoule sous l'effet de la gravité et qui se jette dans une autre rivière ou dans un fleuve, contrairement au fleuve qui se jette, lui, selon cette terminologie, dans la mer ou dans l'océan* ;
(3) *Courant d'eau qui coule d'une altitude élevée à une altitude basse pour arriver dans un lac ou une mer, sauf dans les aires désertiques ou il peut arriver sur rien.*

Dans un deuxième texte, les voisins de *fleuve* peuvent être *mer, eau, océan*. Le sens le plus probable pour *fleuve* est décrit dans BabelNet par deux définitions.

Sens 2

(1) *Cours d'eau se jetant dans une mer* ;
(2) *En hydrographie francophone, un fleuve est un cours d'eau qui se jette dans une mer, dans l'océan, Il se distingue d'une rivière, qui se jette dans un autre cours d'eau*.

Plus la similarité distributionnelle entre les voisins est forte plus la probabilité d'avoir le sens le plus probable est grande. Nous utilisons la méthode proposée par Lin (1998) pour l'analyse distributionnelle de données sur notre corpus de travail. Nous avons à disposition un ensemble de relations grammaticales extraites à partir d'une analyse automatique sur les données du corpus de travail. Cette extraction est limitée à un certain nombre de relations de dépendances syntaxiques et nous a permis d'avoir un ensemble de cooccurrences entre les noms et les verbes. Pour chaque nom, nous avons des relations de cooccurrences, par exemple la relation *objet-de* et *sujet-de ;* pour les verbes, la relation *a-pour-objet*, etc. Ainsi, nous avons un ensemble de triplets de cooccurrences $< w, r, x >$ associés avec leur fréquence d'apparition où *r* est une relation grammaticale et *x* est une cooccurrence associée avec *w* selon la relation *r*. Par exemple, les triplets de dépendances syntaxiques dans la phrase *« leurs regards recouvraient les eaux du fleuve »* retournés par la chaîne de traitement Macaon sont : *(regard det leur), (recouvrer suj regard), (recouvrer obj eau), (eau det le), (eau dep de)* et *(de obj fleuve)*[16]. Nous pouvons voir les triplets comme des traits syntaxiques : pour le triplet *(recouvrer, suj, regard)*, *regard* a pour trait syntaxique *suj (recouvrer)*. La similarité distributionnelle entre deux mots w_1 et w_2 est définie par la fonction suivante :

$$sim(w_1, w_2) = \frac{2x\ I(F(w_1) \cap F(w_2))}{I\big(F(w_1)\big) + I\big(F(w_2)\big)}$$

$F(w_1)\ et\ F(w_2)$ représentent l'ensemble des traits syntaxiques possédés respectivement par w_1 et w_2. $F(w_1) \cap F(w_2)$ représente l'ensemble des traits syntaxiques communs de w_1 et w_2. Si $I(S)$ est la quantité d'information contenue dans l'ensemble des traits de S alors $I(S) = -\sum_{f \in S} \log P(f)$ où $P(f)$ est la probabilité d'avoir le trait syntaxique f. Cette similarité est bornée entre 0 et 1. Elle retourne 1 si w_1 et w_2 partagent les mêmes traits et retourne 0 si les deux mots n'ont aucun trait en commun. La probabilité $P(f)$ est estimée par le pourcentage des mots qui possèdent le trait syntaxique f parmi l'ensemble des mots possédant la même partie de discours du mot analysé. Sur un ensemble de 30% sélectionné aléatoirement depuis la base de triplets et pour lequel nous avons obtenu 22 168 noms différents, la probabilité d'avoir le

[15] Babelfy, un système de désambiguïsation et de détection d'entités nommées. http://babelfy.org

[16] La relation *det* est spécifique à un nom et son déterminant ; *suj* est la relation entre un verbe et son sujet ; *obj* est la relation entre un verbe et son objet ou autres ; la dernière relation est *dep* pour présenter une relation générique par défaut.

trait syntaxique *suj (border)* est de $\frac{38}{22\,168}$ parce que seulement 38 noms uniques sont utilisés comme *sujet* pour le verbe *border*. La quantité d'information pour ce trait est 6.37. Si on prend l'exemple précédant du nom ***fleuve***, ce nom possède le trait *suj (border)* comme il possède le trait *obj (connaître)*. La probabilité d'avoir *obj (connaître)* est de $\frac{582}{22\,168}$. La quantité d'information retournée est 3.64. Dans ce cas, le trait *suj (border)* est plus informatif que le trait *obj (connaître)*.

3.4 Similarités sémantiques

Pour mesurer la similarité sémantique, nous utilisons l'algorithme de Lesk (1986) et ses variantes proposées il y a près de 30 ans. Cet algorithme est très simple, il considère la similarité entre deux sens comme le nombre de mots, simplement les suites de caractères séparées par des espaces, en commun dans leurs définitions. La partie 3.4.1 présente l'algorithme de base de Lesk, la partie 3.4.2 présente une variante de Lesk que nous utilisons comme baseline et la partie 3.4.3 présente l'algorithme de Lesk étendu.

3.4.1 Algorithme de base de Lesk

Cet algorithme nécessite un dictionnaire (BabelNet pour notre cas) et aucun apprentissage. Il consiste à donner un score à une paire de sens de deux mots différents sans tenir compte ni de l'ordre des mots dans les définitions de ces sens ni d'informations morphologiques ou syntaxiques. Nous faisons une comparaison à partir des traits sémantiques (mots pleins) de chaque définition de sens. Nous utilisons TreeTagger[17] pour obtenir ces trais sémantiques dans notre programme. Comme décrit ci-dessus, BabelNet permet d'offrir plusieurs définitions à un sens pour une langue donnée (français pour nos expériences), nous prenons en compte toutes les définitions possibles. Dans le cas où aucune définition n'est proposée pour un sens, nous prenons en considération les synonymes liés avec le mot à comparer. La fonction utilisée pour mesurer la similarité sémantique se présente par : $Sim_{Lesk}(S_1, S_2) = |D(S_1) \cap D(S_2)|$.

3.4.2 Variante de Lesk

Cette variante consiste à retourner le nombre de mots communs entre les unités lexicales (mots pleins) du contexte du mot à désambiguïser et les traits sémantiques des définitions de chaque sens candidat. Navigli (2009) décrit cette variante. Dans nos expériences, le contexte représente le paragraphe. La fonction utilisée pour mesurer la similarité sémantique se présente par : $Lesk_{Variante} = |contexte(w) \cap D(S_i(\mathrm{w}))|$ où w est le mot à désambiguïser et S_i est l'IIème sens du mot w. Un problème important dans la mesure de Lesk est qu'elle est très sensible aux mots présents dans les définitions. Une absence des mots importants dans les définitions retourne des résultats qui ne sont pas de très bonne qualité. L'une des améliorations proposées pour ce problème est l'algorithme de Lesk étendu.

3.4.3 Algorithme de Lesk étendu

Banerjee et Pedersen (2002) proposent la mesure de Lesk étendu. Cette mesure consiste à calculer le recouvrement entre les mots des définitions des deux sens à comparer mais aussi les mots des définitions issues de différentes relations : *hypernyms, hyponyms, meronyms, holonyms et attribute, similar-to, also-see*[18]. Cette mesure est symétrique : une paire de relation (R_1, R_2) est conservée si et seulement si la paire inverse (R_2, R_1) est présente. Un ensemble de relations possibles est obtenu. Si une relation retourne plusieurs sens, toutes les définitions de ces sens sont conservées. Le recouvrement se calcule par la somme des carrés des longueurs de toutes les séquences de mots de la définition *A* dans la définition *B*. La fonction utilisée pour mesurer la similarité se présente par :

$$Sim_{LeskEtendu}(S_1, S_2) = \sum_{\forall (R_1, R_2) \in Relations^2} |D(R_1(S_1)) \cap D(R_2(S_2))|$$

3.5 Notre approche

Afin de trouver le sens d'un mot dans un paragraphe, nous utilisons d'abord la mesure de similarité distributionnelle de Lin (1998) pour déterminer un score entre le mot cible (mot à désambiguïser) et l'ensemble des mots du paragraphe qui

[17] TreeTagger, outil d'annotation morphosyntaxique. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger

[18] Toutes ces relations sont couvertes dans BabelNet.

appartiennent à la même catégorie grammaticale du mot cible. Cela a pour but de retourner les *k* meilleurs voisins qui ont le plus grand score. Ce calcul repose sur les cooccurrences extraites à partir du corpus de travail. Par la suite nous adaptons une méthode structurelle fondée sur une distance sémantique entre les sens selon une formule proposée par Navigli (2009) :

$$\boldsymbol{S}^* = argmax_{S \in \mathrm{Sens(w)}} \sum_{N_i \in N_w:\, N_i \neq \mathrm{w}} \boldsymbol{max} \quad \mathrm{Score}\,(\mathrm{S}, \mathrm{S}')$$

$S' \in \mathrm{Sens}(N_i)\ avec\ i = 1 \ldots k$ et $N_w = \{N_1, N_2, \ldots, N_k\}$ est l'ensemble ordonné des k voisins les plus proches du mot cible w. $\mathrm{Sens}(N_i)$ est l'ensemble des sens du voisin N_i et $\mathrm{Sens(w)}$ est l'ensemble des sens du mot cible w. $\mathrm{Score}\,(\mathrm{S}, \mathrm{S}')$ est la fonction utilisée pour mesurer la similarité entre deux sens S et S′. Nous utilisons les deux algorithmes présentés ci-dessus (algorithme de base de Lesk et algorithme de Lesk étendu) et nous comparons notre approche par rapport à la variante de Lesk et Babelfy.

Au niveau de la comparaison des définitions de sens de mots, on peut facilement se retrouver avec des définitions trop concises et il est difficile d'obtenir des distinctions de similarité fines. Pour les trois algorithmes utilisés, nous nous servons de l'heuristique suivante une fois on obtient le score final de chaque sens candidat : « dans le cas où deux sens ou plus possèdent le score de similarité le plus grand, le sens retourné est celui qui a le plus grand nombre de connexions sémantiques avec les autres sens du réseau ». Nous obtenons cette information directement dans BabelNet grâce à sa représentation graphique. Il est mentionné que le sens d'un mot qui a le plus de connexions sémantiques est le plus important. Par exemple, si on prend un extrait d'un texte :

*« … Leurs regards recouvraient les eaux du **fleuve**. Je ne bougeais plus. Ils m'indiquaient l'étoile du bonheur, quand mon ciel se couvrait de cumulo-nimbus. Je me suis installé derrière eux, confortablement, et j'ai regardé moi aussi couler le **fleuve** du silence … ».*

Le sens de *fleuve* dans ce texte est *« un cours d'eau naturel recevant des affluents et qui se jette dans une rivière ou dans un autre fleuve »*. Il ne s'agit pas d'un cours d'eau qui se jette dans un océan ou dans une mer. Le bon sens par cet exemple possède 2 026 connexions sémantiques et l'autre sens possède 107 connexions sémantiques. L'algorithme de Lesk de base et Lesk étendu retournent le bon sens, en revanche la variante de Lesk (LeskVariante) se trompe.

4 Données des expériences menées

Le tableau 2 ci-dessous résume d'une part le nombre de mots reconnus ou non dans BabelNet par catégorie (mot plein) d'unité lexicale pour les types (mots différents) et les tokens (ensemble total de mots). D'autre part, les taux de couverture obtenus. La couverture globale présente le rapport entre les mots reconnus dans BabelNet et l'ensemble des mots du corpus d'évaluation. La couverture des mots polysémiques présente le rapport entre les mots polysémiques reconnus dans BabelNet et l'ensemble des mots couverts par BabelNet (l'ensemble des mots monosémiques et polysémiques).

POS	Mots polysémiques		Mots monosémiques		Mots non reconnus		Nombre total		**% couverture globale**		**% couverture mots polysémiques**	
	Tokens	Types	Tokens	Types	Tokens	Types	Tokens	Types	**Tokens**	**Types**	**Tokens**	**Types**
Noms	1 660	590	130	39	51	28	1 841	657	**97,23**	**95,74**	**92,74**	**93,8**
Verbes	1 135	327	31	30	99	47	1 265	404	**92,17**	**88,37**	**97,34**	**91,6**
Adjectifs	353	164	28	19	165	48	546	231	**69,78**	**79,22**	**92,65**	**89,62**
Adverbes	375	68	79	6	33	14	487	88	**93,22**	**84,09**	**82,6**	**91,89**
Total	3 523	1 149	268	94	348	137	4 139	1 380	**91,59**	**90,07**	**92,93**	**92,44**

Tableau 2 : Taux de couverture du corpus d'évaluation par la ressource lexicale BabelNet

Nous avons la meilleure couverture globale pour les noms sur les tokens et les types. Nous atteignons 97,23% en tokens contre 92,17% pour les verbes et 95,74% en types contre 88,37% pour les verbes. La couverture en tokens des verbes polysémiques est forte[19] par rapport à la couverture des noms polysémiques (97,34% contre 92,74%). Parmi les cas de non reconnaissance restants, les erreurs d'étiquetage morphosyntaxique représentent la quasi-totalité des cas (seulement 69,78% en tokens reconnus pour les adjectifs). Les mots pleins qui ne sont pas reconnus sont très peu fréquents.

4.1 Jeu de test

Nous avons choisi les données de notre jeu de test selon leur niveau d'ambiguïté. Nous avons à disposition un corpus d'évaluation pour lequel il est difficile de faire le choix des mots polysémiques selon leur fréquence d'apparition (peu fréquent, fréquent et très fréquent). De ce fait, notre choix porte sur le niveau d'ambiguïté (peu ambigu, ambigu et très ambigu). Nous prenons deux mots polysémiques pour chaque niveau d'ambigüité et cela pour les noms et les verbes. Le tableau 3 ci-dessous résume les informations quantitatives utilisées pour la sélection des mots polysémiques du jeu de test. Nous considérons les mots qui ont moins de 4 sens comme peu ambigus (cf. tableau 3), les mots qui ont entre 4 et 6 sens comme ambigus et les mots qui ont plus de 6 sens comme très ambigus.

POS	Candidat	Fréquence	Nombre de synsets	Niveau d'ambigüité
Noms	**fleuve**	3	3	peu ambigu
	fée	8	3	
	pêcheur	4	4	ambigu
	plante	15	5	
	castor	4	9	très ambigu
	souris	10	9	
Verbes	**planter**	2	3	peu ambigu
	naître	7	3	
	obliger	2	5	ambigu
	taire	9	5	
	troubler	2	7	très ambigu
	parler	6	10	

Tableau 3 : Mots polysémiques du jeu de test avec leur fréquence d'apparition et niveau d'ambigüité

5 Évaluation

Pour mesurer les performances des différentes méthodes de désambiguïsation, nous utilisons le taux d'exactitude (*accuracy*). L'évaluation de nos méthodes est effectuée sur des données dont la couverture des sens par BabelNet est de 100%. Ce taux d'exactitude est calculé pour chaque mot du jeu de test et pour chaque méthode de désambiguïsation testée. Il présente le rapport entre le nombre d'occurrences correctement désambiguïsées et le nombre total d'occurrences d'un mot. L'ensemble des taux d'exactitude obtenus est résumé dans le tableau 4. Notre jeu de test contient 44 occurrences pour 6 noms et 28 occurrences pour 6 verbes (un total de 72 occurrences sur 12 mots différents). Nous avons affecté manuellement à chaque occurrence le bon sens proposé dans BabelNet. Notre évaluation porte d'une part sur le niveau d'ambiguïté des mots polysémiques, d'autre part, sur la mesure distributionnelle utilisée pour choisir les *k-plus proches*

[19] Nous avons une occurrence par verbe monosémique (30/31) et une couverture de 327 verbes polysémiques contre 30 verbes monosémiques.

voisins (*k*-PPV). Notre choix s'est porté sur trois valeurs différentes, $k \in \{3, 5, 7\}$. Nous avons choisi aussi de prendre en compte différentes versions du corpus de travail afin de mesurer le degré de confiance de notre approche en sélectionnant aléatoirement une partie de l'ensemble des triplets de dépendances syntaxiques (30%V_1 pour une première version, 30%V_2, 50%V_1 et 50%V_2) ou la totalité des triplets de dépendances. Le tableau 4 présente les résultats obtenus en tenant compte d'une première sélection de 30% sur l'ensemble des triplets.

Jeu de test	LeskBase	LeskÉtendu		LeskVariante	Babelfy
fleuve	100	100		0	100
fée	0	100		100	87,5
pêcheur	0	0		0	0
plante	86,67	100		80	100
castor	100	100		100	100
souris	30	100		0	30
planter	100	100		100	100
naître	0	85,71		85,71	100
obliger	50	100		50	100
taire	erreur POS	erreur POS		erreur POS	-
troubler	0	0		0	0
parler	16,67	100		16,67	50

Tableau 4 : Taux d'exactitude obtenus par méthode de Lesk de base, Lesk étendu et par sélection aléatoire de 30% (V_1) sur l'ensemble des triplets pour les 5 plus proches voisins et comparaison avec la méthode de Lesk variante et Babelfy sur les données du jeu de test

Les résultats retournés par l'algorithme de Lesk étendu sont intéressants en comparaison avec les autres algorithmes ou ce que Babelfy retourne sur l'ensemble des noms étudiés. Lesk étendu retourne le bon sens pour les noms peu ambigus, Babelfy retourne un sens différent sur une occurrence de *fée* où il a détecté une expression polylexicale *« fée carabosse »*. Pour les noms ambigus, Lesk étendu retourne le bon sens sur toutes les occurrences de *plante* par contre il se trompe sur toutes les occurrences de *pêcheur.* Cela en raison qu'il existe deux sens pour lesquels le score retourné par nos méthodes est le même : (sens 1) *« la pêche est l'activité consistant à capturer des animaux aquatiques dans leur milieu naturel »* ; (sens 2) « *personne dont la profession est d'attraper des poissons »*. Sur un extrait de texte : *« … il fut recueilli par un vieux **pécheur** de saumons … »*, le bon sens de *pêcheur* est le deuxième mais le premier est retourné par nos méthodes vu qu'il possède plus de connexions sémantiques (1 576 contre 355). Pour les noms très ambigus, Lesk étendu ne se trompe pas contrairement à Babelfy. Sur quelques textes décrivant *la souris* comme *« genre d'animaux »*, Babelfy retourne une entité nommée *« MouseHunt »* décrivant un long métrage de Gore Verbinski.

Pour les verbes, il est difficile de juger la sensibilité du taux d'exactitude au niveau d'ambiguïté. D'une part, nous avons des erreurs d'étiquetage (exp. *taire* ne se trouve sur aucun des textes utilisés), d'autre part, le manque des définitions en français dans BabelNet, ce qui permet de retourner dans la plupart des cas le sens le plus fort du verbe étudié dans le réseau malgré l'utilisation des synonymes. Nous remarquons que l'algorithme de Lesk étendu est beaucoup plus régulier par rapport à l'algorithme de base de Lesk ou à la baseline (Lesk variante). Le meilleur taux d'exactitude que nous obtenons sur l'ensemble des mots étudiés est celui retourné par l'algorithme de Lesk étendu (90,91% pour les noms et 57,14% pour les verbes). Lesk étendu est meilleur par rapport à Babelfy et Lesk variante pour la désambiguïsation des noms (taux d'exactitude de 72,73% par Babelfy et 54,55% par Lesk variante) ainsi que pour la désambiguïsation des verbes (taux d'exactitude de 50% par Babelfy et 35,71% par Lesk variante). Dans le tableau 5, nous présentons les résultats obtenus par variation du nombre des voisins les plus proches et par sélection aléatoire ou non (100%) d'un ensemble de dépendances syntaxiques.

Algorithme de Lesk étendu	Noms					Verbes				
	k=3	k=5	k=7	Moyenne	Écart typeP	k=3	k=5	k=7	Moyenne	Écart typeP
Depends30%V1	**90,91**	**90,91**	84,09	88,64	3,21	50	57,14	**60,71**	55,95	4,45
Depends30%V2	84,09	84,09	81,82	83,33	1,07	42,86	39,29	**60,71**	47,62	9,37
Depends50%V1	75	**90,91**	75	80,3	7,5	25	28,57	28,57	27,38	1,68
Depends50%V2	75	75	84,09	78,03	4,29	35,71	32,14	**60,71**	42,85	12,71
Depends100%	84,09	77,27	77,27	79,54	3,21	32,14	**60,71**	**60,71**	51,19	13,47
Moyenne	81,82	83,64	80,45	81,97	1,3	37,14	43,57	54,28	45	7,07
Écart typeP	**6,1**	**6,65**	**3,69**	**3,76**	-	**8,63**	**13,05**	**12,86**	**9,80**	-

Tableau 5 : Taux d'exactitude obtenus par algorithme de Lesk étendu par ensemble de triplets pour *k*-PPV

Nous remarquons que la variation de l'ensemble des triplets de dépendances syntaxiques apporte des résultats différents pour la désambiguïsation lexicale (différence légère pour les noms mais forte pour les verbes suite au manque des définitions en français malgré l'utilisation des synonymes). Les voisins d'un mot étudié changent à chaque fois où on utilise un ensemble de triplets différent. Pour l'exemple de *plante*, nous avons sur un texte les voisins (*bande, feuille, oiseau*) par sélection de 30%V_1 sur l'ensemble des triplets alors que nous obtenons un autre ensemble de voisins (*animal, insecte, oiseau*) par sélection de 30%V_2. Pour les noms et sur la variation des *k* voisins les plus proches, nous obtenons le meilleur taux d'exactitude (90,91%) pour $k \in \{3,5\}$ par rapport au cas où *k=7*. Cela signifie qu'un petit nombre de voisins est nécessaire pour retourner le bon sens pour les noms, ce qui est tout le contraire pour les verbes où le meilleur taux d'exactitude retourné est atteint lorsque $k = 7$. Les résultats montrent qu'on obtient un bon degré de confiance pour les noms (écart type de 3,76) par contre un degré de confiance faible pour les verbes (écart type de 9,80). Les figures 1 et 2 présentent les résultats obtenus par utilisation de l'algorithme de Lesk étendu respectivement sur les noms et les verbes. Les figures 3 et 4 présentent les résultats obtenus pour les différents algorithmes utilisés et Babelfy sur un ensemble précis de dépendances syntaxiques.

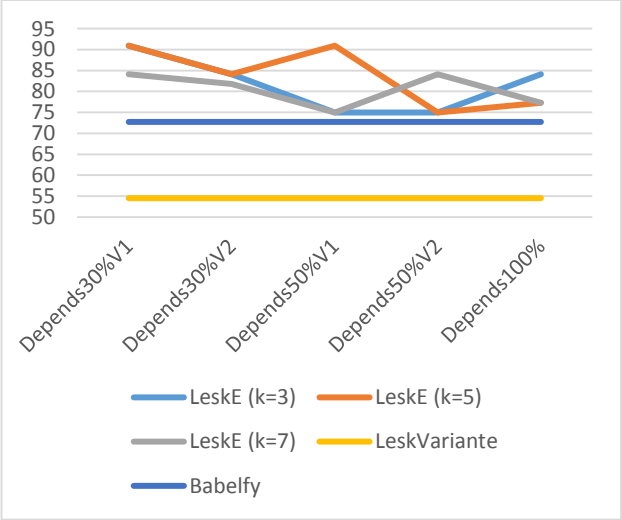

Figure 1: Taux d'exactitude sur les **noms** du jeu de test par utilisation de l'algorithme de Lesk étendu

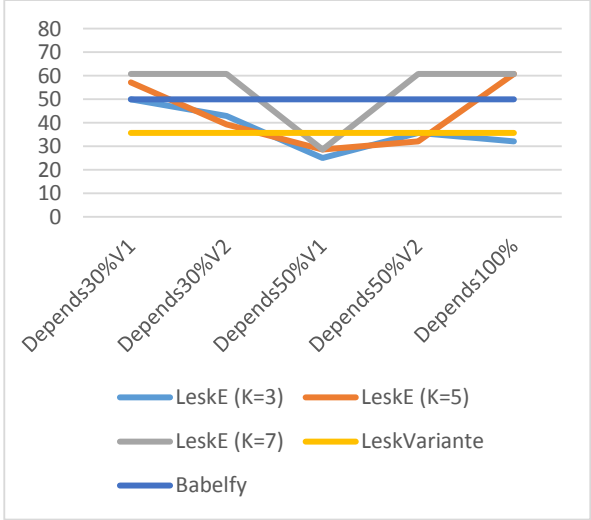

Figure 2: Taux d'exactitude sur les **verbes** du jeu de test par utilisation de l'algorithme de Lesk étendu

L'algorithme Lesk étendu retourne le meilleur résultat par rapport à la baseline et Babelfy pour les noms et cela sur toutes les variations utilisées pour obtenir un ensemble des triplets de dépendances syntaxiques. Pour les verbes, quelques sélections aléatoires des triplets de dépendances apportent à notre approche des résultats faibles par rapport à la baseline et Babelfy. L'utilisation d'une autre mesure de similarité qui ne repose pas sur des traits sémantiques peut corriger ce problème.

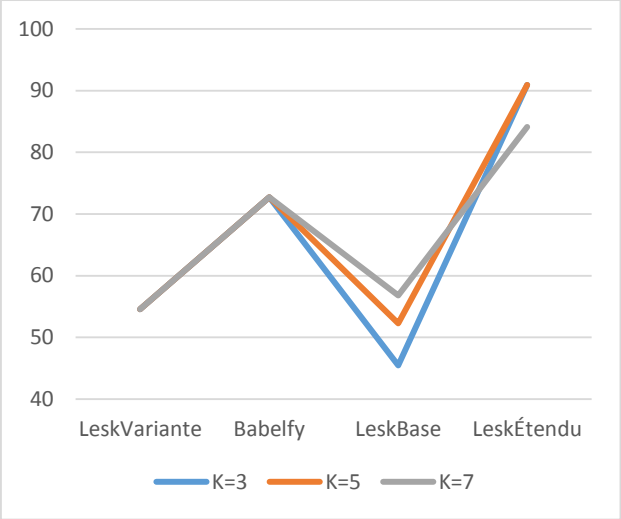

Figure 3: Taux d'exactitude sur les **noms** du jeu de test par sélection aléatoire de 30% sur les dépendances syntaxiques

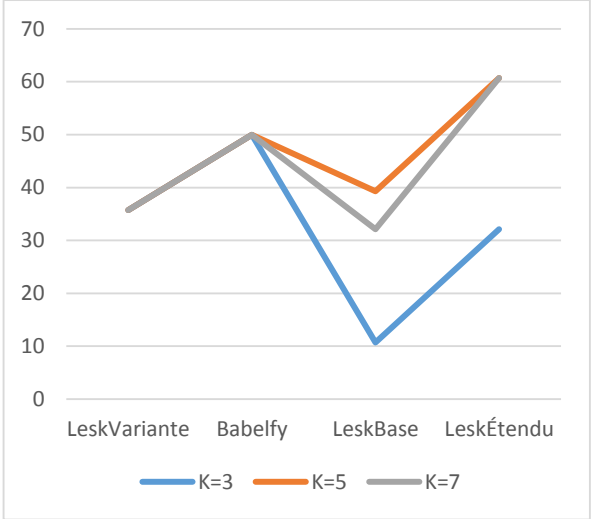

Figure 4: Taux d'exactitude sur les **verbes** du jeu de test sur l'ensemble des dépendances syntaxiques

6 Conclusion et perspectives

Cet article se situe dans le champ de la désambiguïsation lexicale. La méthode que nous avons testée et évaluée s'appuie sur une sélection distributionnelle des voisins les plus proches selon le contexte du mot à désambiguïser. Nous avons adapté l'application d'une approche exhaustive pour se comparer avec *k*-PPV. Cette approche adaptée repose sur l'utilisation de mesures de similarité à base de traits sémantiques. Le contexte utilisé dans cette expérience correspond à un paragraphe et le corpus d'évaluation appartient à un domaine général et non pas à un domaine de spécialité. Le meilleur taux d'exactitude retourné pour les noms est de 90,91% contre 60,71% pour les verbes. La meilleure combinaison retourne 77,78% (k=5 et 30%V_1 de l'ensemble des triplets de dépendances syntaxiques).

Sur le plan des perspectives de ce travail, nous envisageons d'utiliser d'autres algorithmes locaux pour mesurer la similarité sémantique. Par exemple, des algorithmes à base de distance taxonomique qui consistent à compter le nombre d'arcs qui séparent deux sens dans une taxonomie (Wu, Palmer, 1994 ; Hirst, St-Onge, 1998) ou des algorithmes à base de contenu informationnel (Resnik, 1995 ; Seco *et al.*, 2004). Nous envisageons aussi de comparer nos résultats avec d'autres algorithmes globaux comme la recherche des chaînes lexicales (Vasilescu *et al.*, 2004) ou les algorithmes génétiques (Gelbukh *et al.*, 2003).

Remerciements

Nous tenons à remercier Alexis Nasr qui nous a fourni les données du corpus de travail, ainsi que Núria Gala pour sa relecture et son encadrement.

Références

Audibert, L. (2007). Désambiguïsation lexicale automatique : sélection automatique d'indices. *Traitement Automatique des Langues Naturelles* (*TALN*), *Juin 2007, Toulouse, France. IRIT Press,* 13-22.

Banerjee, S., Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. *In CICLing '02, London, UK,* 136–145.

Cowie, J., Guthrie, J., Guthrie, L. (1992). Lexical disambiguation using simulated annealing. *In COLING 92, Stroudsburg, PA, USA. ACL,* 359–365.

Edmonds, P. (2000). Designing a task for SENSEVAL-2. Tech. *note. University of Brighton, Brighton. U.K.* Fellbaum, C. Ed. (1998). WordNet: An Electronic Database. *MIT Press, Cambridge, MA.*

Gelbukh, A., Sidorov, G., Han, S. Y. (2003). Evolutionary approach to natural language word sense disambiguation through global coherence optimization. *WSEAS Transactions on Communications,* 1(2):11–19.

Hanoka, V., Sagot, B. (2012). Wordnet creation and extension made simple: A multilingual lexicon-based approach using wiki resources. *In Proceedings of LREC 2012, Istanbul, Turquie.*

Hearst, M. A. (1991). Noun homograph disambiguation using local context in large corpora. *Proceedings of the 7th Annual Conf. of the University of Waterloo Centre for the New OED and Text Research, Oxford, United Kingdom,* 1-19.
Hirst, G., St-Onge, D. D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet : An electronic Lexical Database. C. Fellbaum. Ed. MIT Press,* 305–332.

Ide, N., Véronis, J. (1998). Word sense disambiguation: The state of the art. *Computat. Ling. 24, 1,* 1–40.
Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries : how to tell a pine cone from an ice cream cone. *In Proceedings of the 5th annual international conference on Systems documentation, SIGDOC '86, New York, NY, USA : ACM,* 24–26.

Lin, D. (1998). An information-theoretic definition of similarity. *In Proceedings of the 15th International Conference on Machine Learning (ICML, Madison, WI),* 296–304.

Miller, G. A., Leacock, C., Tengi, R., Bunker, R. T. (1993). A semantic concordance. *In Proceedings of the ARPA Workshop on Human Language Technology*. 303–308.

Moro, A., Raganato, A., Navigli, R. (2014). Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL), 2,* 231-244.

Nasr, A., Béchet, F., Rey, J. F., Favre, B., Le Roux, J. (2011). MACAON: A linguistic tool suite for processing word lattices. *The 49th Annual Meeting of the Association for Computational Linguistics. ACTI,* 86-91.

Navigli, R., Ponzetto, S. P. (2012). BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence, 193, Elsevier,* 217-250.

Navigli, R. (2009). Word Sense Disambiguation : a Survey. *ACM Computing Surveys 41(2), ACM Press,* 1-69.
Ng, T. H. (1997). Getting serious about word sense disambiguation. *In Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How ? (Washington D.C.),* 1–7.
Ng, H. T., Lee, H. B. (1996). Integrating multiple knowledge sources to disambiguate word sense: An examplar-based approach. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, University of California, Santa Cruz, California,* 40-47.

Pedersen, T., Banerjee, S., Patwardhan, S. (2005). Maximizing Semantic Relatedness to Perform Word Sense Disambiguation. *Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute.*

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *In IJCAI'95, San Francisco, CA, USA,* 448–453.

Seco, N., Veale, T., Hayes, J. (2004). An intrinsic information content metric for semantic similarity in wordnet. *In Proceedings of ECAI'2004, Valencia, Spain,* 1089–1090.

Tchechmedjiev, A. (2012). État de l'art : mesures de similarité sémantique locales et algorithmes globaux pour la désambiguïsation lexicale à base de connaissances. *In Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 3: RECITAL,ATALA/AFCP. June 2012, Grenoble, France,* 295–308.

Vasilescu, F., Langlais, P., Lapalme, G. (2004). Evaluating variants of the lesk approach for disambiguating words. *In Proceedings of LREC 2004, the 4th International Conference On Language Resources And Evaluation, Lisbon, Portugal,* 633–636.

Wilks, Y., Stevenson, M. (1996). The grammar of sense: Is word sense tagging much more than part-of-speech tagging ? *Technical Report CS-96-05, University of Sheffield, Sheffield, United Kingdom.*

Wu, Z., Palmer, M. (1994). Verbs semantics and lexical selection. *In Proceedings of the 32nd annual meeting on ACL, volume 2 de ACL '94, Stroudsburg, PA, USA. Association for Computational Linguistics,* 133–138.

Yarowsky, D (1993). One sense per collocation. *In Proceedings of the ARPA Workshop on Human Language Technology (Princeton, NJ),* 266–271.

Vers un modèle de détection des affects, appréciations et jugements dans le cadre d'interactions humain-agent

Caroline Langlet[1]
(1) Institut Mines-Télécom ; Télécom ParisTech ; CNRS LTCI, Paris
caroline.langlet@telecom-paristech.fr

Résumé. Cet article aborde la question de la détection des expressions d'attitude – i.e affect, d'appréciation et de jugement (Martin & White, 2005) – dans le contenu verbal de l'utilisateur au cours d'interactions en face-à-face avec un agent conversationnel animé. Il propose un positionnement en termes de modèles et de méthodes pour le développement d'un système de détection adapté aux buts communicationnels de l'agent et à une parole conversationnelle. Après une description du modèle théorique de référence choisi, l'article propose un modèle d'annotation des attitudes dédié l'exploration de ce phénomène dans un corpus d'interaction humain-agent. Il présente ensuite une première version de notre système. Cette première version se concentre sur la détection des expressions d'attitudes pouvant référer à ce qu'aime ou n'aime pas l'utilisateur. Le système est conçu selon une approche symbolique fondée sur un ensemble de règles sémantiques et de représentations logico-sémantiques des énoncés.

Abstract.

Toward a detection model of affects, appreciations, jugements within human-agent interactions
This article concerns the detection of attitudes – i.e affects, appreciations and jugements (Martin & White, 2005) – in the user's verbal content during a face-to-face interaction with an embodied conversational agent. It tackles the issue of the adaptation to the ECA's communicational goals and to the conversational speech. After a description of our theoretical model, it introduces an annotation model dedicated to the study of attitudes in a human-agent interaction corpus. Then, it describes a first version of our detection system, focusing on the attitude which can refer to a user's like or dislike. The system is rule-based and embeds logic and semantic representations of the sentences.

Mots-clés : Analyse de sentiments, interaction humain-agent, agents conversationnels animés, dialogue homme-machine.

Keywords: Sentiment analysis, human-agent interaction, embodied conversational agents, dialogue homme-machine.

1 Introduction

Dans le domaine croissant des agents conversationnels animées (ACA), de nombreuses applications permettent de faire interagir des agents virtuels avec des utilisateurs humains. Ces applications se fondent sur des scénarios variés où l'agent peut jouer différents rôles : assistant, tuteur ou encore compagnon. Pour de telles applications, la gestion de la composante affective de l'interaction est cruciale, tant du côté de la génération que de celui de la détection. Il est en effet nécessaire que l'agent puisse détecter les émotions, les sentiments et les attitudes sociales exprimés par l'utilisateur, afin de pouvoir produire des réactions affectives et sociales appropriées. Du côté de la détection, une majorité des travaux se concentrent sur l'analyse d'indices socio-affectifs non-verbaux (expressions faciales, indices acoustiques). Le contenu verbal reste quant à lui encore partiellement exploité. Deux études seulement présentent un système intégrant un module de détection des sentiments dans le contenu verbal pour les interactions humain-agent (Smith *et al.*, 2011; Yildirim *et al.*, 2011). Si ces méthodes proposent une première solution à cette question, les méthodes de détection qu'elles utilisent ne prennent néanmoins pas en compte la nature conversationnelle du contenu verbal qu'elles traitent.

Notre objectif est de développer un système de détection des sentiments exprimés par l'utilisateur au cours d'une interaction multi-modale, en anglais et en face-à-face avec un ACA. Si le rôle de l'ACA n'est pas défini a priori – notre modèle doit être adapté à des ACAs pouvant tenir différents rôles, comme celui d'assistant, de tuteur ou de compagnon – la prise en compte des spécificités énonciatrices d'une conversation humain-agent en face-à-face est néanmoins nécessaire pour

la conception de notre modèle de détection. Tout d'abord, le phénomène linguistique à détecter doit être circonscrit au regard des buts communicationnels de l'ACA et des modèles de relations sociales qu'il intègre. Dans un modèle socio-affectif de l'interaction, les expressions de sentiments pertinentes pour la détection sont celles qui permettront à l'ACA, à court-terme, de réagir de manière appropriée, à long-terme, de maintenir sa relation avec l'utilisateur. La nature de ces expressions sera donc différente de celles majoritairement visées en analyse de sentiments et en fouille d'opinions, où l'objectif est d'obtenir des informations sur des avis de consommateurs ou d'internautes. Cette première contrainte pose également la question du type d'analyse à fournir pour les expressions considérées : les résultats doivent être suffisamment précis pour être exploitables pour la création d'un modèle de préférences de l'utilisateur permettant le calcul de relations sociales ou la gestion de l'engagement de ce dernier dans la conversation. Ensuite, le système de détection doit pouvoir s'adapter à une parole spontanée et conversationnelle : il doit ainsi être en mesure d'en gérer des particularités tant au niveau syntaxique que pragmatique. Sur le plan syntaxique, il est nécessaire de prendre en compte les disfluences – hésitations, répétitions – conférant aux énoncés oraux une organisation syntaxique éloignée de la régularité de l'écrit. Sur le plan pragmatique, le caractère conversationnel du discours pris en charge implique une élaboration sémantique des expressions de sentiments pouvant s'effectuer sur plusieurs tours de parole et en collaboration avec l'agent.

Cet article s'intéresse plus particulièrement à deux de ces contraintes : l'adaptabilité aux buts communicationnels de l'agent et la prise en compte de la parole conversationnelle. Dans un premier temps, les méthodes développées en analyse de sentiments sont mises en regard des contraintes et des objectifs propres à notre cadre applicatif (Section 2). Ce panorama permettra de justifier la méthode et le modèle théorique de référence choisis : une méthode symbolique permettant d'intégrer une modélisation logico-sémantique des énoncés et s'appuyant sur le modèle théorique proposé par (Martin & White, 2005). Dans un deuxième temps, nous présentons une étude exploratoire des attitudes dans le contexte conversationnel de corpus Semaine (McKeown *et al.*, 2011) (Section 3). Enfin, sur la base des résultats de cette analyse, nous décrivons une première version du système reposant sur l'analyse conjointe des énoncés de l'agent et des énoncés de l'utilisateur pour la détection des attitudes de l'utilisateur 4 et 5.

2 Analyse de sentiments : modèles et méthodes face au contexte humain-agent

Le développement d'un système de détection des sentiments est, dans notre cadre applicatif, motivé par la volonté d'améliorer la relation sociale de l'ACA avec l'utilisateur. Dans cette perspective, les analyses fournies par le système doivent être suffisamment précises pour aider l'agent à se positionner et à réagir de manière appropriée face aux expressions détectées. Ainsi, le système doit tout d'abord être en mesure de détecter des expressions individualisées, dont la source est l'utilisateur. En effet, dans la majorité des cas, les sentiments ne pouvant être attribués à l'utilisateur ne seront que d'une utilité limitée pour le système. Ensuite, la catégorisation doit être suffisamment fine pour qu'il soit possible de cibler les phénomènes les plus appropriés aux besoins de l'agent. Enfin, le système doit intégrer une analyse précise des propriétés caractéristiques des expressions de sentiments : le calcul de la polarité doit être solide et la cible clairement identifiée afin que l'agent ne commette pas d'erreurs dans ses prises de décision et dans sa gestion de l'interaction. En vue de répondre à ces questions, cette section propose une synthèse des modèles linguistiques de référence et des méthodes utilisées en analyse de sentiments. L'objectif est de cibler dans l'existant les méthodes et modèles appropriés à nos objectifs et d'en définir les limites afin de proposer une adaptation pour l'analyse de sentiments dans le contexte conversationnel.

2.1 Un modèle théorique adapté aux buts communicationnels de l'ACA

La conception d'un système de détection des sentiments implique généralement de s'appuyer sur une modélisation – même minimale – du phénomène tel qu'il s'exprime dans le discours. Concernant le développement d'un tel système dans le cadre d'interactions humain-agent, le modèle linguistique doit répondre à certains critères. Du point de vue de l'agent et selon son modèle de relations sociales, toutes les expressions verbales de sentiments ne seront pas pertinentes pour la détection. Dans certains scénarios, l'ACA peut avoir besoin d'informations concernant des expressions plus affectives (« I'm sad », par exemple), tandis que pour d'autres, l'intérêt portera davantage sur des expressions exprimant un jugement de valeur (« This painting is beautiful »). Ainsi, le modèle théorique doit fournir une typologie détaillée des sentiments dans le langage, proposant une hiérarchie complexe de catégories. Ensuite, il doit également modéliser les phénomènes d'expression de sentiments comme un processus évaluatif impliquant une source et une cible, la détection de ces deux éléments étant indispensable pour une exploitation efficace. Enfin, afin de faciliter le développement de ressources linguistiques (lexiques, patrons d'extraction), ce modèle doit s'accompagner d'une description linguistique précise des réalisations verbales de sentiments. Parmi les modèles présentés ci-dessous, le modèle proposé par (Martin &

White, 2005) nous est apparu comme le plus approprié.

Oppositions subjectif-objectif, positif-négatif De nombreux travaux en analyse de sentiments se concentrent sur une opposition objectif-subjectif pour classer des textes, des phrases (Wiebe & Riloff, 2005) ou des groupes syntaxiques (Wilson *et al.*, 2004). Cette opposition est également sollicitée lors de la constitution de ressources linguistiques, qu'elles soient lexicales ou syntaxiques. Alors que (Esuli & Sebastiani, 2005) constituent un lexique, dans (Riloff & Wiebe, 2003), les auteurs décrivent une méthode permettant d'apprendre automatiquement des patrons de phrases subjectives. A cette distinction subjective-objective, s'ajoute fréquemment une distinction relative à la polarité des expressions considérées comme subjectives. Cette distinction peut être établie en référence au modèle d'Osgood (Osgood *et al.*, 1975) ou des *Private States* (Quirk, 1985), dont il n'est retenu que cette notion d'axe de valence. Ainsi dans (Turney, 2002) et (Hu & Liu, 2004), les auteurs proposent de classer ou de résumer des revues d'internautes selon leur orientation sémantique. Dans certains cas, la granularité choisie peut être plus fine. Ainsi, dans (Nasukawa & Yi, 2003), les auteurs présentent un algorithme pour classer les phrases selon leur caractère favorable-défavorable, tandis que dans (Wilson *et al.*, 2005), le focus est mis sur l'attribution d'une polarité à des groupes syntaxiques. Dans un contexte humain-agent, une classification en termes de valence peut apporter des informations essentielles. Néanmoins, n'offrant pas de catégorisation précise des expressions de sentiments, cette approche ne peut suffire à elle seule.

Modèles psychologiques Afin de pouvoir classer plus précisément les énoncés ou les items lexicaux, certains travaux reprennent des classifications fournies par des modèles développés en psychologie cognitive. Ainsi (Neviarouskaya *et al.*, 2007) et (Neviarouskaya *et al.*, 2010a) reprennent les 9 classes d'émotions proposée par (Izard, 1977) : joie, dégoût, peur, colère, tristesse, surprise, honte, intérêt, culpabilité. D'autres travaux (par exemple (Ishizuka, 2012)) choisissent de s'appuyer sur la classification du modèle OCC (Ortony, Clore and Collins (Ortony *et al.*, 1990)). Si ces modèles offrent l'avantage d'une classification plus complexe et détaillée, ils restent néanmoins lacunaires quant à la description proprement linguistiques des expressions de sentiments. Leur objectif n'est pas de décrire des réalisations verbales mais bien d'appréhender des mécanismes psychologiques. Une telle description est cependant nécessaire pour le développement de notre système : elle sera un outil théorique indispensable à la conception de ressources linguistiques (lexiques, patrons d'extractions). Pour cette raison, nous avons décidé de fonder notre système de détection sur un modèle de référence plus orienté langage, celui décrit dans (Martin & White, 2005).

Affects, appréciations et jugements dans le langage Ce modèle, issu de la linguistique systémique fonctionnelle, fournit une description détaillée des réalisations verbales de sentiments. Son utilisation dans certains travaux d'analyse de sentiments (Neviarouskaya *et al.*, 2010b; Bloom *et al.*, 2007; Whitelaw *et al.*, 2005) a permis de démontrer son adaptabilité aux problèmes de modélisation des expressions de sentiments. Le modèle présente une hiérarchie complexe des *attitudes* divisées en trois sous-classes : les *affects*, qui réfèrent à des réactions émotionnelles ; les *jugements*, qui réfèrent à des évaluations axiologiques de comportements humains ; les *appréciations* qui expriment des évaluations d'artefacts ou d'événements naturels. Dans le cadre de ce modèle, les expressions d'attitudes sont décrites comme reposant sur trois éléments : la source, la personne évaluant ou expérimentant l'attitude, la cible, l'entité ou processus évalué, et enfin, l'expression linguistique permettant de référer à l'attitude en question. Du point de vue humain-agent, le modèle de (Martin & White, 2005) apparaît comme le plus approprié. Il cumule les avantages des modèles psychologiques – classification précise et modélisation des sources et des cibles – tout en gardant un point de vue linguistique sur le phénomène, facilitant ainsi la formalisation symboliques d'expressions pouvant intégrer un modèle de détection. De plus, le grand intérêt qu'il accorde à la modélisation des propriétés de ce phénomène – comme l'intensité ou l'engagement – garantit à terme la possibilité d'intégrer l'analyse de ces dernières par le module de détection.

2.2 Des méthodes de classification vers des analyses à grain fin

L'adaptabilité aux buts communicationnels de l'ACA doit déterminer la nature de l'analyse fournie par le module de détection : une analyse à grain fin capable (i) de distinguer différentes expressions d'attitudes au sein d'une même phrase, (ii) d'identifier la source et de la cible de chaque expression, (iii) de déterminer précisément la polarité. Enfin, le processus d'analyse devra être suffisamment modulaire pour être adaptable au contexte conversationnel. Cette section propose une présentation des différentes méthodes développées pour la détection de sentiments dans les textes et interroge la possibilité de leur adaptation aux interactions humain-agent. Parmi l'ensemble des méthodes proposées en analyse de sentiments, il est possible de distinguer celles optant pour un niveau de granularité haut et faisant une utilisation assez minimale de

modélisation linguistique, de celles s'intéressant à un niveau de granularité plus bas – celui de l'expression de sentiments – et intégrant des représentations symboliques des énoncés exprimant des sentiments.

Méthodes statistiques et d'apprentissage automatique pour la classification des sentiments Un grand nombre de travaux proposés en analyse de sentiments se fondent sur des méthodes statistiques ou d'apprentissage automatique et ont pour objectif de classifier des items lexicaux, des textes ou des phrases. Les premiers travaux proposés ont ainsi majoritairement porté sur la classification d'items lexicaux. L'objectif est ainsi de distinguer des adjectifs objectifs d'adjectifs subjectifs. Ainsi dans (Hatzivassiloglou & McKeown, 1997), les auteurs cherchent à prédire l'orientation d'adjectifs conjoints, i.e. des adjectifs reliés par un conjonction de coordination, en exploitant un modèle de régression logistique. Rapidement ont également émergé des méthodes pour la classification de textes – généralement des revues de produits ou de films. Ainsi, dans (Pang & Lee, 2004), les auteurs comparent deux types de classifieurs pour déterminer automatiquement l'orientation positive ou négative des revues de films : SVM (Support Vector Machines) et NB (naive bayes). Si les systèmes de classification de textes subjectifs ont montré des résultats intéressants, de nombreux travaux ont néanmoins eu l'ambition de descendre au niveau de la phrase afin de ne plus traiter le texte de sa globalité. Là encore, différents algorithmes de classification par apprentissage automatique ont pu être utilisés : *boostrapping* (Wilson *et al.*, 2005; Riloff & Wiebe, 2003; Wilson *et al.*, 2004), *classifieur naïf bayésien* (Wiebe & Riloff, 2005), *rule learning* (Wilson *et al.*, 2004), *machine à vecteur de support* (Wilson *et al.*, 2004), *conditional random fields* (Breck *et al.*, 2007).

L'ensemble des méthodes présentées ici permettent de résoudre un certain nombre de difficultés liées à la détection de sentiments et d'opinions dans les textes. Dans le cadre de conversation humain-agent, elles peuvent permettre d'attribuer une polarité globale au tour de parole de l'utilisateur. Néanmoins, davantage développées pour des problématiques de fouille d'opinion dans des larges corpus, leur niveau de granularité est trop haut pour répondre à l'ensemble de nos problèmes. Tout d'abord, l'agent doit pouvoir distinguer les différentes expressions d'attitudes produites au sein d'un même tour de parole afin de se positionner clairement face à chacune d'elles. Ensuite, l'exploitation des données issues de la détection pour la création d'un modèle utilisateur exige un niveau de précision en termes de calcul de polarité que ne fournissent pas ces méthodes. Le calcul qu'elles effectuent ne prend généralement pas en compte le traitement des modificateurs de valence et lorsque cela est fait, celui-ci reste minimal. Enfin, la source et la cible sont rarement prises en charge.

Méthodes symboliques et hybrides pour l'analyse profonde des expressions de sentiments Les méthodes à grain fin semblent davantage répondre aux contraintes posées par notre cadre applicatif. Ces travaux intègrent généralement des représentations formelles des énoncés. Celles-ci sont exploitées soit par des algorithmes à base de règles, soit par des algorithmes hybrides associant analyse profonde de la phrase et méthode d'apprentissage. L'intérêt de telles méthodes est de pouvoir accorder une plus grande importance à l'analyse de propriétés intrinsèques des expressions de sentiments et d'opinions.

Tout d'abord, la prise en compte de la structure logico-sémantique des énoncés leur permet de gérer le principe de compositionnalité sémantique et d'améliorer ainsi sensiblement le calcul de la polarité. La prise en compte de ce principe, déterminant le sens d'un énoncé comme composé du sens de l'ensemble de ses constituants et de leurs relations hiérarchiques, permet de définir un certain nombre de règles de calcul de la polarité s'appuyant sur une représentation symbolique des structures logico-sémantiques. Ainsi les travaux présentés dans (Neviarouskaya *et al.*, 2010b) et (Moilanen & Pulman, 2007) proposent des méthodes de détection à base de règles, se fondant sur une analyse des relations de dépendance entre constituants et permettant un calcul fin de la polarité des expressions de sentiments. Ils modélisent ainsi des règles de propagation ou d'inversion pour résoudre des conflits de polarité à différents niveaux de la structure syntaxique. Cette approche est également adoptée dans (Shaikh *et al.*, 2009). Les auteurs y offrent une interprétation du modèle OCC (Ortony *et al.*, 1990). Sur la base de cette interprétation, les auteurs définissent des règles linguistiques pour la détection d'expressions référant à des émotions et le calcul précis de la polarité.

Au-delà d'un calcul plus fin de la polarité, ces méthodes ont également l'avantage d'être plus optimisées pour la détection des sources et des cibles des expressions d'opinions ou de sentiments. Ainsi, dans (Choi *et al.*, 2005), les auteurs proposent une méthode de détection des sources des opinions utilisant conjointement les CRF (*conditional random fields*) et des patrons d'extraction acquis via AutoSlog (Riloff, 1996). Ce type d'approche a également été exploité par (Yang & Cardie, 2013) pour la détection conjointe des sources et des cibles des expressions d'opinions. Là encore, les auteurs associent CRF et patrons syntaxiques.

Pour une première approche de la détection des attitudes dans le cadre de conversations humain-agent, nous avons fait le choix d'une méthode symbolique. Si les méthodes hybrides présentent des avantages en termes de coût de développement

et de rapidité de traitement, leur application à un contexte d'interaction humain-agent nécessite néanmoins une validation des ressources linguistiques qu'elles emploient dans le cadre d'une parole conversationnelle. Le développement d'une méthode symbolique permettra ainsi de pouvoir élaborer et valider un ensemble de règles linguistiques concernant tant le niveau logico-sémantique que pragmatique. Celles-ci pourront être exploitées à plus long-terme par une méthode hybride.

3 Les attitudes en contexte conversationnel : modèle d'annotation et exploration du corpus Semaine

Cette section décrit une étude exploratoire de corpus, visant à appréhender les attitudes telles qu'elles se manifestent dans le contexte d'une parole conversationnelle. Cette étude a pour objectif de servir de base au développement des règles linguistiques utilisées par le système. Dans un premier temps, nous présentons un modèle d'annotation (Section 3.1) s'attachant à décrire la relation entre les attitudes exprimées par l'utilisateur et les énoncés de l'agent. Dans un second temps, nous détaillons les résultats de l'application de ce modèle au corpus Semaine (Section 3.2).

3.1 Modéliser la relation des attitudes de l'utilisateur avec le contexte antérieur

Afin de pouvoir modéliser les expressions d'attitudes de l'utilisateur dans le contexte d'une parole conversationnelle, nous sommes partis du postulat développé en théorie de la conversation, considérant une partie des énoncés produits dans le cadre conversationnel comme des actes participatifs ou collectifs, fonctionnant comme des contributions au discours (Clark & Schaefer, 1989). En tant qu'énoncés à part entière, les expressions d'attitudes peuvent être considérées comme des formes de contribution intrinsèquement liées au déroulement de la conversation : motivées par ce qui a été énoncé au préalable, elles ont également une influence sur ce qui sera dit par la suite. En guise de première approche, le modèle d'annotation présenté ici s'intéresse aux liens énonciatifs que les attitudes de l'utilisateur entretiennent avec le contexte antérieur de la conversation. Plus particulièrement, il s'agit d'appréhender la manière dont l'agent peut influencer et déterminer les expressions d'attitude chez l'utilisateur. Pour cela, le modèle d'annotation considère à la fois le contenu verbal de l'utilisateur et celui de l'agent. Formellement, le modèle intègre des unités (« séquences d'élément textuels adjacents »), des relations (« rapports binaires entre deux unités ») et des schémas (« configurations textuelles complexes récurrentes impliquant unités et relations » (Widlöcher & Mathet, 2012)).

Les actes illocutoires des énoncés de l'agent Des labels spécifiques ont été définis à la fois pour les attitudes exprimées par l'utilisateur et pour les énoncés de l'agent. Afin de prendre en compte la manière dont l'agent collabore et influence la production d'attitudes chez l'utilisateur, les énoncés de l'agent sont annotés selon l'acte illocutoire qu'ils performent. Pour cela, nous utilisons la classification établie par Searle (Searle, 1976) qui inclut cinq catégories : les actes assertifs qui engagent le locuteur sur la véracité de son propos (par exemple, « It's raining ») ; les actes directifs qui tentent d'obtenir quelque chose de l'interlocuteur (par exemple, « I order you to leave ») ; les actes commissifs qui engagent le locuteur sur des événements futurs (« I promise to pay you the money ») ; les actes expressifs qui expriment l'état mental du locuteur à propos de quelque chose (« I apologize for stepping on your toe ») ; les actes de déclaration dont la performance permet de faire correspondre le contenu propositionnel à un état de fait du monde (par exemple, « You're fired »). Pour étiqueter chaque énoncé de l'agent, nous utilisons l'unité *Enoncé Agent*, à laquelle nous ajoutons un trait spécifiant la nature de l'acte illocutoire. Un même tour de parole peut ainsi contenir plusieurs énoncés réalisant différents actes illocutoires.

Annotation des attitudes Le modèle prend compte les attitudes exprimées dans les énoncés de l'utilisateur et dans ceux de l'agent. Une expression d'attitude est composée de trois éléments qu'il est nécessaire de pouvoir annoter : l'indice linguistique exprimant le caractère évaluatif ou affectif de l'énoncé, la source et la cible. Des informations à propos du type d'attitude et de la polarité doivent également être spécifiées. Le type de l'attitude et la polarité (positive ou négative) sont indiqués grâce à une structure de traits associée soit au schéma utilisateur – décrit ci-dessous – soit à l'unité *Énoncé Agent*. Concernant le type d'attitude, la catégorie *affect* est conservée telle quelle, tandis que les catégories *appréciation* et *jugement* sont regroupées au sein de la catégorie *évaluation*. Les unités d'annotation *Source* et *Target* sont utilisées pour annoter les groupes syntaxiques référant à la source et à la cible des attitudes lorsqu'elles sont exprimées. Dans le but de vérifier l'influence de l'agent sur les expressions d'attitudes des l'utilisateur, la cible du côté utilisateur est reliée à la cible du côté agent lorsqu'elles réfèrent à la même entité. L'indice linguistique exprimant le caractère évaluatif ou affectif de

l'énoncé (*Indice Attitude*) est uniquement pris en compte dans le cas des attitudes exprimées par l'utilisateur. Concernant l'agent, le trait spécifiant le type d'attitude (*Type Attitude*) est suffisant pour indiquer si l'énoncé véhicule une expression d'attitude. L'unité d'annotation *Indice Attitude* est appliquée au niveau du syntagme et couvre à la fois les lexies référant à une attitude mais aussi les modifieurs de valence. Par exemple, dans une phrase comme « I don't really like my work », « don't really like » est annoté comme *Indice Attitude*, incluant le marqueur de négation « don't »

Relier les attitudes de l'utilisateur aux énoncés de l'agent A un niveau de granularité plus haut, notre modèle d'annotation se compose de deux types de schémas : des schémas utilisateur et des schémas agent. Chaque *Schéma Agent* est composé d'une unité *Enoncé Agent* et des éventuelles unités *Source* et *Cible* auxquelles elle peut être reliée. De même, un *Schéma Utilisateur* est composé de l'unité *Indice Attitude* et des unités *Source* et *Cible* auxquelles elle est éventuellement associée. Afin de pouvoir modéliser la relation entre les expressions d'attitudes de l'utilisateur et les énoncés de l'agent, chaque *Schéma Utilisateur* est relié au *Schéma Agent* qui le précède.

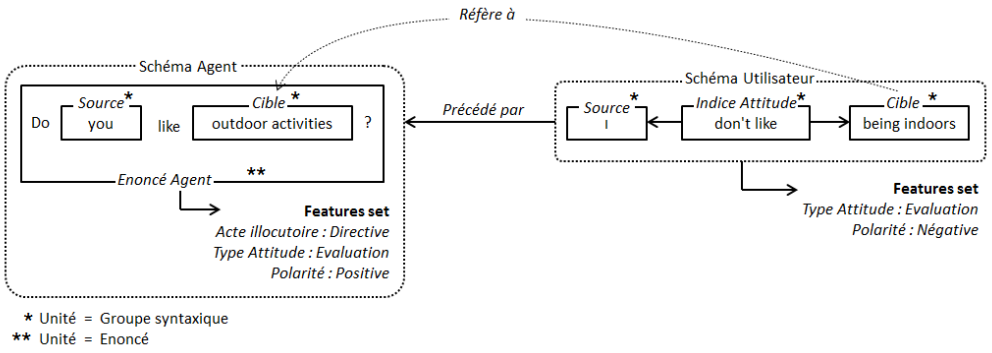

FIGURE 1 – Modèle d'annotation

3.2 Application sur un corpus d'interaction humain-agent

Corpus Semaine Le corpus Semaine (McKeown *et al.*, 2011) est composé de 65 transcriptions manuelles de sessions où un utilisateur humain interagit, en anglais, avec un opérateur humain jouant le rôle d'un agent conversationnel. Ces interactions sont fondées sur un scénario impliquant quatre personnages d'agent émotionnellement typés : Poppy, joyeuse et extravertie, Prudence, sensible et raisonnable, Spike, colérique et conflictuel, et enfin Obadiah, dépressif et maussade. Les énoncés que peuvent prononcer les opérateurs sont contraints par un script – néanmoins, certains énoncés dans le corpus s'écartent de ce script – ayant l'objectif de pousser l'utilisateur vers le même état émotionnel que celui du personnage joué par l'opérateur humain. Au total, pour notre étude, 16 sessions (4 pour chaque personnage) ont été annotées manuellement. Pour l'annotation nous avons utilisé la plate-forme Glozz (Widlöcher & Mathet, 2012). Dans la mesure où il s'agit d'une étude exploratoire et non de la constitution d'un corpus annoté pour l'apprentissage d'un algorithme supervisé, un seul annotateur a procédé à l'annotation. Les sessions annotées ont un nombre variable de tours de parole (132 pour la session la plus longue, 38 pour la session la plus courte). Parmi les 450 schémas agent annotés, 46% expriment un acte directif, 42% un acte expressif et 12% un acte représentatif. Aucun acte déclaratif ou commissif n'a été trouvé. Cela est probablement dû à la nature du scénario sur lequel est fondé le corpus Semaine : une conversation narrative où l'utilisateur est poussé à parler de sa vie. Parmi l'ensemble des schémas agent, 339 réfèrent à une attitude (187 à un affect, 152 à une évaluation). Du côté de l'utilisateur, 238 schémas ont été annotés : 44% sont notés comme affect, 56% comme évaluation (jugements et appréciations).

Quelle est l'influence des énoncés de l'agent sur l'expression des attitudes de l'utilisateur ? Au regard de l'annotation manuelle effectuée sur le corpus Semaine, il est possible d'avoir un premier aperçu de l'influence de l'agent sur les expressions d'attitudes de l'utilisateur. Tout d'abord, sur un plan pragmatique, la nature de l'acte illocutoire précédant les expressions d'attitude de l'utilisateur est un premier indicateur. En effet, les attitudes exprimées par l'utilisateur sont majoritairement précédés par des actes directifs de type interrogatif. Parmi ces questions, certaines concernent directement des attitudes de l'utilisateur. Ensuite, sur le plan du contenu propositionnel, il est à noter que, lorsqu'une attitude de

l'utilisateur est précédée par un énoncé de l'agent contenant lui aussi une expression d'attitude, les polarités concordent généralement. De plus, concernant les cibles, un quart des cibles évaluées par l'utilisateur réfèrent à une cible évaluée par l'agent.

4 Développement d'un système de détection adapté au modèle de relations sociales de l'ACA : focus sur les *likes* et les *dislikes*

Cette section décrit la première version de notre système de détection. Celle-ci a été décrite de manière exhaustive dans (Langlet & Clavel, 2015). Cette section et la suivante présentent donc une synthèse de son principe de fonctionnement et de son évaluation. Cette première version du système restreint volontairement le type d'attitudes à détecter. La délimitation est définie relativement à l'une des dimensions utilisées pour modéliser les relations sociales de l'ACA, le *liking*. La définition de ce concept est fondée sur la théorie de Heider – *Heider's Balance Theory* (Heider, 1958) qui envisage la manière dont les relations entre deux personnes, impliquant des entités impersonnelles, s'équilibrent relativement à la relation que chacune de ces deux personnes entretient avec ces entités individuelles. La théorie de Heider est utilisée par les modèles d'agents sociaux définissant des scénarios où la nature de la relation (relation de *liking*) entre l'agent et l'utilisateur est déterminée par leur goût (*liking*) pour d'autres entités (entités inanimées, processus ou événements). Afin de pouvoir fournir des informations nécessaires au calcul du liking, notre système est conçu pour détecter les expressions d'attitudes pouvant référer à ce que l'utilisateur aime ou n'aime pas (*likes* et *dislikes*). Ces expressions d'attitudes peuvent trouver leur place dans les trois catégories – affect, appréciation, jugement – définies dans le modèle de (Martin & White, 2005). Par exemple, si les phrases « This painting makes me sad » (« Cette peinture me rend triste ») et « this painting is a master-work » (« cette peinture est un chef-d'oeuvre ») appartiennent respectivement aux classes *affect* et *apprécition*, elles réfèrent également, chacune de manière différente, à ce qu'aime ou n'aime pas l'utilisateur.

Pour détecter les expressions d'attitudes exprimant un *like* ou un *dislike*, le système se fonde sur une analyse conjointe des énoncés de l'agent et de ceux de l'utilisateur. En effet, si l'analyse du corpus Semaine a montré un lien entre la forme et le contenu propositionnel des énoncés de l'agent et l'expression d'attitude de l'utilisateur, elle a également révélé l'importance de la prise en compte des énoncés de l'agent pour la détection des attitudes de l'utilisateur. Ainsi, un énoncé produit par l'utilisateur peut contenir l'expression d'une attitude sans qu'un indice lexical ne soit utilisé. Par exemple, dans le cadre d'un échange du type, Agent « Do you like this painting ? » (« Aimez-vous cette peinture ? ») – Utilisateur « Yes » (« Oui »), la seule analyse de l'énoncé de l'utilisateur ne permet pas d'y déceler l'expression d'une attitude. Afin de pouvoir gérer ces difficultés, pour chaque énoncé de l'utilisateur analysé, le système analyse parallèlement l'énoncé de l'agent auquel il répond. Du côté de l'agent, le système cherche des expressions d'attitude pouvant être confirmées et infirmées par l'utilisateur. Cette analyse ne peut se faire qu'automatiquement : dans la plate-forme ACA que nous utilisons, Greta (Poggi *et al.*, 2005), les énoncés de l'agent ne sont pas produits automatiquement, mais scriptés manuellement. Il n'est donc pas possible de s'appuyer sur des ressources de génération pour obtenir des informations sur la présence d'une expression d'attitude. Du côté de l'utilisateur, le système cherche des expressions de confirmations ou d'infirmations ainsi que des attitudes pleinement formulées.

4.1 Analyse de l'énoncé de l'agent

Dans l'énoncé de l'agent, le système cherche à détecter des attitudes exprimées sous la forme d'une affirmation ou d'une question fermée, pouvant référer à un *like* ou *dislike* et dont la source peut être l'agent ou l'utilisateur. Trois niveaux sont considérés : un niveau lexical, un niveau syntagmatique, et un niveau phrastique. Au niveau lexical, après une tokenisation et un POS-tagging, le système vérifie la présence d'un indice lexical d'attitude à l'aide du lexique WordNet-Affect (Valitutti, 2004). Trois parties de discours sont considérées : les noms, les adjectifs et les verbes. La définition des affects proposée par WordNet-Affect étant plus large que celle de Martin et White (Martin & White, 2005), le lexique intègre notamment des lexies pouvant référer à des appréciations et à des jugements. Afin néanmoins d'adapter cette ressource à nos objectifs de détection, nous avons procédé à une sélection des lexies les plus pertinentes. Parmi l'ensemble des synsets, nous avons conservé ceux pouvant référer aux concepts de *like* et de *dislike* et appartenant, dans la taxinomie de WordNet-Affect, à ces trois catégories : *positive-emotion*, *negative-emotion*, *neutral-emotion*. Une fois l'analyse lexicale effectuée, si un indice d'attitude est trouvé, le système lance l'analyse du niveau syntagmatique. Ce niveau est géré par une grammaire formelle implémentée sous forme d'automates via la plate-forme Unitex (Paumier, 2015). Les règles définies par la grammaire sont fondées sur les patrons permettant de reconnaître trois types de syntagmes : des syntagmes

verbaux, adjectivaux ou nominaux. Cette phase applique également un certain nombre de règles sémantiques permettant de calculer la polarité attribuée au syntagme lorsque celui-ci comporte une lexie étiquetée comme indice d'attitude. Le niveau phrastique permet de vérifier si la structure logico-sémantique de la phrase de l'agent correspond à l'expression d'une attitude référant à un *like* ou un *dislike* de modalité affirmative ou interrogative et ayant une forme attributive ou processive. Là encore, l'analyse emploie une grammaire formelle implémentée sous forme d'automates. Les figures 2 et 3 présentent une version simplifiée des règles supérieures des grammaires respectivement dédiées à la détection de ces deux formes syntaxiques d'expressions d'attitudes. Les non-terminaux $SyntVb$, $SyntNoun$ et $SyntAdj$ correspondent aux syntagmes verbaux, nominaux et adjectivaux détectés lors de la phase précédente de l'analyse. Leur trait sémantique att à une valeur égale à $true$ lorsqu'ils comportent une lexie référant à une attitude. Les arguments int et aff indiquent si l'expression est de modalité interrogative ou affirmative.

`Att(aff) -> Src(usr|agt),SyntVb(cop), SyntAdj(att:true), Target.` `Att(int) -> Aux, Source(usr), SyntVb(cop), SyntAdj(att:true), Target.`	*I am really happy to do that* *Are you really happy to do that?*
`Att(aff) -> Target, SyntVb(make), Src(usr|agt), SyntAdj(att:true).` `Att(int) -> Aux, Target, SyntVb(make), Src(usr|agt), SyntAdj(att:true).`	*This book makes me sad* *Does this book make you sad?*
`Att(aff) -> Src(usr|agt), SyntVb(have), SyntNoun(att:true).` `Att(int) -> Aux, Src(user), SyntVb(have), SyntNoun(att:true).`	*I had an awful week* *Did you have an awful week?*
`Att(aff) -> Target, SyntVb(cop), [SyntNoun|SyntAdj]`	*This book is amazing*
`Att(aff) -> SyntNoun(PronDem), SyntVb(cop),[SyntNoun(att:true)|SyntAdj(att:true)].`	*It is amazing to do that*
`Att(aff) -> SyntNoun(PronDem),SyntVb(cop),[SyntNoun(att:true)|SyntAdj(att:true)], "for"|"of", Target, InfClause.`	*It is silly of them to do that*
`Att(aff) -> Src(usr|agt), SyntVb(opinion), Target, "as"|"like", [SyntNoun(att:true)|SyntAdj(att:true)].` `Att(int)-> Aux, Src(usr), SyntVb(opinion),Target,"as"|"like", [SyntNoun(att:true)|SyntAdj(att:true)].`	*I consider this painting as beautiful* *Do you consider this book as beautiful?*

Règle de polarité

$\texttt{If Neg}_{\texttt{SyntVb}} == \texttt{True : Att(pol:inv(Pol}_{\texttt{SyntAdj}}\texttt{|Pol}_{\texttt{SyntNoun}}\texttt{))}$
$\texttt{Else: Attitude(pol:Pol}_{\texttt{SyntAdj}}\texttt{|Pol}_{\texttt{SyntNoun}}\texttt{)}$

Dans ce type de phrase, la valeur attitudinale est portée par le syntagme attribut (nominal ou adjectival). Quand le verbe a une forme négative la polarité attribuée à l'expression est l'inverse de celle portée par le syntagme attitudinalement marqué.

FIGURE 2 – Règles du niveau phrastique prenant en compte des phrases de type attributif

Le niveau phrastique permet également de vérifier la nature de la source (agent ou utilisateur) et de catégoriser la cible. Pour la source, elle est assimilée à l'agent lorsque sa forme correspond à un pronom de la première personne ($Src(agt) \rightarrow$ "I"|"me") et à l'utilisateur lorsque forme correspond à un pronom de seconde personne ($Src(usr) \rightarrow$ "you'). Pour les cible, le système ne procédant pas à une résolution des anaphores, il n'est capable de leur assigner que des classes génériques. Les deux premières sont relatives aux deux membres de la conversation : l'agent et l'utilisateur. La troisième, appelée *other*, implique toutes les entités ou processus qui ne sont ni l'agent ni l'utilisateur. La dernière, appelée *unknown*, concerne des entités ou processus dont la classe même générique ne peut être connue. Enfin, lorsqu'une expression d'attitude est trouvée, le système produit une structure de traits de la forme suivante : $source = \{user, agent\}$, $polarity = \{neg, pos\}$, $targetType = \{user, agent, other, unknown\}$. A ces attributs sémantiques, est également ajouté un attribut syntaxique spécifiant la présence d'une négation dans l'expression d'attitude $negation = \{true, false\}$

`Att(aff) -> Src(usr|agt), SyntVb(att:true), Target.`	I like this book
`Att(int) -> Aux, Src(usr|agt), SyntVb(att:true), Target.`	Do you like this book?

Règle de polarité

$\texttt{Attitude(pol:Pol}_{\texttt{ChkVb}}\texttt{)}$

Dans ce type de phrase, la valeur attitudinale est portée par le syntagme verbal. Les règles de polarité sont donc appliquées au niveau syntagmatique de l'analyse (inversion de la valeur portée par le verbe en cas de négation). Au niveau phrastique, aucun modifieur n'entre en jeu, la polarité de l'expression est donc la même que celle du syntagme verbal.

FIGURE 3 – Règles du niveau phrastique prenant en compte des phrases de type verbal (processif)

4.2 Analyse des énoncés de l'utilisateur

4.2.1 Confirmation ou infirmation d'attitudes exprimées par l'agent

Lorsque l'agent exprime une attitude sous forme d'affirmation ou de question, l'utilisateur peut être amené à formuler : (i) une confirmation ou une infirmation simple, la première définissant le contenu propositionnel de l'agent comme vrai, la seconde comme faux ; (ii) une confirmation ou une infirmation modalisée, faisant porter sur le contenu propositionnel une modalité pouvant être aléthique, déontique, temporelle ou épistémique (Le Querler, 1996). Pour résumer de manière formelle le fonctionnement sémantique de ces deux types de segments, il est possible de dire qu'ils définissent une valeur pour un attribut $value_p$ prise dans l'ensemble $\{p, \neg p, \diamond p, \Box p, \diamond\neg p, \Box\neg p\}$. Pour le moment, le système ne détecte que les confirmations ou infirmations simples. Leurs versions modalisées seront intégrées dans une version ultérieure. Lorsque la réponse fait suite à un énoncé référant à une attitude, la valeur qu'elle attribue à $value_P$ va également entraîner une confirmation ou une infirmation de la valeur que l'énoncé de l'agent avait attribuée à $polarity$. En cas d'une infirmation, la valeur de l'attribut sera ainsi inversée. Il est également important de noter que, de même qu'une confirmation ne s'exprime pas nécessairement par un *yes* (ou ses synonymes), une infirmation ne l'est pas nécessairement par un *no* (ou ses synonymes). Ainsi, dans l'énoncé interro-négatif « Don't you love outdoors activities », le contenu propositionnel « you don't love outdoors activities » est affirmé par la réponse « no » et infirmé par « yes ». Afin de pouvoir déterminer la valeur des l'attribut $Value_p$ et par la suite celle de l'attribut $Polarity$, un typage de la réponse permettant de savoir si elle équivaut à un *yes* ou à un *no* est nécessaire. Ce typage est effectué par une simple vérification de la présence d'un «yes», d'un «no» ou d'un de leurs synonymes dans les cinq premiers mots de l'énoncé.

Une fois déterminé le type de la réponse (*yes*, *no*), le système lance le calcul de la valeur de l'attribut $Value_P$. Pour calculer cette valeur, les règles présentées ci-dessus s'appuient sur le type de la réponse et la valeur de l'attribut $negation$ définie au cours de l'analyse de l'énoncé de l'agent. Ainsi, lorsque la réponse équivaut à un *yes* : si $negation = false$ alors $value_P = p$, si $negation = true$ alors $value_P = \neg p$. En revanche, lorsque la réponse équivaut à un *no*, si $negation = false$ alors $value_P = \neg p$, si $negation = true$ alors $value_P = p$. A partir de la valeur attribuée à $value_P$, la valeur de $Polarity$ peut être définie. Lorsque p est nié, l'attribut est $polarity$ prend la valeur inverse de celle de la fiche de l'agent : $if value_P = \neg p, alors$: $polarity_{user} = reverse(polarity_{agent})$. La valeur sera en revanche conservée lorsque p est défini comme vrai par la réponse de l'utilisateur : $if Value_P = p, alors polarity_{user} = polarity_{agent}$.

4.2.2 Détection d'attitudes pleinement formulées

Le processus de détection des attitudes pleinement formulées et référant à des *likes* ou *dislikes* est ici le même que celui appliqué pour la détection d'attitudes exprimées par l'agent, à l'exception que les expressions d'attitudes recherchées ne prennent pas de forme interrogative. Là encore, le processus s'applique en trois phases : lexicale, syntagmatique et phrastique. L'ensemble des patrons utilisés pour ces dernières phases correspondent aux patrons notés *affirm* dans les figures 2 et 3. Concernant la source, dans la mesure où le système ne cherche à détecter, dans l'énoncé de l'utilisateur, que des attitudes dont ce dernier est la source, une expression d'attitude n'est considérée comme pertinente que lorsque sa source correspond à un pronom de première personne ($Src(usr) \rightarrow$ "I"|"me"). Au terme de l'analyse de l'énoncé de l'utilisateur, le système fournit en sortie pour chaque attitude pertinente détectée – exprimées soit par confirmation-infirmation soit par formulation pleine – une structure de traits $source = user$, $polarity = \{neg, pos\}$, $targetType = \{user, agent, other, unknown\}$.

5 Première évaluation du système

5.1 Campagne d'annotation sur Mechanical Turk

Protocole Afin d'évaluer notre système, nous avons mené une campagne d'annotation via la plate-forme Amazon Mechanical Turk. 60 sous-ensembles du corpus Semaine, chacun composé de 10 paires d'énoncés (un énoncé de l'agent et la réponse de l'utilisateur), ont été annotés par 240 annotateurs anglophones natifs (4 pour chaque sous-ensemble du corpus). Pour chaque paire présentée aux annotateurs, une série de questions leur sont posées visant à vérifier la présence d'une expression référant à un *like* ou *dislike* de l'utilisateur. Tandis que la première question interroge sur la présence d'une telle expression, la seconde permet d'en spécifier le nombre d'occurrences dans la paire sélectionnée. Si l'annotateur détecte plusieurs expressions, les questions suivantes sont posées pour chaque d'elles. La troisième question interroge ensuite la

nature de la cible : seules quatre catégories – celles détectables par le système – sont proposées. La dernière question concerne la polarité des expressions détectées par l'annotateur (positive ou négative).

Mesures d'accord inter-annotateurs Concernant la réponse à la première question, nous mesurons le taux d'accord entre annotateurs sur la présence d'au moins une expression de référant à un *like* ou un *dislike*, grâce au Kappa de Fleiss (Fleiss, 1971) (Tableau 1). Nous mesurons la cohérence entre annotateurs concernant le nombre d'expressions détectées dans une paire en calculant le coefficient alpha de Cronbach (Cronbach, 1951). Pour la polarité et le type de cible, nous sélectionnons les paires où au moins deux annotateurs sont d'accord sur la présence d'une expression de *like* ou de *dislike* et nous considérons uniquement les annotations fournies par ces annotateurs. Nous mesurons ensuite le pourcentage d'accord. En fait, suite à la sélection des données montrant un consensus, nous obtenons un sous-ensemble d'annotations avec un taux non-fixe d'annotateurs : une simple mesure de pourcentage d'accord nous a donc semblé appropriée. Concernant la polarité, 41% des sous-corpus ont un pourcentage d'accord compris entre 50% et 75%, et 52% des sous-corpus ont un pourcentage d'accord supérieur à 75%. Concernant la catégorisation de la cible, 61% des sous-corpus ont un pourcentage d'accord supérieur à 50%.

	Fleiss' Kappa	**Cronbach's alpha**
Max	0.79	0.90
Min	0.00	0.29
Median	0.32	0.72
Average	0.25	0.59

TABLE 1 – Scores du Kappa de Fleiss et coefficients alpha de Cronbach pour chaque sous-ensemble annoté

5.2 Évaluation du système

Des 600 paires annotées, nous avons supprimé les paires où aucun consensus – une majorité d'annotateurs appliquant une même annotation – a été trouvé. 503 paires ont ainsi été conservées. Nous considérons la référence et notre système comme des annotateurs distincts. En effet, en analyse de sentiments, une vérité terrain construite à partir d'indices objectifs est difficile à obtenir : les observations faites par les annotateurs restent des interprétations subjectives. Sur la base de ce postulat, nous avons choisi d'évaluer l'accord entre notre système et la référence. Pour mesurer l'accord entre notre système et cette référence ainsi constituée, nous avons appliqué les trois mesures utilisées précédemment : kappa de Fleiss, pour la présence d'une expression référant à un *like* ou un *dislike* et la polarité, coefficient alpha de Cronbach, pour le nombre d'expression et pourcentage d'accord, pour le type de cible. L'accord entre les sorties du système et la référence est substantiel concernant la présence d'au moins une expression ($k = 0.61$). Le nombre d'expressions est lui-aussi correctement détecté par le système, le coefficient équivalant 0.67 (il est souvent admis que la valeur d'accord minimale acceptable est de 0.60). Le kappa de Fleiss obtenu pour la polarité est lui-aussi encourageant puisqu'il est égal à 0.844. Concernant le type de cible, on obtient un pourcentage d'accord de 53%. Le désaccord est souvent lié à un confusion entre les catégories *unknown* et *other*.

6 Conclusion et perspectives

Dans cet article, nous avons proposé une approche des attitudes – affects, appréciations, jugements – dans le cadre des conversations humain-agent. Afin de pouvoir s'adapter aux enjeux et contraintes de ce contexte spécifique, nous avons fourni un modèle d'annotation dédié à l'étude de ces expressions telles qu'elles se réalisent dans le contexte d'une parole conversationnelle. Appliqué au corpus Semaine, le modèle révèle un lien entre les attitudes exprimées par l'utilisateur et les énoncés de l'agent qui le précédent, tant le plan pragmatique que sur celui du contenu propositionnel. Sur la base de cette étude, nous proposons une première version de notre système se concentrant sur la détection des attitudes pouvant exprimer un *like* ou *dislike*. L'évaluation de cette première version montre des résultats encourageants. Après une analyse des résultats, il semble que les causes de désaccord entre le système et la référence soient en partie liées au manque de contexte des paires d'énoncés annotées. En effet, pour cette première version, le système ne considère que des paires d'énoncés. Si cela a permis de détecter un grand nombre d'expressions, il est néanmoins probable que cela ait créé une confusion interprétative dans certains cas, tant du côté du système que de celui des annotateurs humains. Cette confusion

peut ainsi être cause de désaccords. Dans l'exemple suivant, Agent : « good. Ah good » (« bien. Ah bien ») – Utilisateur : « my favorite emotion » (« mon émotion favorite »), si la source peut être identifiée, la détection de la cible est impossible pour une analyse se concentrant sur cette seule paire d'énoncés.

Plusieurs éléments peuvent être pris en compte pour améliorer ce résultat mais aussi les performances du système. Tout d'abord, sur le plan de l'organisation conversationnelle, il est important que le système puisse considérer un plus large contexte d'analyse. Tout d'abord, pour chaque tour de parole utilisateur analysé, il est important de pouvoir avoir accès à l'ensemble des tours de parole – agent et utilisateur – énoncés en amont de la conversation. Les règles sémantiques utilisées par le système devront modéliser la manière dont l'utilisateur et l'agent collaborent sur plusieurs tours de parole successifs à l'expression d'attitude. Ensuite, afin de pouvoir correctement gérer le prise en compte de ce contexte plus large, il sera nécessaire que chaque analyse effectuée sur un tour de parole spécifique soit rendue accessible par les analyses des tours de parole ultérieurs. Cette approche permettra une meilleure détection des expressions d'attitude mais aussi une meilleure identification de leur cible – notamment dans le cas de problème de référence anaphorique.

Remerciements

L'auteur souhaite remercier l'équipe Greta pour sa contribution à la plateforme Greta-VIB. Ce travail a été développé dans le cadre du Labex SMART (ANR-11-LABX-65) supporté par l'ANR au sein du programme Investissements d'Avenir (ANR-11-IDEX-0004-02).

Références

BLOOM K., GARG N. & ARGAMON S. (2007). Extracting appraisal expressions. *HLT-NAACL*, p. 165–192.

BRECK E., CHOI Y. & CARDIE C. (2007). Identifying expressions of opinion in context. In S. S., M. H. & B. R. K., Eds., *International Joint Conference On Artifical Intelligence*, p. 2683–2688, San Francisco, CA : Morgan KoffMann Publishers.

CHOI Y., CARDIE C., RILOFF E. & PATWARDHAN S. (2005). Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, p. 355–362, Stroudsburg, PA, USA : Association for Computational Linguistics.

CLARK H. H. & SCHAEFER E. F. (1989). Contributing to discourse. *Cognitive Science*, **13**, 259–294.

CRONBACH L. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, **16**(3), 297–334.

ESULI A. & SEBASTIANI F. (2005). In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, p. 617–624, New York, NY, USA : ACM.

FLEISS J. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, **76**(5), 378–382.

HATZIVASSILOGLOU V. & MCKEOWN K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, p. 174–181, Stroudsburg, PA, USA : Association for Computational Linguistics.

HEIDER F. (1958). *The psychology of interpersonal relations*. Lawrence Erlbaum associates Inc.

HU M. & LIU B. (2004). Mining opinion features in customer reviews. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, p. 755–760 : AAAI Press.

ISHIZUKA M. (2012). Textual affect sensing and affect communication. In *IEEE Transaction 11th International Conference on Cognitive Informatics and Cognitive Computing*, p. 2–3.

IZARD C. E. (1977). *Human Emotions*. New York, USA : Plenum Press.

LANGLET C. & CLAVEL C. (2015). Improving social relationships in face-to-face human-agent interactions : when the agent wants to know user's likes and dislikes. In *The Association for Computer Linguistics*, Beijin, China. to appear.

LE QUERLER N. (1996). *Typologie des modalités*. Presse Universitaire de Caen.

MARTIN J. R. & WHITE P. R. (2005). *The Language of Evaluation. Appraisal in English*. London and New York : Macmillan Basingstoke.

MCKEOWN G., VALSTAR M., COWIE R., PANTIC M. & SCHRODER M. (2011). The semaine database : Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE Transactions on Affective Computing*, **3**(1), 5–17.

MOILANEN K. & PULMAN S. (2007). Sentiment composition. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007)*, p. 378–382.

NASUKAWA T. & YI J. (2003). Sentiment analysis : Capturing favorability using natural language processing. In *Proceedings of the 2Nd International Conference on Knowledge Capture*, K-CAP '03, p. 70–77, New York, NY, USA : ACM.

NEVIAROUSKAYA A., PRENDINGER H. & ISHIZUKA M. (2007). Textual affect sensing for sociable and expressive online communication. In *Proceedings of the 2Nd International Conference on Affective Computing and Intelligent Interaction*, ACII '07, p. 218–229, Berlin, Heidelberg : Springer-Verlag.

NEVIAROUSKAYA A., PRENDINGER H. & ISHIZUKA M. (2010a). Emoheart : Conveying emotions in second life based on affect sensing from text. *Adv. in Hum.-Comp. Int.*, **2010**.

NEVIAROUSKAYA A., PRENDINGER H. & ISHIZUKA M. (2010b). Recognition of affect, judgment, and appreciation in text. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, p. 806–814, Stroudsburg, PA, USA : Association for Computational Linguistics.

ORTONY A., CLORE G. & COLLINS A. (1990). *The Cognitive Structure of Emotions*. Cambridge, University Press.

OSGOOD C., MAI W. H. & MIRON M. S. (1975). *Cross-cultural Universals of Affective Meaning*. Urbana : University of Illinois Press.

PANG B. & LEE L. (2004). A sentimental education : Sentiment analysis using subjectivity. In *Proceedings of ACL*, p. 271–278.

PAUMIER S. (2015). *Unitex user manual*. Université de Paris-Est Marne-la-Vallée.

POGGI I., PELACHAUD C., DE ROSIS F., CAROFIGLIO V. & DE CAROLIS B. (2005). Greta. a believable embodied conversational agent. In *Multimodal intelligent information presentation*, p. 3–25. Springer.

QUIRK R. (1985). *A Comprehensive grammar of the English language*. General Grammar Series. Longman.

RILOFF E. (1996). An empirical study of automated dictionary construction for information extraction in three domains. *Artificial Intelligence*, **85**, 101–134.

RILOFF E. & WIEBE J. (2003). Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, p. 105–112, Stroudsburg, PA, USA : Association for Computational Linguistics.

SEARLE J. R. (1976). A classification of illocutionary acts. *Language in society*, **5**(01), 1–23.

SHAIKH M., PRENDINGER H. & ISHIZUKA M. (2009). A linguistic interpretation of the occ emotion model for affect sensing from text. In *Affective Information Processing*, p. 378–382 : Springer London.

SMITH C., CROOK N., DOBNIK S. & CHARLTON D. (2011). Interaction strategies for an affective conversational agent. In *Presence : Teleoperators and Virtual Environments*, volume 20, p. 395–411 : MIT Press.

TURNEY P. D. (2002). Thumbs up or thumbs down ? : Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, p. 417–424, Stroudsburg, PA, USA : Association for Computational Linguistics.

VALITUTTI R. (2004). Wordnet-affect : an affective extension of wordnet. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, p. 1083–1086.

WHITELAW C., GARG N. & ARGAMON S. (2005). Using appraisal taxonomies for sentiment analysis. *Proceedings of CIKM-05, the ACM SIGIR Conference on Information and Knowledge Management*.

WIDLÖCHER A. & MATHET Y. (2012). The glozz platform : A corpus annotation and mining tool. In *Proceedings of the 2012 ACM Symposium on Document Engineering*, p. 171–180, Paris, France.

WIEBE J. & RILOFF E. (2005). Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'05, p. 486–497, Berlin, Heidelberg : Springer-Verlag.

WILSON T., WIEBE J. & HOFFMANN P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, p. 347–354, Stroudsburg, PA, USA : Association for Computational Linguistics.

WILSON T., WIEBE J. & HWA R. (2004). Just how mad are you ? finding strong and weak opinion clauses. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, p. 761–767 : AAAI Press.

YANG B. & CARDIE C. (2013). Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 1640–1649, Sofia, Bulgaria : Association for Computational Linguistics.

YILDIRIM S., NARAYANAN S. & POTAMIANOS A. (2011). Detecting emotional state of a child in a conversational computer game. *Computer Speech & Language*, **25**(1), 29–44.

Résumé Automatique Multi-Document Dynamique : État de l'Art

Maâli Mnasri[1, 2]
(1) CEA,LIST, Laboratoire Vision et Ingénierie des Contenus, Gif-sur-Yvette, F-91191 France.
(2) Univ. Paris Sud, Orsay, France
maali.mnasri@cea.fr

Résumé. Les travaux menés dans le cadre du résumé automatique de texte ont montré des résultats à la fois très encourageants mais qui sont toujours à améliorer. La problématique du résumé automatique ne cesse d'évoluer avec les nouveaux champs d'application qui s'imposent, ce qui augmente les contraintes liées à cette tâche. Nous nous intéressons au résumé extractif multi-document dynamique. Pour cela, nous examinons les différentes approches existantes en mettant l'accent sur les travaux les plus récents. Nous montrons ensuite que la performance des systèmes de résumé multi-document et dynamique est encore modeste. Trois contraintes supplémentaires sont ajoutées : la redondance inter-document, la redondance à travers le temps et la grande taille des données à traiter. Nous essayons de déceler les insuffisances des systèmes existants afin de bien définir notre problématique et guider ainsi nos prochains travaux.

Abstract.
Automatic multi-document update summarization : State of the Art
The field of automatic text summarization is characterized both by some interesting achievements and a lot of issues to address, especially with the introduction of new tasks brought by applications. In this article, we focus more particularly on the multi-document update summarization task and review the existing work about it with a special emphasis on recent work. We show that the results for this task are still low because of the necessity to take into account three important constraints : information redundancy through documents and time and the size of data. We analyze the strengths and weaknesses of existing systems according to these constraints to propose subsequently new solutions.

Mots-clés : Résumé multi-document, résumé dynamique, redondance, évaluation.

Keywords: Multi-document summarization, update summarization, redundancy, evaluation.

1 Introduction

Le résumé multi-document dynamique (dit aussi évolutif ou mis à jour) est l'une des thématiques de recherche récentes à laquelle les chercheurs en Traitement Automatique des Langues (TAL) se sont intéressés, en particulier à l'occasion des campagnes d'évaluation TAC (Text Analysis Conference) et précisément à travers les tâches de résumé dynamique. Étant donné un flux de documents (par exemple des dépêches de presse), l'objectif est de générer un résumé se concentrant sur les nouveautés apportées par les documents les plus récents par rapport aux documents plus anciens. Cet article dresse un état de l'art des travaux sur le résumé automatique. Il présente une revue des développements récents dans ce domaine qui constitueront le point de départ dans nos travaux à venir. Plusieurs revues sur le résumé automatique ont déjà été publiées (Spärck Jones, 2007; Das & Martins, 2007; Gupta & Lehal, 2010; Nenkova & McKeown, 2012; Torres-Moreno, 2014). Nous essayons, à travers cet article, de les compléter. À cet effet, nous présentons l'évolution du domaine ainsi que les dernières réalisations remarquables. De plus, nous structurons notre analyse autour de deux axes principaux : le résumé multi-document et le résumé dynamique. Il existe d'autres axes intéressants, par exemple les résumés multilingues et crosslingues, mais qui ne font pas partie des objectifs de cet article.
Les recherches que nous considérons ici sont motivées par le besoin croissant d'applications variées de synthèse d'information en général et de résumé automatique de texte en particulier. Dans le cas des moteurs de recherche par exemple, un système de résumé permet de présenter un contenu bref et pertinent par rapport à une requête de l'utilisateur (Nenkova & McKeown, 2012). Le résumé d'articles scientifiques fait aussi partie des principaux cas d'utilisation (Jaidka *et al.*, 2013) compte tenu du nombre croissant de documents scientifiques disponibles sous une forme numérique dont la lecture et la compilation deviennent de plus en plus difficiles. Le domaine de la presse est aussi un champ d'application majeur. La consultation des dépêches en ligne à partir des terminaux mobiles est pénible si les articles sont longs et nombreux. Le résumé automatique résout ce problème en présentant les informations les plus importantes sous une forme réduite (Plaza

et al., 2010).
Dans la section suivante, nous analysons le problème du résumé automatique en caractérisant les différents types de résumés. Nous présentons dans la troisième section le principe et les réalisations traitant la problématique du résumé abstractif. L'explication détaillée des techniques utilisées dans le cadre du résumé extractif fera l'objet de la quatrième section. Nous nous focalisons dans la cinquième section sur le résumé multi-document dynamique, point central de notre attention dans cet article. La sixième section est consacrée aux méthodes d'évaluation actuelles des systèmes de résumé automatique. Nous clôturons cet article par un travail de synthèse présenté dans la dernière section.

2 Résumé automatique : Analyse du problème

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

Mode de production du résumé. Nous distinguons les méthodes *extractives* (Dalal & Malik, 2013) des méthodes *abstractives* (Genest & Lapalme, 2012). Le résumé extractif est formé de phrases extraites du texte source. 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). Jusqu'à aujourd'hui, ces méthodes sont les plus exploitées parce qu'elles évitent le problème de la génération de texte, toujours considéré comme une tâche complexe. Les méthodes abstractives sont inspirées des travaux en psycholinguistique cognitive et en intelligence artificielle et notamment du modèle théorique de la compréhension de van Dijk et Kintsch (Kintsch & 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.

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 lexical 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.

Généricité du résumé. Un résumé de texte est soit générique, soit orienté. Le résumé générique est produit en se référant uniquement au contenu du texte source et 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.

Style du résumé. 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 rend compte des sujets les plus importants évoqués par le texte. Certains systèmes de résumés guidés (Saggion & 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.

3 Résumé abstractif

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 & 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 & 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 & 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). Ces méthodes ne sont pas très largement exploitées. Ceci peut être dû à la rareté des outils de génération du texte. Le domaine de la génération est toujours en cours de développement mais pas encore à maturité, ce qui freine parfois l'implémentation des systèmes abstractifs. Les chercheurs préfèrent alors se tourner vers les méthodes extractives, qui ne dépendent pas de ce prérequis. La majorité des travaux s'étant intéressés aux méthodes extractives, ces dernières ont connu un développement plus important. La majorité des évaluations menées sur le résumé ont aussi été conçues pour des résumés plutôt extractifs. Ceci explique en partie les résultats moins encourageants des systèmes abstractifs. Néanmoins, certains chercheurs pensent que ceux-ci pourraient susciter un certain regain d'intérêt. Cette prédiction est justifiée d'une part, par le besoin de résumés plus proches des résumés manuels (Nenkova & McKeown, 2012) et d'autre part, par le plafonnement des performances des techniques extractives.

4 Résumé extractif

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 de 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.

4.1 Critères de sélection

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

4.1.1 Critères liés au contenu du texte

Cet ensemble de critères s'intéressent 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 les 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. La différence est qu'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 des textes est une notion très importante en TAL. Plusieurs mesures de similarité textuelle ont été établies (Bär *et al.*, 2015). Dans le domaine du résumé automatique, elle 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 & 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 sert aussi à répondre à des requêtes standards

(OÙ, QUI, QUAND, etc.) dans le résumé guidé (Ng *et al.*, 2011). Certains vont au delà de cette étape et déterminent les rôles sémantiques des entités reconnues (Trandabâţ, 2011). L'entité la plus fréquente est identifiée, c'est l'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 communique. 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.

4.1.2 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 arbitraires. 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 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 & 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* sont extraites. 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 sources.

Similarité avec le titre. Plus une phrase est similaire au titre, plus elle est liée au sujet principal du texte (Edmundson, 1969; Perret, 2005) 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 soupçonnées de détailler des informations déjà exprimées dans l'ensemble des documents par des phrases plus courtes et donc de 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 words*). 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. La Rhetorical Structure Theory (RST) (Mann & Thompson, 1988) est une méthode d'analyse de discours. Elle s'appuie sur une segmentation des textes en EDU, unités discursives élémentaires (phrases ou parties de phrase). Chaque EDU est soit noyau soit satellite selon son importance dans le texte. Un noyau apporte à lui seul une information jugée pertinente. Un satellite est en revanche complémentaire. La RST représente la cohérence et la structure du texte par un ensemble de relations rhétoriques entre les noyaux et les satellites : exemplification, preuve, justification, etc. Contrairement aux autres critères de sélection, ceux exploitant la RST tiennent ainsi compte de la structure du texte (Marcu, 1997, 1998). Les analyseurs RST (Joty *et al.*, 2013; Feng & Hirst, 2014) produisent un arbre binaire dont les feuilles représentent les EDU. Les autres nœuds représentent les relations rhétoriques entre les EDU et portent le numéro de leurs fils noyaux. De cette façon, les nœuds les plus proches de la racine sont les plus importants. L'ordre d'importance des EDU peut alors être déduit en parcourant l'arbre de la racine vers les feuilles. L'architecture d'arbre imposée par la RST n'est pas souvent représentative du texte, notamment lorsque deux EDU présentent plus qu'une relation rhétorique ou quand un nœud doit avoir deux parents. Pour résoudre ce problème, un nouveau modèle sous forme de graphe dont les arcs représentent les relations rhétoriques a été proposé (Wolf & Gibson, 2005). La comparaison de ces deux représentations (graphe et arbre) pour le résumé automatique a fait l'objet d'un certain nombre d'études (Louis *et al.*, 2010).

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.

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.

4.2.1 Méthodes par apprentissage

Du point de vue de l'apprentissage, le résumé automatique a été considéré 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. 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. 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). Les réseaux de neurones artificiels (RNA) ont aussi été utilisés pour l'apprentissage supervisé (Kaikhah, 2004). Chaque phrase du texte est modélisée par un vecteur de n composantes, chacune correspondant à un critère de sélection. Le RNA est composé de n neurones d'entrées (un neurone par critère), une couche cachée de p neurones et un neurone de sortie. Ce dernier indique si la phrase en entrée doit être incluse dans le résumé. L'apprentissage permet d'adapter le poids des liaisons entre les couches en fonction d'un ensemble d'exemples. À l'issue de cet apprentissage, les liaisons de très faible poids sont éliminées, de même que les neurones isolés.

4.2.2 Méthodes fondées sur la programmation linéaire en nombres entiers

McDonald fut le premier à modéliser le problème du résumé automatique multi-document par le biais de la Programmation Linéaire en Nombres Entiers (PLNE) (McDonald, 2007). Depuis, plusieurs chercheurs utilisent cette méthode pour combiner différents critères de sélection (Li *et al.*, 2011; Woodsend & Lapata, 2012; Li *et al.*, 2013). Le principe consiste à maximiser une fonction objectif favorisant les unités textuelles satisfaisant les critères de sélection et pénalisant la redondance entre les unités sélectionnées. Le calcul est effectué sous un ensemble de contraintes à fixer selon le système de résumé. La contrainte triviale est la taille souhaitée du résumé. Le modèle proposé par McDonald attribue des poids aux phrases du texte source. La PLNE permet de choisir les phrases ayant les poids maximaux et une redondance minimale avec les phrases déjà sélectionnées sans dépasser la taille maximale du résumé. D'autres modèles extraient du texte les concepts qu'il évoque et leur associent des poids. La PLNE permet de choisir les phrases couvrant les concepts ayant les poids les plus élevés sans dépasser la longueur du résumé (Gillick & Favre, 2009). Une contrainte d'exclusivité des concepts a été ajoutée pour éviter la redondance. Un concept n'est autorisé à apparaître qu'une seule fois dans le résumé. La difficulté réside dans le choix des concepts pouvant être les sujets les plus importants, des entités nommées ou des relations sémantiques. Cette étape a été simplifiée dans certains travaux en supposant que les concepts sont tous les bigrammes du texte source (Gillick & Favre, 2009). Ce choix a été justifié par la complexité et la fragilité des analyses sémantiques, qui peuvent donner des concepts erronés ou ambiguës et par conséquent un résumé hors-sujet.

4.2.3 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 liens par les arcs. Il attribue récursivement à chaque nœud un score dépendant de la structure de tout le graphe. 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 graphes 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 leur longueur. 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 en français (Boudin & Torres-Moreno, 2009). Pour éliminer la redondance, un seuil de similarité maximal a été fixé. Au-delà de ce seuil, deux sommets ne sont plus connectés.

5 Résumé multi-document dynamique

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.

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.

5.1.1 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 sur-représenter 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 proposées. La première famille d'approches commence 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. 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 combinatoire 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 & Ravindran, 2008; Li *et al.*, 2011). Ensuite, ils classifient les phrases par thème et choisissent une ou plusieurs phrases pour couvrir chaque thème.

5.1.2 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.*, 2011) 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.

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 & Kumar, 2008; McCreadie *et al.*, 2014).

5.2.1 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.

5.2.2 Redondance à travers le temps

La contrainte nouvelle 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 ordonnées, 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.

5.2.3 Travaux récents

Dans ce qui suit, nous décrivons les méthodes adoptées par les meilleurs systèmes de résumé dynamique dans TAC 2011 en soulignant parallèlement les approches proposées pour le résumé multi-document puisque la majorité des travaux traitent les deux problèmes simultanément. Certaines approches consistent à entraîner un modèle de régression SVR (*Support Vector Regression*) sur les données de TAC 2010 afin de prédire le score de chaque phrase des textes sources (Ng *et al.*, 2011). Pendant la phase d'apprentissage, les scores sont calculés en fonction de la similarité ROUGE-2 (cf. paragraphe 6.1) avec les résumés manuels. Le problème de redondance est traité lors du classement des phrases en utilisant l'algorithme *Maximal Marginal Relevance*. Celui-ci pénalise le score d'une phrase par sa similarité avec les phrases déjà retenues dans le résumé et par sa similarité avec la phrase la plus similaire de l'ensemble de documents A. D'autres chercheurs ont suivi une approche par évitement (Wan, 2012). Ils modélisent toutes les phrases des ensembles A et B par les sommets d'un seul graphe. De cette manière, la redondance est évitée *a priori*. Pour chaque sommet, sont calculés itérativement l'apport de nouveauté et la corrélation avec les sujets principaux du document en déterminant les similarités et dissimilarités entre les composantes de tout le graphe. À l'issue de la convergence de l'algorithme, les sommets ayant l'apport de nouveauté maximal sont sélectionnés pour le résumé final. Le résumé dynamique était proposé comme tâche à DUC 2007 et dans le cadre de TAC Summarization de 2008 à 2011. En 2013, TREC (*Text REtrieval Conference*) propose pour la première fois une tâche de résumé temporel (track Temporal Summarization). Ce dernier est un résumé dynamique avec un forte variabilité des documents à résumer en fonction du temps. Les données d'entrée sont des flux de documents horodatés. L'évaluation TREC 2013 a ainsi fourni, pour la tâche de résumé temporel, un corpus de documents allant d'octobre 2011 jusqu'à janvier 2013. L'objectif était de fournir à l'utilisateur des résumés liés à un événement ou à un sujet donné et mis à jour au fil du temps.

6 Évaluation des résumés automatiques

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 standards, 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. Trois métriques sont généralement utilisées 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 communes entre le résumé candidat et les 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 sensé 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.

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.

6.1 ROUGE

ROUGE évalue les résumés en les comparant à des résumés modèles. Cette comparaison est automatique et ne nécessite pas un pré-traitement particulier. Elle est déduite à partir du recouvrement entre les N-grammes des deux textes. Cette méthode a montré une forte corrélation avec les jugements humains (Lin, 2004). La corrélation de Pearson des scores ROUGE-2 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.

6.2 PYRAMID

Cette méthode permet de comparer un résumé candidat à un ensemble de résumés de référence (Nenkova & 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'utilisation 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.

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. 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. Une des solutions proposées 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 sources et leurs résumés.

Ensuite, une méthode d'évaluation automatique fondée sur des métriques de similarité entre les documents sources et les résumés générés a été proposée (Louis & Nenkova, 2008). Elle permet d'atteindre une corrélation de 0,88 avec le score PYRAMID et 0,74 avec le score Responsiveness (Louis & Nenkova, 2013). D'autres travaux ont étendu cette approche dans la perspective du multilinguisme (Saggion *et al.*, 2010). 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.

7 Synthèse

Depuis les années 2000, les travaux sur le résumé automatique ont connu un fort développement, caractérisé par une grande quantité et une grande diversité de méthodes dont nous n'avons rendu compte que partiellement dans cet article. Les premières approches du résumé multi-document exploitaient des méthodes statistiques, des classifieurs bayésiens ou des techniques de regroupement. Les réseaux de neurones artificiels ont été aussi utilisés pour pondérer les critères de sélection. Les résultats préliminaires obtenus ont motivé les chercheurs à faire évoluer ces systèmes. L'analyse des documents textuels a été approfondie et les recherches ont été poussées vers la compréhension du texte par des analyses sémantiques et thématiques. Pour ce faire, des ressources externes comme Wikipédia et WordNet ont été exploitées et d'autres ont été crées (Ferret, 2004). Toutefois, les performances atteintes restent à améliorer. Les approches sémantiques peuvent ainsi être étendues pour intégrer davantage de critères dans le calcul des similarités entre les mots. C'est le cas dans le système CHORAL (García Flores *et al.*, 2009) sur lequel nous nous appuyons au niveau de son exploitation des sens de mots. L'établissement d'un modèle abstrait soulignant les concepts mis en valeur par les documents a toujours été l'objectif de plusieurs travaux de recherche. Néanmoins, cet objectif n'a pas été réalisé complètement compte tenu de la complexité de l'analyse sémantique et de la rareté des ressources. Il a été souvent remplacé par un modèle des N-grammes les plus fréquents et de mots indices (Schlesinger *et al.*, 2008; Gillick & Favre, 2009). Une autre piste pourrait être constituée par la construction de représentations lexicales distribuées par des réseaux de neurones dans le cadre du *Deep Learning*, approche qui a montré des résultats intéressants en TAL (Yu & Deng, 2011; Luong *et al.*, 2013; Zheng *et al.*, 2013). Très peu de travaux ont exploité ce type de représentations pour le résumé automatique (Denil *et al.*, 2014) jusqu'à présent. En ce qui concerne le résumé dynamique, qui est plus récent, la performance des systèmes actuels est faible. Les travaux existants peuvent être divisés en deux familles. Certains réutilisent des systèmes de résumé multi-document pour le résumé dynamique et se contentent d'éliminer la redondance entre les anciens et les nouveaux documents après avoir classé les segments textuels. Cette solution n'apporte pas de résultats très satisfaisants car elle dépend de la précision de la mesure de similarité utilisée. D'autres travaux considèrent la problématique du résumé dynamique plutôt comme un problème d'extraction d'information. Ces systèmes ciblent la détection de nouveauté dans le document mais n'accordent pas assez d'importance aux analyses sémantique, discursive et thématique des documents. Par ailleurs, le contexte d'usage influence considérablement la formulation du problème du résumé dynamique. Alors que le problème générique est l'élimination de la redondance entre les anciens et les nouveaux documents, certaines situations exigent cette redondance. La variation des données à travers le temps peut changer la pertinence d'une information. De plus, il peut s'avérer nécessaire de reprendre certaines informations afin de rappeler ce qui est important et situer le nouveau résumé dans son cadre. L'élimination de la redondance dans les résumés dynamiques doit certainement prendre en compte ces aspects.
Nos prochains travaux seront inspirés des études présentées dans cet article et fondés sur les leçons qui en sont tirées. Nous essaierons aussi d'exploiter de nouvelles pistes dans l'intention de rapprocher les performances des systèmes de résumé multi-document et dynamique vers les attentes de l'utilisateur final.

Références

ANGHELUTA R., BUSSER R. D. & MOENS M.-F. (2002). The use of topic segmentation for automatic summarization. In *Proceedings of the ACL-2002 Workshop on Automatic Summarization*, p. 11–12, New Brunswick, NJ.

ARORA R. & RAVINDRAN B. (2008). Latent dirichlet allocation based multi-document summarization. In *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data*, AND'08, p. 91–97, New York, NY, USA.

BÄR D., ZESCH T. & GUREVYCH I. (2015). *Composing Measures for Computing Text Similarity*. Rapport interne TUD-CS-2015-0017, TU Darmstadt, Allemagne.

BOUDIN F. & TORRES-MORENO J.-M. (2009). Résumé automatique multi-document et indépendance de la langue : une première évaluation en français. In *Actes de la 16ème conférence sur le Traitement Automatique des Langues Naturelles*, Senlis, France.

CHEUNG J. C. K., POON H. & VANDERWENDE L. (2013). Probabilistic Frame Induction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 837–846, Atlanta, Georgia.

CHOWDARY C. R. & KUMAR P. S. (2008). An incremental summary generation system. In *Proceedings of the 14th International Conference on Management of Data, December 17-19, 2008, IIT Bombay, Mumbai, India*, p. 83–92.

CONROY J. M., SCHLESINGER J. D. & KUBINA J. (2011). CLASSY 2011 at TAC : Guided and Multi-lingual Summaries and Evaluation Metrics. In *Proccedings of the Fourth Text Analysis Conference (TAC 2011)*.

DALAL V. & MALIK L. (2013). A Survey of Extractive and Abstractive Text Summarization Techniques. In *6th International Conference on Emerging Trends in Engineering and Technology (ICETET)*, p. 109–110.

DAS D. & MARTINS A. F. T. (2007). *A Survey on Automatic Text Summarization*. Rapport interne, Literature Survey for the Language and Statistics II course at Carnegie Mellon University.

DE CHALENDAR G. (2014). 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*, p. 2932–2937.

DENIL M., DEMIRAJ A., KALCHBRENNER N., BLUNSOM P. & DE FREITAS N. (2014). *Modelling, Visualising and Summarising Documents with a Single Convolutional Neural Network*. Rapport interne arXiv :1406.3830, University of Oxford.

EDMUNDSON H. P. (1969). New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery*, **16**(2), 264–285.

FENG V. W. & HIRST G. (2014). 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*, p. 511–521.

FERRET O. (2004). Discovering word senses from a network of lexical cooccurrences. In *Proceedings of Coling 2004*, p. 1326–1332, Geneva, Switzerland.

FERRET O., CHÂAR S. L. & FLUHR C. (2004). Filtrage pour la construction de résumés multi-documents guidée par un profil. *Traitement Automatique des Langues*, **45**(1), 65–93.

FILIPPOVA K. (2010). *Dependency Graph-Based Sentence Fusion and Compression*. PhD thesis, TU Darmstadt, Allemagne.

GARCÍA FLORES J., FERRET O. & DE CHALENDAR G. (2009). 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.

GENEST P.-E. & LAPALME G. (2011). Framework for Abstractive Summarization using Text-to-Text Generation. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, p. 64–73, Portland, Oregon.

GENEST P.-E. & LAPALME G. (2012). Fully Abstractive Approach to Guided Summarization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, p. 354–358.

GILLICK D. & FAVRE B. (2009). A Scalable Global Model for Summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, p. 10–18, Boulder, Colorado.

GUPTA V. & LEHAL G. S. (2010). A Survey of Text summarization Extractive Techniques. *Emerging Technologies in Web Intelligence*, **2**(3), 258–268.

HASSEL M. (2003). 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.

HEARST M. A. (1997). TextTiling : Segmenting Text into Multi-paragraph Subtopic Passages. *Computational Linguistics*, **23**(1), 33–64.

JAIDKA K., KHOO C. & NA J.-C. (2013). Deconstructing Human Literature Reviews – A Framework for Multi-Document Summarization. In *Proceedings of the 14th European Workshop on Natural Language Generation*, p. 125–135.

JOTY S., CARENINI G., NG R. & MEHDAD Y. (2013). Combining Intra- and Multi-sentential Rhetorical Parsing for Document-level Discourse Analysis. In *Proceedings of ACL*.

KAIKHAH K. (2004). Automatic Text Summarization with Neural Networks. In *Proceedings of IEEE International Conference on Intelligent Systems*, volume 1, p. 40–44.

KHAN A. & SALIM N. (2014). A Review on Abstractive Summarization Methods. *Journal of Theoretical and Applied Information Tech nology*, **59**(1), 64–72.

KINTSCH W. & VAN DIJK T. A. (1978). Toward a model of text comprehension and production. *Psychological Review*, **85**(5), 363 – 394.

LI C., QIAN X. & LIU Y. (2013). Using supervised bigram-based ILP for extractive summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, p. 1004–1013, Sofia, Bulgaria.

LI P., WANG Y., GAO W. & JIANG J. (2011). Generating aspect-oriented multi-document summarization with event-aspect model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, p. 1137–1146, Edinburgh, Scotland, UK.

LIN C.-Y. (2004). Rouge : A package for automatic evaluation of summaries. In *Text Summarization Branches Out : Proceedings of the ACL-04 Workshop*, p. 74–81, Barcelona, Spain.

LIN C.-Y. & HOVY E. (1997). Identifying Topics by Position. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, p. 283–290, Washington, DC, USA.

LOUIS A., JOSHI A. & NENKOVA A. (2010). Discourse indicators for content selection in summarization. In *Proceedings of the SIGDIAL 2010 Conference*, p. 147–156, Tokyo, Japan.

LOUIS A. & NENKOVA A. (2008). Automatic summary evaluation without human models. In *Proceedings of the First Text Analysis Conference (TAC 2008)*.

LOUIS A. & NENKOVA A. (2013). Automatically assessing machine summary content without a gold standard. *Computational. Linguistics*, **39**(2), 267–300.

LUHN H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research Development*, **2**(2), 159–165.

LUONG M.-T., SOCHER R. & MANNING C. D. (2013). Better Word Representations with Recursive Neural Networks for Morphology. In *CoNLL*, p. 104–113, Sofia, Bulgaria.

MADNANI N. & DORR B. J. (2010). Generating phrasal and sentential paraphrases : A survey of data-driven methods. *Computational Linguistics*, **36**(3), 341–387.

MANI I. (2001). *Automatic Summarization*. John Benjamins Publishing.

MANN W. C. & THOMPSON S. A. (1988). Rhetorical Structure Theory : Toward a functional theory of text organization. *Text*, **8**(3), 243–281.

MARCU D. (1997). *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. Rapport interne CSRG-371, Computer Systems Research Group, University of Toronto.

MARCU D. (1998). Improving summarization through rhetorical parsing tuning. In *Proceedings of The Sixth Workshop on Very Large Corpora*, p. 206–215, Montreal, Canada.

MCCREADIE R., MACDONALD C. & OUNIS I. (2014). 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, p. 301–310, New York, NY, USA : ACM.

MCDONALD R. (2007). A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, p. 557–564, Berlin, Heidelberg : Springer-Verlag.

MCKEOWN K. R., KLAVANS J. L., HATZIVASSILOGLOU V., BARZILAY R. & ESKIN E. (1999). 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, p. 453–460, Menlo Park, CA, USA.

MIHALCEA R. (2004). Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*.

NENKOVA A. & MCKEOWN K. (2012). *Mining Text Data*, chapter A Survey of Text Summarization Techniques. Springer.

NENKOVA A. & PASSONNEAU R. J. (2004). Evaluating content selection in summarization : The pyramid method. In *HLT-NAACL*, p. 145–152.

NETO J. L., FREITAS A. A. & KAESTNER C. A. A. (2003). Automatic text summarization using a machine learning approach. In *Advances in Artificial Intelligence*, p. 205–215 : Springer Berlin Heidelberg.

NETO J. L., SANTOS A., KAESTNER C. & FREITAS A. (2000). Generating text summaries through the relative importance of topics. In M. MONARD & J. SICHMAN, Eds., *Advances in Artificial Intelligence*, volume 1952 of *Lecture Notes in Computer Science*, p. 300–309. Springer Berlin Heidelberg.

NG J. P., BYSANI P., LIN Z., KAN M.-Y. & TAN C.-L. (2011). SWING : Exploiting Category-Specific Information for Guided Summarization. In *Proceedings of the Text Analysis Conference (TAC 2011)*, Gaithersburg, Maryland, USA.

PAGE L., BRIN S., MOTWANI R. & WINOGRAD T. (1999). *The PageRank Citation Ranking : Bringing Order to the Web*. Technical Report 1999-66, Stanford InfoLab.

PASSONNEAU R. J., CHEN E., GUO W. & PERIN D. (2013). Automated pyramid scoring of summaries using distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Short Papers)*, p. 143–147, Sofia, Bulgaria.

PERRET L. (2005). *Extraction automatique d'information : génération de résumé et question-réponse*. PhD thesis, Université de Neuchâtel Faculté des Sciences.

PLAZA L., DÍAZ A. & GERVÁS P. (2010). Automatic Summarization of News Using WordNet Concept Graphs. *IADIS International Journal on Computer Science and Information Systems*, **5**(1), 45–57.

RADEV D. R., JING H., STYŚ M. & TAM D. (2004). Centroid-based summarization of multiple documents. *Information Processing and Management*, **40**(6), 919–938.

RADEV D. R., TEUFEL S., SAGGION H., LAM W., BLITZER J., QI H., ÇELEBI A., LIU D. & DRABEK E. (2003). Evaluation challenges in large-scale document summarization. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, p. 375–382, Sapporo, Japan.

SAGGION H. & LAPALME G. (2002). Generating Indicative-Informative Summaries with SumUM. *Computational Linguistics*, **28**(4), 497–526.

SAGGION H., TORRES-MORENO J.-M., DA CUNHA I., SANJUAN E. & VELAZQUEZ MORALES P. (2010). Multilingual summarization evaluation without human models. In *23rd COLING International Conference on Computational Linguistics (Posters)*, p. 1059–1067.

SALTON G., SINGHAL A., BUCKLEY C. & MITRA M. (1996). Automatic text decomposition using text segments and text themes. In *Proceedings of the the Seventh ACM Conference on Hypertext*, HYPERTEXT '96, p. 53–65, New York, NY, USA : ACM.

SAMMUT C. & WEBB G. I. (2010). *Encyclopedia of Machine Learning*. Springer US.

SCHIFFMAN B., NENKOVA A. & MCKEOWN K. (2002). Experiments in multidocument summarization. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, p. 52–58, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

SCHLESINGER J. D., O'LEARY D. P. & CONROY J. M. (2008). Arabic/English Multi-document Summarization with CLASSY : The Past and The Future. In *Proceedings of the 9th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'08, p. 568–581, Berlin, Heidelberg : Springer-Verlag.

SPÄRCK JONES K. (2007). Automatic summarising : The state of the art. *Inf. Process. Manage.*, **43**(6), 1449–1481.

TORRES-MORENO J.-M. (2014). *Automatic Text Summarization*. Wiley-ISTE.

TRANDABĂŢ D. (2011). Using semantic roles to improve summaries. In *Proceedings of the 13th European Workshop on Natural Language Generation*, p. 164–169, Nancy, France.

WAN X. (2012). Update summarization based on co-ranking with constraints. In *Proceedings of COLING 2012 (Posters)*, p. 1291–1300, Mumbai, India.

WOLF F. & GIBSON E. (2005). Representing Discourse Coherence : A Corpus-Based Study. *Computational Linguistics*, **31**(2), 249–287.

WOODSEND K. & LAPATA M. (2012). 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*, p. 233–243, Jeju Island, Korea.

XU T., MCNAMEE P. & OARD D. W. (2013). HLTCOE Submission at TREC 2013 : Temporal Summarization. In *The Twenty-Second Text REtrieval Conference Proceedings*.

YANG Z., YAO F., SUN H., ZHAO Y., LAI Y. & FAN K. (2013). BJUT at TREC 2013 Temporal Summarization Track. In *The Twenty-Second Text REtrieval Conference Proceedings*.

YU D. & DENG L. (2011). Deep learning and its applications to signal and information processing. *Signal Processing Magazine, IEEE*, **28**(1), 145–154.

ZHENG X., CHEN H. & XU T. (2013). Deep Learning for Chinese Word Segmentation and POS Tagging. In *Proceedings of the Conference on EMNLP*, p. 647–657, Seattle, USA.

Alignement multimodal de ressources éducatives et scientifiques

Hugo Mougard
Université de Nantes
hugo.mougard@univ-nantes.fr

Résumé. Cet article présente certaines questions de recherche liées au projet COCo [1]. L'ambition de ce projet est de valoriser les ressources éducatives et académiques en exploitant au mieux les différents médias disponibles (vidéos de cours ou de présentations d'articles, manuels éducatifs, articles scientifiques, présentations, etc). Dans un premier temps, nous décrirons le problème d'utilisation jointe de ressources multimédias éducatives ou scientifiques pour ensuite introduire l'état de l'art dans les domaines concernés. Cela nous permettra de présenter quelques questions de recherche sur lesquelles porteront des études ultérieures. Enfin nous finirons en introduisant trois prototypes développés pour analyser ces questions.

Abstract.

Multimodal Alignment of Educative and Scientific Resources
This article presents some questions linked to the COCo [1] project. The ambition of this project is to better exploit educative and academic resources by leveraging the different available supports (courses and conference talks videos, scientific articles, textbooks, slides, etc). We will first describe the problem of joint educative or scientific multimedia resources use. We will then give an overview of the related state of the art that will allow us to introduce a few research questions that will be the subject of further studies. Finally we will introduce three research prototypes that are helpful to start to investigate those research questions.

Mots-clés : alignement, e-research, e-learning, multimodalité.

Keywords: alignment, e-research, e-learning, multimodality.

1 Introduction

À chaque avancée technologique, les professionnels de l'éducation doivent repenser et adapter leurs méthodes et matériels pour tirer partie des nouvelles possibilités. La récente arrivée des formations en ligne ouvertes à tous (*massive open online courses* ou *MOOC*s en anglais) a ainsi entraîné de nombreux changements d'usage pour les institutions éducatives et les étudiants. Un des points essentiels ayant permis l'essor des *MOOC*s est la disponibilité de plus en plus fréquente d'excellentes connexions internet, rendant les cours vidéos téléchargeables rapidement. Malgré ce rôle central des vidéos dans les nouveaux cours disponibles sur internet, celles-ci restent souvent traitées comme des blocs indépendants et ne sont pas coordonnées au mieux avec le reste du matériel éducatif. Ce constat constitue le point de départ des études détaillées par la suite, qui ont pour but de contribuer au changement de cet état de fait.

De plus, la disponibilité croissante de matériel vidéo ne se limite pas aux plates-formes de *MOOC*s mais concerne aussi la recherche. Ce phénomène s'explique par plusieurs facteurs : (1) le travail fondateur de VIDEOLECTURES [2], qui permet d'accéder à de nombreuses conférences et tutoriels scientifiques filmés, (2) les coûts de production réduits grâce à des initiatives comme MATTERHORN [3], (3) l'intérêt grandissant des institutions pour promouvoir leurs événements, et (4) le mouvement grandissant en faveur des données libres (open data). Ce phénomène d'augmentation de la quantité de matériel de recherche multimédia disponible se constate en particulier pour les vidéos correspondant aux présentations faites lors de conférences. La disponibilité de ces vidéos représente une opportunité sans précédent car elles sont souvent accompagnées de l'article qui leur est associé dans les actes de la conférence, ce qui constitue un cadre idéal pour expérimenter dans le domaine de l'intégration profonde de différents médias (hypermédia). En effet, traditionnellement,

1. http://www.comin-ocw.org/
2. http://videolectures.net
3. http://opencast.org/matterhorn/

les résultats scientifiques sont étudiés au travers de l'article dans lequel ils sont exposés, et peu de médias alternatifs sont disponibles pour le compléter. Pour guider la recherche qui suit, nous considérons que cette étude de l'article seule n'est pas optimale et qu'il faut inventer de nouveaux moyens de combiner les différents supports disponibles pour fournir une expérience riche et efficace aux utilisateurs de systèmes éducatifs ou de recherche.

2 Description du problème

Dans ce travail au sein du projet COCo, nous visons l'exploitation pleine des ressources multimédias dans un cadre éducatif ou scientifique. Pour atteindre ce potentiel nous nous appuyons sur le travail des communautés du traitement de la langue mais aussi du web sémantique, de l'hypertexte, de l'hypermédia et du multimédia.

Le système cible que nous souhaitons obtenir est composé de deux sous-systèmes qui interagissent :

1. Le premier système prend en entrée un ensemble de documents multimédia, par exemple un article scientifique au format PDF et un enregistrement de sa présentation donnée durant une conférence, et renvoie des liens typés allant d'un document à l'autre, avec des types allant de la thématique à la rhétorique (*e.g.* une définition dans un article pourra être liée au segment vidéo expliquant le concept défini à l'aide d'un lien de type `explication`. Dans le sens inverse, le segment vidéo pourra être lié à la définition par un lien de type `définition formelle`).
2. Le second système utilise ces liens pour proposer à l'utilisateur une navigation intuitive et précise dans le matériel pédagogique. Il doit aussi pouvoir retransmettre un retour utilisateur (implicite ou explicite) afin que le premier système puisse s'améliorer au fur à mesure de son utilisation.

Le but premier du système est de transformer les vidéos en hypervidéos, selon le modèle de Sadallah *et al.* (2012) et les articles en hyperarticles selon un modèle similaire. Une vidéo peut n'être utilisée que comme un bloc atomique, séquentiel, pour éclaircir un concept (Navarrete & Blat, 2002) ou au contraire être intégrée en profondeur dans d'autres ressources pour permettre une navigation aisée et non linéaire du matériel, elle devient alors une hypervidéo (Sadallah *et al.*, 2012).

Les types d'hyperliens (liens inter-documents) à utiliser et les possibilités de navigations offertes doivent tenir compte des caractéristiques complémentaires des principaux média disponibles. Nous les détaillons ci-dessous avec l'exemple de l'article scientifique et de sa présentation vidéo :

Capacité à être lu en diagonale Une présentation enregistrée est dure à parcourir et en particulier à survoler. Cette difficulté peut être contournée par un alignement avec son article grâce à la structure de ce dernier.

Définitions formelles Quand un utilisateur souhaite "se pencher sur les maths", une vidéo de présentation n'est habituellement pas suffisante. L'article peut être utilisé pour compléter efficacement certaines parties de la présentation avec des définition précises.

Références Les références présentes dans un article peuvent soit être utilisées directement soit constituer des pistes intéressantes pour des hyperliens. Il est par exemple possible d'inclure un morceau d'une présentation vidéo d'un article référencé pour éclaircir un point compliqué dans la présentation qui intéresse l'utilisateur, ou plus précisément une partie précise de celle-ci qui répond au besoin d'information de l'utilisateur.

Détails Le manque de détails dans une présentation vidéo est double : (1) Le manque de détails dans le contenu d'une présentation induit par les contraintes de temps imposées durant les conférences est solvable d'une manière similaire au manque de définitions formelles par un alignement avec l'article correspondant, et (2) Les parties de l'article correspondant à la présentation vidéo qui n'ont pas été utilisées peuvent aussi être intéressantes pour l'utilisateur. Un alignement permet de les récupérer en considérant les parties non alignées entre les deux média.

Figures La nature dynamique d'une présentation vidéo lui permet souvent de mieux expliquer les figures complexes qu'un article. Intégrer un clip de l'auteur qui explique certaines parties des concepts cruciaux de ses figures peut faire gagner un temps précieux à l'utilisateur.

Biais de l'auteur Les vidéos de présentations sont habituellement plus subjectives que les articles correspondants. Il est en conséquence plus facile d'y discerner le biais de l'auteur par rapport à son travail.

Vision globale Les présentations ont pour but principal de faire passer l'idée forte d'une recherche scientifique dans le contexte qui la met le mieux en valeur. Les articles correspondant aux présentations ont pour objectif de détailler davantage la recherche scientifique exposée, rendant la vision globale du contexte plus délicate à acquérir. Encore une fois combiner les deux média est donc profitable.

Synthèse D'un côté, le texte de l'article est plus aisé à résumer que la transcription d'une vidéo, de l'autre, le matériel est habituellement plus synthétique dans une présentation que dans un article scientifique. Conséquemment, les deux médias peuvent contribuer à améliorer la capacité de synthèse du système.

Une fois les types d'hyperliens formalisés pour satisfaire aux différents besoins de navigation soulignés par ces complémentarités, il est nécessaire de concevoir une interface de navigation pour les exploiter.

Nous prévoyons deux versions de ce système, l'une orientée e-learning sur la plate-forme COCo, l'autre orientée e-research sur la plate-forme VIDEOLECTURES :

E-learning sur COCo Le système s'intègrera ici dans une plate-forme de diffusion de ressources éducatives. Son but sera de dépasser le paradigme de la vidéo reine qui est dominant dans les plates-formes actuelles. Il devra ainsi permettre une navigation efficace entre le cours principal, les manuels et les lectures complémentaires.

E-research sur VIDEOLECTURES Sur VIDEOLECTURES, l'objectif est principalement d'utiliser les nombreux couples d'article et de leur présentation vidéo disponibles afin d'améliorer l'expérience d'étude de résultat scientifique.

3 État de l'art

Le système présenté ci-dessus nécessite des algorithmes et idées de plusieurs domaines de l'informatique. Dans cet article, nous nous concentrons sur trois d'entre eux :

1. l'alignement multimodal ;
2. les typologies spécialisées pour le traitement informatique des ressources scientifiques et éducatives ;
3. l'évaluation extrinsèque et l'apprentissage par renforcement, qui sont nécessaires pour pallier le manque de données et qui sont envisagés de manière jointe.

Ces trois domaines ne couvrent pas le problème de manière exhaustive mais fournissent déjà un point de départ satisfaisant pour notre travail.

3.1 Alignement multimodal

La création des liens entre les différents médias est au cœur d'un système hypermédia. Nous voyons ces liens comme la concrétisation d'un alignement multimodal. Brunning (2010) distingue trois granularités d'alignement : (1) niveau du document : identifier des paires de documents semblables au sein de corpus comparables ou parallèles, (2) niveau de la phrase : identifier les phrases similaires dans deux documents comparables ou parallèles, et (3) niveau du mot : trouver des mots similaires dans deux phrases comparables ou similaires. De par la nature de notre corpus (paires d'articles et leur présentation), l'alignement au niveau du document n'est pas nécessaire. Par ailleurs, l'alignement des paires au niveau du mot peut servir pour certaines tâches (lien d'entités) mais n'est pas nécessaire à la majorité des cas d'utilisations mentionnés en partie 2. Cet état de l'art se concentrera donc sur l'alignement au niveau de la phrase.

En outre, l'alignement multimodal est un sujet peu traité de manière directe. Nous le verrons donc comme la combinaison de trois facteurs :

1. des prétraitements et adaptions spécifiques aux extractions textuelles des modalités traitées ;
2. une mesure de similarité entre les différentes unités choisies (*e.g.* phrases pour un article scientifique, fenêtres fixes, groupes de souffle ou phrases reconstituées pour une vidéo) ;
3. un alignement entre les différents modes utilisant cette mesure.

Prétraitements et adaptions spécifiques aux extractions textuelles Le caractère multimodal des documents traités requiert une attention particulière. Dans cet article, Nous exploitons les matériels multimodaux par l'intermédiaire de l'extraction de leur modalité textuelle et le résultat de cette extraction est bruité (la modalité textuelle correspond au texte extrait du pdf, pour les documents écrits, et à la transcription automatique de la parole, pour les documents vidéos). Pour comprendre l'impact de ce bruit sur une chaîne de traitements classique, il suffit de remarquer qu'une des premières étapes nécessaires au traitement de la langue — la séparation du texte d'entrée en phrases — n'est plus possible aux

précisions habituelles sur du texte non bruité (98%) ; les erreurs de cette étape de pré-traîtement ont des répercussions multiplicatives sur les erreurs de la chaîne complète. En effet, la ponctuation ne peut pas être inférée à partir du seul son de la vidéo et n'est donc renvoyée par aucun système de transcription (hormis au prix d'étapes de post-traitement de la sortie du système). De même, alors que l'on peut détecter automatiquement des segments porteurs de sens dans un texte (paragraphes, sections), cette détection est beaucoup plus compliquée dans une extraction. À ces problèmes de structure manquante s'ajoutent les 20% de taux d'erreur sur les mots transcrits malgré l'utilisation du système état de l'art de TRANSLECTURES.

Pour ces raisons, depuis 2001, des campagnes d'évaluation sont organisées pour améliorer le traitement des supports multimédias (Smeaton *et al.*, 2001). Plus récemment, une initiative spécialisée a vu le jour sous le nom de MEDIAEVAL. Une de ses tâches, SEARCH & ANCHORING IN VIDEO ARCHIVES [4], évalue des systèmes sur un cas d'utilisation très courant : un utilisateur recherche par une requête textuelle une information dans un corpus vidéo et suit les liens proposés par le système depuis les résultats fournis pour satisfaire son besoin d'information. Ce cas d'utilisation est très proche de ceux rencontrés dans une application hypermédia, où un utilisateur suit des hyperliens pour satisfaire son besoin d'information. Cette tâche est donc d'un grand intérêt pour sa similarité aux besoins de notre étude. Des articles issus de ces campagnes d'évaluation, on peut relever des approches d'apprentissage automatique pour segmenter les transcriptions (Galuščáková, 2013) afin de maximiser les chances de renvoyer un segment vidéo qui aura du sens pour un utilisateur ou encore des travaux d'adapation des mesures de recherche d'information classiques aux supports multimédias (Eskevich *et al.*, 2012).

Mesures de similarité Parmi les différentes méthodes que l'on trouve dans la littérature pour comparer deux chaînes de caractères, deux grandes classes émergent : (1) les méthodes basées sur le calcul d'un coût de l'alignement optimal des deux chaînes, et (2) les méthodes basées sur un apprentissage supervisé.

L'étude des chaînes de caractères (*Stringology*) remonte aux années 1960 et est le domaine dans lequel ont été publiées la majorités des approches par alignement. La première distance qui utilise ce concept est due à Hamming (1950). Elle correspond au le nombre d'erreurs dans l'alignement direct, caractère par caractère, de deux séquences de caractères. Levenshtein (1966) a introduit par la suite une distance prenant en compte le nombre pondéré de suppressions, insertions et substitutions nécessaires pour transformer la chaîne source en la chaîne cible. L'adaptation de cette distance à des domaines de spécialité (*e.g.* traitements en temps réel, biologie, etc) est encore l'objet d'articles scientifiques (Uhl & Wild, 2010). On peut citer certaines variantes de la distance de Levenshtein comme la distance de Jaro-Winkler (Winkler, 1990), plus adaptée aux chaînes courtes, ou la distance de Needleman & Wunsch, qui, parmi d'autres propriétés détaillées dans la section suivante, permet de donner un score de similarité plus précis aux couples d'objets alignés que les trois poids des opérations de la distance de Levenshtein et permet ainsi d'obtenir une distance plus fine.

Outre les mesures de similarité issues du domaine de l'étude des chaînes de caractères, de nombreuses approches par apprentissage permettent de rendre compte de la similarité de deux phrases. Par exemple, Hatzivassiloglou *et al.* (2001) ont créé SIMFINDER, un module de résumé multi-documents qui regroupe des phrases de même sens provenant de différents documents fournis, afin de sélectionner les phrases à inclure dans le résumé. Ils utilisent une approche d'apprentissage supervisé avec divers traits comprenant les groupes nominaux, noms propres, sens dans WordNet et les comptes de mots. Ou encore, dans le domaine de l'identification de paraphrases, Madnani *et al.* (2012) détectent des paraphrases en se basant sur des mesures de traduction automatique et Socher *et al.* (2011) les détectent au moyen d'un réseau de neurones profond. Smith *et al.* (2010), pour leur part, utilisent les alignements de mots utilisés traditionnellement en traduction automatique ainsi que des traits complémentaires pour parvenir aux mêmes fins. Nelken & Shieber (2006), quant à eux, calculent une régression logistique sur la similarité cosinus des TF-IDF d'une paire de phrases afin de déterminer si elle est ou non paraphrastique. Dans le cadre multilingue, Munteanu & Marcu (2005) utilisent l'alignement au niveau des mots entre deux phrases pour déterminer si elles sont paraphrastiques.

Alignement de phrases dans deux documents comparables De la même manière que pour la définition d'une mesure de similarité, la stratégie globale d'alignement à utiliser pour aligner deux séquences d'objets dépend fortement du domaine d'application et de contraintes spécifiques. En conséquence, les algorithmes ont également été proposés dans des domaines variés.

En bio-informatique, un besoin courant est d'aligner deux génomes ou deux séquences d'acides aminés. Ces alignements sont contraints : l'ADN a des séquences non informatives qu'il ne faut pas considérer, les séquences d'acides aminés peuvent varier sans grands effets biologiques (certains acides aminés sont très proches dans leur fonction) et peuvent

4. http://www.multimediaeval.org/mediaeval2015/searchandanchor2015/

également contenir des séquences non informatives. Ces contraintes ont donné naissance à de nouveaux algorithmes. Comme présenté dans l'article de Navarro (2001), l'algorithme central qui a répondu à la contrainte — l'algorithme de Needleman-Wunsch (Needleman & Wunsch, 1970) — a aussi été inventé dans dans la communauté du traitement automatique de la parole sous le nom de Dynamic Time Warping (Vintsyuk, 1968). Cet algorithme a été adapté par Smith & Waterman (1981) pour accentuer l'importance des alignements locaux et diminuer l'importance des trous (*gaps*). Dans leur article, Nelken & Shieber (2006) adaptent la distance de Needleman & Wunsch pour prendre en compte les besoins spécifiques de l'alignement de texte : ne pas pénaliser les alignements de n objets vers 1 objet en particulier. (Bott & Saggion, 2011) proposent, quant à eux, une méthode qui s'appuie sur des modèles de Markov cachés (*Hidden Markov Models* ou *HMM*s) pour modéliser l'alignement de textes contraints (le texte cible est la simplification du texte source dans leur cas).

Dans la continuation des algorithmes basés sur la distance de Levenshtein, on trouve l'algorithme de Myers & Miller (1988), qui calcule en espace linéaire l'alignement entre deux documents. Cet algorithme est la base de tous les outils modernes de différenciation de documents (*diff tools*) tels que ceux trouvés dans les gestionnaires de version comme GIT [5] ou SUBVERSION [6].

Cette famille d'algorithmes est très puissante pour modéliser certains alignements mais manque cependant de généralité : en particulier, un document et sa réorganisation par blocs (*e.g.* un article scientifique et sa présentation orale qui réarrangerait les sections pour mieux mettre en valeur certains points) rompent la linéarité nécessaire au bon fonctionnement de ces algorithmes — les opérations d'insertion, substitution et suppression ne permettent pas de modéliser l'alignement de telles paires de documents sans multiplier les non-correspondances (*mismatchs*). C'est pour cela que des algorithmes différents ont été proposés dans d'autres domaines où ces lacunes sont problématiques.

Par exemple, pour évaluer les systèmes de traduction automatique, Snover *et al.* (2009) introduisent TER-Plus, une extension de la distance de Levenshtein qui prend en compte les réorganisations par blocs, au prix de la NP-complétude du calcul. Dans un domaine différent, Chen *et al.* (2009) modélisent la structure d'un document au sein d'un corpus donné par un processus Bayésien appliqué aux thèmes qu'il aborde. L'alignement de deux documents devient alors la comparaison de l'issue de ce processus pour ces deux documents. Un autre moyen utilisé pour aboutir à des concepts d'alignement plus expressifs a été introduit par Barzilay & Elhadad (2003). Il s'agit d'abord de regrouper les objets dans des ensembles homogènes (*clusters*) sur un critère donné et de manière jointe dans les deux documents à aligner. Une fois ces *clusters* formés, un alignement local peut être effectué au moyen d'un des algorithmes de la famille citée ci-dessus. Ainsi, le réorganisation par blocs des *clusters* peut être modélisée sans problème durant leur formation, hors du processus d'alignement et permettant à celui-ci d'utiliser les algorithmes performants dérivés de l'algorithme de Levenshtein.

3.2 Typologies

L'intérêt de l'expérience d'apprentissage ou d'étude proposée à l'utilisateur dépend directement de la qualité des typologies de liens utilisées pour lier les documents. Pour correctement couvrir les cas d'utilisations envisagés, nous distinguons principalement deux types de typologies : (1) les typologies à ambition ontologique, qui cherchent à modéliser le savoir scientifique ou éducatif présenté, et (2) les typologies rhétoriques, nécessaires dès lors que l'on constate que les communications scientifiques sont éminemment rhétoriques par nature (Bazerman *et al.*, 1988).

Ces typologies ont été spécifiquement étudiées depuis plus de quinze ans (Teufel *et al.*, 1999) et ont abouti à plusieurs schémas d'annotation (Guo *et al.*, 2010). Parmi ceux-ci, nous retenons en particulier le schéma ARGUMENT ZONING II (Teufel *et al.*, 2009) pour la dimension rhétorique qui partitionne un article scientifique en quinze types d'arguments. Par exemple, on y trouve les classes AIM pour souligner un objectif de recherche ou USE pour mentionner l'utilisation des travaux d'autrui dans un travail, etc. Nous retenons aussi le schéma CORESC (Liakata *et al.*, 2010) qui adopte une approche ontologique et a été utilisé en conjonction avec ARGUMENT ZONING II. Liakata *et al.* ont montré que les classes des deux schémas d'annotation n'étaient pas redondantes et couvraient ainsi correctement les différents aspects d'une communication scientifique.

5. http://git-scm.com/
6. https://subversion.apache.org/

3.3 Évaluation et apprentissage par renforcement

Mesurer la qualité d'applications hypermédias est difficile pour deux raisons : (1) Cette qualité est subjective, dépend des besoins d'information de l'utilisateur, de ses habitudes d'apprentissage ou d'étude et de son niveau d'intérêt pour le sujet exposé, et (2) Un résultat négatif comme positif de l'évaluation peut être dû à de nombreuses briques différentes du système, étant donnée sa complexité. Il devient alors difficile d'interpréter les résultats pour améliorer le système.

De manière générale, il est possible d'employer deux stratégies complémentaires pour évaluer un système : (1) mesurer la manière dont il accomplit sa tâche caractéristique (*e.g.*, dans notre cas, aligner des documents et extraire des hyperliens de cet alignement), et (2) mesurer le degré auquel il satisfait les utilisateurs ou systèmes qui consomment ses sorties (*e.g.*, dans notre cas, la qualité de l'étude ou de l'apprentissage de l'utilisateur). Le premier point correspond à une évaluation intrinsèque alors que le second point correspond à une évaluation extrinsèque.

L'évaluation intrinsèque nécessite traditionnellement un ensemble de données annotées, formant une vérité terrain (*gold standard*) afin de pouvoir comparer les sorties du système à des sorties correctes. Il est toutefois prohibitif de créer un tel ensemble de données pour un système hypermédia éducatif complet (qui devrait donc comporter des liens corrects sur suffisament de types entre suffisament de documents pour assurer une significativité statistique des expérimentations). Pour envisager ce type d'évaluation il faut donc se concentrer tour à tour sur des sous-parties du système complet. Nous envisageons un découpage en trois sous-systèmes évaluables :

1. l'alignement, sur le corpus Britannica qui est classique dans ce domaine (Barzilay & Elhadad, 2003; Nelken & Shieber, 2006). Il consiste en plusieurs articles déclinés en deux versions : une version tirée de l'encylopedie Britannica standard et l'autre d'une version simplifiée rédigée de manière indépendante ;
2. la multimodalité, sur la tâche SEARCH & ANCHORING IN VIDEO ARCHIVES de la campagne MEDIAEVAL. Cette campagne propose une référence annotée par des humains pour évaluer la qualité d'un système de recherche d'information hypermédia ;
3. la création d'hyperliens, au cours de la même tâche de MEDIAEVAL.

Dans un premier temps, ces évaluations permettront de se situer par rapport aux approches de la littérature et d'avoir des retours rapides durant les phases de développement de ces sous-systèmes. Elles devront cependant être complétées par une évaluation plus complète, extrinsèque. Les évaluations extrinsèques sont souvent menées par test A/B : les utilisateurs se voient proposer aléatoirement un système témoin ou un système à évaluer et la comparaison des deux repose sur l'analyse de leurs comportements. Dans ce cadre, Radlinski *et al.* (2008) montrent que les métriques facilement collectables (*e.g.* clics, temps de visite) ne reflètent pas la qualité des systèmes évalués. Pour pallier ce problème, ils introduisent un mécanisme astucieux de combinaison des systèmes qui permet une évaluation correcte des performances.

La mise en place d'un dispositif analogue d'utilisation des retours des utilisateurs est nécessaire pour apprendre par renforcement (ce qui, comme mentionné en section introduction de cet état de l'art, est nécessaire pour pallier le manque de données). Radlinski *et al.* (2008) montrent qu'en conséquence cela peut être fait conjointement. Le processus de développement du système consiste alors à présenter aux utilisateurs un système évalué intrinsèquement puis à l'évaluer de manière extrinsèque et régulière pour effectuer et mesurer des améliorations.

4 Questions de recherche

Alignement Nous distinguons trois grandes questions :

1. Comment modéliser les alignements qui tolèrent les réorganisations par blocs pour pouvoir les apprendre efficacement ?
2. Comment segmenter le texte extrait des documents écrits et celui provenant de la transcription automatique de la vidéo pour que leur alignement et leur parcours soient aisés ?
3. Quelles sont les mesures de similarité pertinentes dans un contexte multimodal et faut-il leur apporter des modifications pour une pleine efficacité dans la tâche d'alignement au niveau du document ?

Répondre à la question (1) constitue un premier pas vers une adaptation des algorithmes d'alignement de la famille de l'étude des chaînes de caractères aux problèmes nécessitant une tolérance aux réorganisations par blocs. Cette adaptation permettrait d'avoir un cadre théorique et opérationnel fiable pour gérer les alignements multimodaux. Les réponses aux questions (2) et (3), quant à elles, sont primordiales pour pallier le manque de structure des modalités textuelles extraites et sont certainement les questions centrales de la gestion de la multimodalité dans cette étude.

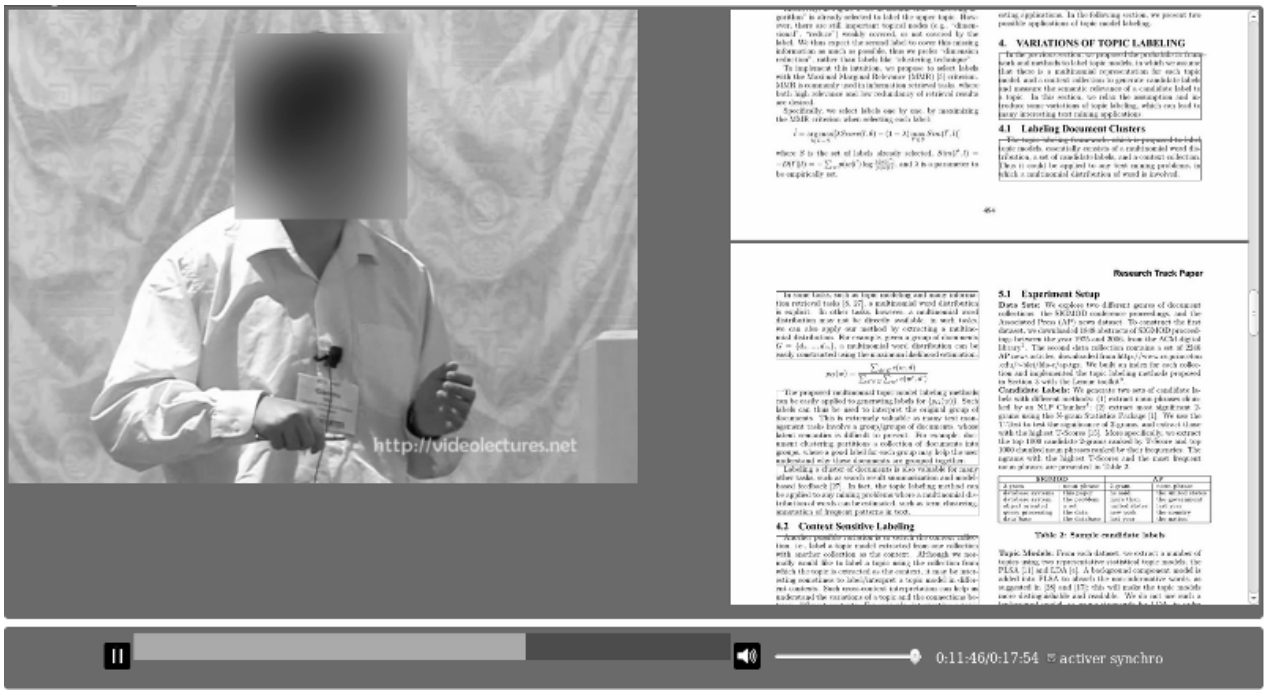

FIGURE 1 – Capture d'écran du prototype de navigation jointe.

Typologie Les schémas d'annotation rhétorique offrent de grands bénéfices aux utilisateurs mais il est extrêmement coûteux de produire des corpus annotés pour entraîner des systèmes d'apprentissage supervisé. La question de recherche intéressante qui découle de ce constat est la suivante : est-il possible d'apprendre à appliquer ces schémas de manière non supervisée ? Des études se sont récemment orientées vers ces questions (Guo *et al.*, 2011) mais utilisent encore des données annotées ou des méthodes d'apprentissage actif.

Évaluation L'approche envisagée est basée sur le travail de Radlinski *et al.* (2008) traitant des systèmes de recommandation. Les applications hypermédias, bien qu'elles aient de nombreuses similarités avec ces systèmes (*i.e.* il est possible de voir les hyperliens comme des recommandations faites aux utilisateurs pour satisfaire leurs besoins d'information), ont des caractéristiques propres. Il est donc nécessaire d'adapter les stratégies et métriques de la littérature pour obtenir un retour utilisateur automatique efficace. Ce travail est en soi une question de recherche à part entière.

5 Prototypes actuels

Nous avons développé trois prototypes [7] pour étudier certaines des questions de recherche mentionnées en Section 4. À ce stade les prototypes se concentrent sur l'alignement multimodal non typé. Le code est disponible sur GITHUB [8] [9]

Illustré par la Figure 1, le premier prototype permet de naviguer de manière jointe dans une présentation enregistrée et dans un article si leur alignement est disponible : cliquer sur un paragraphe entraîne la lecture du segment vidéo lui correspondant et de la même manière, lire la vidéo surligne les parties de l'article liées au sujet courant. Ce prototype utilise POPPLER [10] pour délimiter spatialement les paragraphes d'un article.

Le deuxième prototype, montré en Figure 2, sert à calculer les alignements entre un article scientifique et la vidéo de la présentation correspondante et à étudier leur qualité. Il implémente pour l'instant une approche de base en utilisant les similarités cosinus sur les TF-IDF des phrases de l'article et des segments de la transcription vidéo pour aligner les deux

7. http://alignement.comin-ocw.org/
8. https://github.com/m09/alignment-demo
9. https://github.com/m09/alignment
10. http://poppler.freedesktop.org/

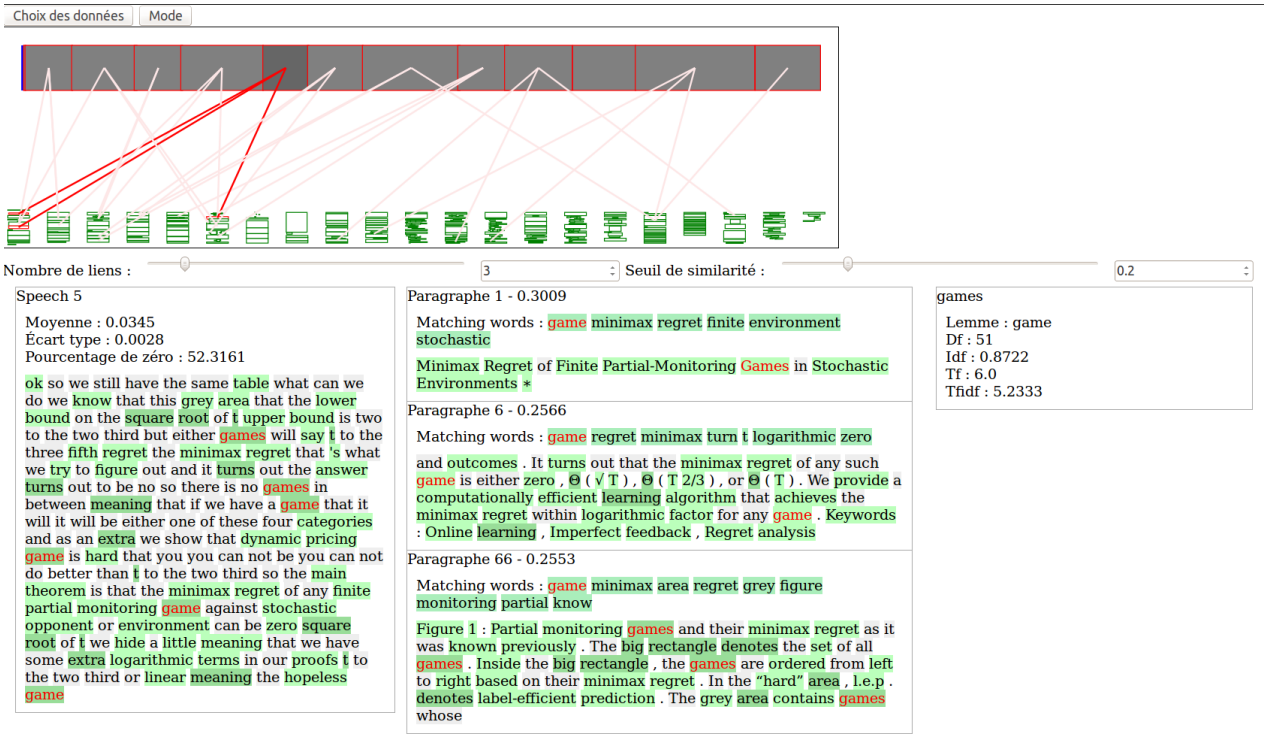

FIGURE 2 – Capture d'écran du prototype d'alignement.

médias. Ces calculs sont réalisés en python avec le framework NLTK [11]. Il présente en résultat les alignements obtenus sur une interface web réalisée avec D3.JS [12].

Quant au troisième prototype, il se concentre sur l'implémentation des algorithmes et métriques de la littérature et l'utilisation des corpus liés (en particulier les approches de Barzilay & Elhadad (2003) et Nelken & Shieber (2006)). Il utilise le framework UIMA [13], les librairies open source OPENNLP [14] et GROBID.

Choix d'utilisabilité Le design d'interface et l'étude d'utilisabilité constituent un travail en cours de réalisation. Pour l'instant, les prototypes sont pensés pour favoriser des itérations de développement rapides et une analyse facile pour les développeurs au détriment du confort d'utilisation pour un utilisateur non expert.

6 Conclusion

Dans cet article, nous avons présenté le problème d'utilisation optimale des ressources éducatives et de recherche en cours d'étude dans le projet COCO. Il mobilise des idées et approches de nombreux domaines de l'informatique. Nous avons ensuite introduit la littérature sur lesquels notre étude s'appuie. Cela nous a permis de discuter quelques pistes pour de futures recherches : (1) la modélisation des alignements qui tolèrent des réorganisations par blocs, (2) les mesures de similarité et segmentations à utiliser pour limiter l'effet détériorant des modalités textuelles bruitées extraites sur le système d'alignement, (3) l'annotation automatique des documents scientifiques respectant des schémas d'annotation rhétoriques et ontologiques et minimisant les données annotées nécessaires, et (4) l'évaluation extrinsèque et l'apprentissage par renforcement dans le cadre d'un système hypermédia. Nous avons fini par présenter trois logiciels en cours de développement qui permettent de débuter l'analyse de ces questions.

11. http://www.nltk.org/
12. http://d3js.org/
13. https://uima.apache.org/uimafit.html
14. https://opennlp.apache.org/

7 Remerciements

Je remercie mes co-auteurs — Matthieu RIOU, Colin DE LA HIGUERA, Solen QUINIOU et Olivier AUBERT — de l'article (Mougard *et al.*, 2015) sur lequel sont basées les sections de cet article qui ne présentent ni l'état de l'art ni les questions de recherche liées au problème étudié.

Nous remercions aussi l'Agence Nationale de la Recherche pour son soutien au programme « Investissements pour le Futur » sous la référence ANR-JO-LABX-07-0J (projet COCO). Nous remercions aussi les collègues de de JSI, Ljubljana pour l'aide qu'ils nous ont apportée ainsi que VIDEOLECTURES pour le matériel multimédia mis à disposition.

References

BARZILAY R. & ELHADAD N. (2003). Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, p. 25–32: Association for Computational Linguistics.

BAZERMAN C. *et al.* (1988). *Shaping written knowledge: The genre and activity of the experimental article in science*. University of Wisconsin Press Madison.

BOTT S. & SAGGION H. (2011). An unsupervised alignment algorithm for text simplification corpus construction. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, p. 20–26, Stroudsburg, PA, USA: Association for Computational Linguistics.

BRUNNING J. J. J. (2010). *Alignment Models and Algorithms for Statistical Machine Translation*. PhD thesis, University of Cambridge.

CHEN H., BRANAVAN S., BARZILAY R., KARGER D. R. *et al.* (2009). Content modeling using latent permutations. *Journal of Artificial Intelligence Research*, **36**(1), 129–163.

ESKEVICH M., MAGDY W. & JONES G. J. (2012). New metrics for meaningful evaluation of informally structured speech retrieval. In *Advances in Information Retrieval*, p. 170–181. Springer.

GALUŠČÁKOVÁ P. (2013). Segmentation strategies for passage retrieval in audio-visual documents. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, p. 1143–1143, New York, NY, USA: ACM.

GUO Y., KORHONEN A., LIAKATA M., KAROLINSKA I. S., SUN L. & STENIUS U. (2010). Identifying the information structure of scientific abstracts: An investigation of three different schemes. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, BioNLP '10, p. 99–107, Stroudsburg, PA, USA: Association for Computational Linguistics.

GUO Y., KORHONEN A. & POIBEAU T. (2011). A weakly-supervised approach to argumentative zoning of scientific documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, p. 273–283, Stroudsburg, PA, USA: Association for Computational Linguistics.

HAMMING R. W. (1950). Error detecting and error correcting codes. *Bell System Technical Journal*, **29**(2), 147–160.

HATZIVASSILOGLOU V., KLAVANS J. L., HOLCOMBE M. L., BARZILAY R., YEN KAN M. & MCKEOWN K. R. (2001). Simfinder: A flexible clustering tool for summarization. In *In Proceedings of the NAACL Workshop on Automatic Summarization*, p. 41–49.

LEVENSHTEIN V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, p. 707–710.

LIAKATA M., TEUFEL S., SIDDHARTHAN A. & BATCHELOR C. R. (2010). Corpora for the conceptualisation and zoning of scientific papers. In *LREC*.

MADNANI N., TETREAULT J. & CHODOROW M. (2012). Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, p. 182–190, Stroudsburg, PA, USA: Association for Computational Linguistics.

MOUGARD H., RIOU M., DE LA HIGUERA C., QUINIOU S. & AUBERT O. (2015). The paper or the video: Why choose? In *Proceedings of the Companion Publication of the 24th International Conference on World Wide Web Companion*, WWW Companion '15, p. In press, Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.

MUNTEANU D. S. & MARCU D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Comput. Linguist.*, **31**(4), 477–504.

MYERS E. W. & MILLER W. (1988). Optimal alignments in linear space. *Computer applications in the biosciences: CABIOS*, **4**(1), 11–17.

NAVARRETE T. & BLAT J. (2002). VideoGIS: Segmenting and indexing video based on geographic information. In *5th AGILE Conference on Geographic Information Science*, p. 1–9.

NAVARRO G. (2001). A guided tour to approximate string matching. *ACM Comput. Surv.*, **33**(1), 31–88.

NEEDLEMAN S. B. & WUNSCH C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, **48**(3), 443–453.

NELKEN R. & SHIEBER S. M. (2006). Towards robust context-sensitive sentence alignment for monolingual corpora. In *In Proc. EACL*: Association for Computational Linguistics.

RADLINSKI F., KURUP M. & JOACHIMS T. (2008). How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM conference on Information and knowledge management*, p. 43–52: ACM.

SADALLAH M., AUBERT O. & PRIÉ Y. (2012). CHM: an Annotation- and Component-based Hypervideo Model for the Web. *Multimedia Tools and Applications*.

SMEATON A. F., OVER P. & TABAN R. (2001). The TREC-2001 video track report. *Proceedings of TREC-2001*.

SMITH J. R., QUIRK C. & TOUTANOVA K. (2010). Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, p. 403–411: Association for Computational Linguistics.

SMITH T. F. & WATERMAN M. S. (1981). Identification of common molecular subsequences. *Journal of molecular biology*, **147**(1), 195–197.

SNOVER M., MADNANI N., DORR B. & SCHWARTZ R. (2009). Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, **23**(2-3), 117–127.

SOCHER R., HUANG E. H., PENNIN J., MANNING C. D. & NG A. Y. (2011). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, p. 801–809.

TEUFEL S., CARLETTA J. & MOENS M. (1999). An annotation scheme for discourse-level argumentation in research articles. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, p. 110–117, Stroudsburg, PA, USA: Association for Computational Linguistics.

TEUFEL S., SIDDHARTHAN A. & BATCHELOR C. (2009). Towards discipline-independent argumentative zoning: evidence from chemistry and computational linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, p. 1493–1502: Association for Computational Linguistics.

UHL A. & WILD P. (2010). Enhancing iris matching using levenshtein distance with alignment constraints. In *Advances in Visual Computing*, p. 469–478. Springer.

VINTSYUK T. K. (1968). Speech discrimination by dynamic programming. *Cybernetics and Systems Analysis*, **4**(1), 52–57.

WINKLER W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *ERIC*.

État de l'art : analyse des conversations écrites en ligne porteuses de demandes d'assistance en termes d'actes de dialogue

Soufian Salim
LINA UMR 6241 - Université de Nantes, 2 rue de la houssinière, 44322 Nantes Cedex 03
soufian.salim@univ-nantes.fr

Résumé. Le développement du Web 2.0 et le processus de création et de consommation massive de contenus générés par les utilisateurs qu'elle a enclenché a permis le développement de nouveaux types d'interactions chez les internautes. En particulier, nous nous intéressons au développement du support en ligne et des plate-formes d'entraide. En effet, les archives de conversations en ligne porteuses de demandes d'assistance représentent une ressource inestimable, mais peu exploitée. L'exploitation de cette ressource permettrait non seulement d'améliorer les systèmes liés à la résolution collaborative des problèmes, mais également de perfectionner les canaux de support proposés par les entreprises opérant sur le web. Pour ce faire, il est cependant nécessaire de définir un cadre formel pour l'analyse discursive de ce type de conversations. Cet article a pour objectif de présenter l'état de la recherche en analyse des conversations écrites en ligne, sous différents médiums, et de montrer dans quelle mesure les différentes méthodes exposées dans la littérature peuvent être appliquées à des conversations fonctionnelles inscrites dans le cadre de la résolution collaborative des problèmes utilisateurs.

Abstract.

State of the Art : analysis of problem-oriented online written conversations in terms of dialog acts

The advent of Web 2.0 and the massive creation and consumption of user generated content it triggered allowed for new kinds of interactions among web users. Indeed, freely available archives of problem-oriented online conversations represent an invaluable resource, however it remains largely unexploited. Exploiting this resource would not only allow for the improvement of systems related to collaborative problem resolution, but also to refine the customer support channels that web-based companies put at their users' disposal. However, in order to achieve this, it is first necessary to define a formal framework for the fine-grained analysis of online written conversations bearing requests for assistance. This paper presents an overview of the state of the art for discourse analysis in online written conversations, and discusses the applicability of the different methods for the analysis of conversations set within the scope of collaborative problem-solving.

Mots-clés : Analyse discursive, Conversation, Résolution de problèmes, Schéma d'annotation, Acte de dialogue.

Keywords: Discourse analysis, Conversation, Problem solving, Annotation scheme, Dialog act.

1 Introduction

Le développement d'Internet a déclenché des révolutions majeures. Parmi elles, l'avènement du contenu généré par les utilisateurs. Avec le développement du Web 2.0 et l'accroissement des capacités d'interaction entre les internautes, nous avons été témoins d'un phénomène de démultiplication de l'information en ligne. Voilà plus de dix ans que les réseaux sociaux, les blogs, les forums et les autres formes de médias interactifs en ligne sont devenus d'usage courant. Sur ces plate-formes, de nombreux types d'interactions peuvent être identifiés. Nous nous intéressons au cas des conversations orientées vers la résolution de problèmes, *i.e.* celles s'opérant sur des plate-formes dont les utilisateurs sont invités à transmettre leurs demandes d'assistance à la communauté, qui en retour tente d'apporter son aide. Par exemple, demander comment configurer un routeur, où trouver une pièce pour réparer son véhicule, ou encore comment préparer des sushis. Ces conversations se retrouvent, notamment, sur des forums d'entraide (*e.g.* CommentÇaMarche, CNET), des listes de diffusion [1], et des salons de chat (*e.g.* canaux IRC). L'exploitation de cet ensemble massif de données représente un enjeu majeur pour les scientifiques et industriels qui s'intéressent aux problématiques liées à l'assistance aux utilisateurs.

Ce type de conversation se retrouve également dans des environnements privés et plus strictement contrôlés : les canaux

1. Aussi appelées "listes de discussion".

d'assistance en ligne mis à disposition directement par les entreprises, pour leurs clients. L'importance de ces canaux n'est pas négligeable : disposer d'un service client en ligne efficace est devenu une part intégrale du succès pour les entreprises opérant sur le net. Les compagnies basées sur le web savent depuis longtemps que le service client des commerces virtuels est tout aussi important que pour les magasins traditionnels (Bernett, 2000).

Ces demandes d'assistance sont le plus souvent gérées au cas par cas, ce qui mène à un nombre considérable de conversations redondantes pour des problèmes et questions communément soulevés. Dans le cas des forums et des listes de diffusion, les conversations sont généralement perpétuellement sauvegardées, permettant ainsi le partage de support entre utilisateurs. C'est-à-dire qu'elles dotent les utilisateurs de la capacité de faire des recherches dans les archives de la plateforme pour essayer de trouver une solution documentée et directement applicable à leur situation. Bien que moins souvent exploités, les messages transmis dans les salons de chat peuvent aussi faire l'objet d'un archivage automatique. Dans le cas des systèmes de support client, les conversations sont toujours enregistrées et sont généralement manuellement exploitées. Cette exploitation se place, notamment, dans le cadre de l'évaluation des agents, de l'amélioration des techniques de marketing, ou de l'enrichissement des bases de connaissances et des FAQ (Foire Aux Questions).

Un cadre bien défini permettant une analyse fine de ce type de conversations représenterait un socle solide sur lequel pourraient reposer différents systèmes liés à l'aide à la résolution des problèmes. Comme, par exemple, des systèmes de RI plus performants, pour aider les utilisateurs cherchant des solutions à leurs problèmes dans des archives de conversations. Ou encore, des systèmes automatiques d'enrichissement de bases de connaissances et de FAQ à partir de conversations résolues. Il est aussi possible d'imaginer des systèmes automatisés d'aide à la résolution des problèmes qui proposeraient automatiquement des solutions aux utilisateurs, ou au moins les redirigeraient directement vers un message comportant une solution validée pour un problème identique. Toutes ces applications seraient considérablement enrichies par les masses énormes de données qui sont actuellement librement disponibles sur Internet, mais très peu exploitées.

Cet article a pour objectif de présenter l'état de la recherche en modélisation des conversations écrites en ligne, et de montrer dans quelle mesure les différentes approches proposées dans la littérature sont adaptées ou non à des applications liées à la problématique de l'assistance aux utilisateurs. Nous commençons par nous intéresser aux théories fondatrices de l'analyse du dialogue, dans la section 2. Ensuite, en section 3, nous verrons comment leurs concepts peuvent être appliqués via deux schémas d'annotation influents : DAMSL (*Dialog Act Markup in Several layers*) et DIT++ (*Dynamic Information Theory* ++). Puis, en section 4, nous examinerons divers travaux spécifiques à différents médiums en ligne (forums, courriels, chats), tout en gardant à l'esprit la problématique de l'assistance aux utilisateurs. Enfin, en section 5, nous apporterons nos conclusions et présenterons quelques perspectives de recherche.

2 Théories pour l'analyse du dialogue

Nous nous intéressons dans cette section aux actes de discours [2], qui sont largement utilisés dans les études sur les phénomènes conversationnels, pour l'annotation de dialogues, et dans la conception d'agents conversationnels [3]. Nous commençons par expliquer ce qu'est la théorie des actes de discours avant de voir comment ces fondements théoriques ont été étendus vers l'analyse des conversations au travers des travaux de Traum & Hinkelman (1992) et Poesio & Traum (1997).

2.1 Actes de discours

En linguistique et en philosophie du langage, un acte de discours est un énoncé porteur d'une fonction performative. En effet, pour Austin, qui a introduit le terme dans la langue contemporaine, les énonciations doivent être considérées comme des actions effectuées par le locuteur. Apparaît ici l'idée selon laquelle tout acte d'énonciation serait la réalisation d'un acte social. Les verbes qui spécifient ces actions sont appelés verbes performatifs (*i.e. « Je vous confère le titre de capitaine »*). Mais les actes de discours ne sont pas constitués uniquement de ces types de verbes.

Austin (1975) développe une théorie des actes de discours défendant la thèse selon laquelle tout énoncé peut être analysé à trois niveaux. D'abord, au niveau locutoire : il s'agit de sa forme de surface, *i.e.* de la signification de l'énoncé, représenté par ses aspects phonétique, syntaxique et sémantique. Puis au niveau de l'acte illocutoire, porteur de l'intention rhétorique du locuteur. Et enfin, au niveau de l'acte perlocutoire, qui s'intéresse aux conséquences de l'exécution de l'énoncé ou de son interprétation par les allocutaires : à son effet pragmatique.

2. Aussi appelés « actes de langage »
3. Systèmes informatiques conçus pour converser avec des êtres humains.

Austin (1975)	Searle (1976)	Exemples
Expositifs	Assertifs	affirmer, nier, postuler, remarquer...
Exercitifs	Directifs	commander, conseiller, ordonner, pardonner, léguer...
Promissifs	Promissifs	promettre, inviter, faire vœu de, garantir, parier, jurer de...
Comportatifs	Expressifs	s'excuser, remercier, féliciter, déplorer, critiquer...
Verdictifs	Déclaratifs	acquitter, condamner, décréter, baptiser...

TABLE 1 – Taxonomies fondatrices en théorie des actes de discours, alignées

La notion d'acte illocutoire est centrale au concept d'acte de discours. Cet acte permet de décrire les énoncés en termes de fonctions communicatives portées par chacun d'eux (*e.g.* question, réponse, remerciement...). Austin propose cinq classes d'actes de discours : les verdictifs (qui donnent un verdict), les exercitifs (qui exercent un pouvoir), les promissifs (qui engagent le locuteur), les comportatifs (qui expriment l'attitude) et les expositifs (qui exposent de l'information).

Pour Searle (1969), dont la conception des actes de discours diffère légèrement de celle d'Austin, tout acte de discours est illocutoire (sa définition se rapproche ainsi de ce que Austin appelle "acte de dialogue"). Il propose cinq classes d'actes : les assertifs (affirment un état de fait), les directifs (poussent l'interlocuteur à agir), les promissifs (engagent le locuteur), les expressifs (expriment un état psychologique) et les déclaratifs (ont un impact réel, *e.g.* prononcer un jugement) (Searle, 1976). La table 1 illustre ces deux taxonomies fondatrices et montre comment elles peuvent être alignées.

Historiquement, cette théorie a rapidement gagné en influence dans un ensemble de disciplines varié. En psychologie, par exemple, il a été suggéré que l'acquisition des actes de discours puisse être un prérequis pour l'acquisition du langage (Bruner, 1975). Des experts littéraires se sont tournés vers Austin pour mettre en lumière des particularités textuelles (Ohmann, 1971). En linguistique, des chercheurs ont trouvé que des notions de la théorie des actes de discours permettaient d'expliquer des problèmes en sémantique (Fillmore, 1971), en syntaxe (Sadock, 1974) et en apprentissage d'un second langage (Jakobovits & Gordon, 1974). Même en philosophie, des applications pouvaient être trouvées, par exemple pour déterminer le statut de postulats éthiques (Searle, 1969).

En informatique, les actes de discours sont communément utilisés pour modéliser les conversations dans le cadre d'applications de classification automatique et de recherche d'information (Twitchell *et al.*, 2004). Des modèles pour l'interaction homme-machine ont également été développés en se basant sur ces concepts (Morelli *et al.*, 1991). Ainsi, dans la plupart des travaux liés à la linguistique informatique, c'est en termes d'actes de discours que les interactions entre participants d'une conversation sont modélisées.

2.2 Applications à l'analyse des conversations fonctionnelles

Jusque dans les années 1990, la théorie des actes de discours s'est largement limitée à l'examen d'énoncés isolés, et n'a pas cherché à prendre en charge l'analyse de conversations entières où plusieurs participants peuvent interagir (Vanderveken, 1992). Cependant, les locuteurs accomplissent des actes illocutoires tout au long des conversations qu'ils peuvent avoir avec d'autres participants. Vanderveken souligne que ces derniers répondent et accomplissent à leur tour leurs propres actes de discours, tout en cherchant collectivement à atteindre des objectifs communs. Une application sociale du langage est donc constituée, en général, de séquences ordonnées d'énoncés par différents locuteurs qui cherchent ensemble à poursuivre un même but, comme décider d'une marche à suivre, résoudre un problème, accomplir une action, etc.

Dans le cas de ces deux derniers exemples, on parlerait de « conversations fonctionnelles », *i.e.* de conversations construites autour d'une tâche, c'est-à-dire consacrées à la transmission d'information dans le but de réaliser un objectif individuel ou collectif dans le monde réel. C'est à ce type de conversations, qui inclue notamment les conversations porteuses de demandes d'assistance, que s'intéressent la plupart des travaux cherchant à étendre la théorie des actes de discours aux interactions multipartites. Cela s'explique par le fait que les applications informatiques de l'analyse du dialogue sont presque toujours motivées par le besoin de faciliter ou d'automatiser l'exécution d'une tâche par un utilisateur humain.

2.2.1 Théorie des actes de la conversation

Dans cette perspective d'extension de la théorie des actes de discours, Traum & Hinkelman (1992) décrivent une théorie des *actes de la conversation* (*Conversation Act Theory*), qui se veut plus générale. Ils étudient le corpus TRAINS (Gross

et al., 1993)[4], tiré du projet homonyme, dont l'objectif est de développer un assistant de planification intelligent qui puisse communiquer en langage naturel avec des opérateurs humains. Le corpus est constitué de dialogues fonctionnels entre un manager devant résoudre des problèmes de planification et une personne jouant le rôle du système, disposant d'informations additionnelles sur la tâche, et chargé d'assister le manager.

Traum & Hinkelman constatent que l'un des traits les plus flagrants des dialogues fonctionnels est la prépondérance des signes d'accord et d'acquittement (*e.g. « There are oranges at Corning, right ? » « Right. »*). C'est l'un des éléments qui les poussent à remettre en question certains postulats généralement implicites dans les travaux antérieurs. Le premier de ces postulats voudrait que les énoncés soient toujours entendus et correctement compris par les allocutaires, d'une part, et que les participants ne s'attendent jamais à ce que ce ne soit pas le cas, d'autre part. Mais non seulement les énoncés sont souvent mal compris ou mal perçus, mais en plus Traum & Hinkelman avancent que les conversations sont structurées de manière à prendre en compte ce phénomène : les participants cherchent systématiquement à obtenir des preuves que leur interlocuteur a bien compris ce qu'il voulaient dire. Ces preuves peuvent prendre la forme d'un acquittement explicite (*e.g. « Right. »*), d'un acquittement implicite via une réaction pertinente (par exemple en répondant à la question posée), ou encore par des signaux non-verbaux (hochement de tête, *etc.*). Cette quasi-nécessité de l'acquittement les pousse également à remettre en cause l'idée selon laquelle les actes de discours sont des actions réalisées uniquement par le locuteur, et que l'allocutaire n'a qu'une fonction passive face à eux. Les actes de discours ne peuvent être analysés que dans le contexte d'un dialogue multi-agent. Enfin, le troisième postulat que Traum & Hinkelman remettent en cause suite à cette observation, c'est que chaque énoncé n'est porteur que d'un seul acte de discours. En effet, si certains énoncés peuvent non seulement réaliser leur fonction communicative affichée et en plus servent à acquitter un autre énoncé, c'est qu'ils peuvent réaliser deux actes simultanément.

La taxonomie des actes de la conversation qu'ils proposent prend en compte ces trois observations. Elle détaille une catégorisation de ces actes en quatre classes : les actes de prise de parole (*turn-taking acts*), les actes de synchronisation (*grounding acts*), les actes de discours fondamentaux (*core speech acts*), et les actes argumentatifs (*argumentation acts*). Les actes de prise de parole se situent à un niveau inférieur à l'énoncé, les actes de synchronisation au niveau de l'énoncé, tandis que les actes de discours fondamentaux (informer, promettre et requérir) se trouvent au niveau de ce qu'ils appellent une « unité du discours ». Cette unité peut contenir un énoncé introductif suivi d'autant d'énoncés de synchronisation que nécessaire pour assurer une bonne communication (*e.g. « Because there are oranges in Vermont. Right ? You agree ? »*). Enfin, les actes argumentatifs se situent à un niveau encore supérieur puisqu'ils peuvent contenir un nombre illimité d'unités du discours dont les actes fondamentaux sont utilisés pour former des composés complexes (par exemple le descriptif d'un système, l'exposé d'un problème *etc.*).

2.2.2 Contexte et connaissances communes

Poesio & Traum s'accordent à dire que les conversations, même fonctionnelles, ont des aspects nettement séparés de la réalisation de la tâche qui en est l'objet, et que l'exercice du langage est une action coordonnée, ce qui impose le développement d'une théorie du *contexte* (Poesio & Traum, 1997). Les théories développées à ce sujet se déclinent en deux traditions : d'une part, les approches linguistiques construites autour notamment de la résolution d'anaphores, et d'autre part les modèles computationnels proposés pour représenter les effets des actes de discours sur les participants d'une conversation, par exemple en termes de croyances, d'obligations et de besoins. C'est cette deuxième approche qui nous intéresse, puisque la première n'a que peu de rapport avec les exercices de planification et de coordination de l'information qui sont propres aux conversations fonctionnelles, et *a fortiori* aux conversations orientées vers la résolution de problèmes. Si la résolution d'anaphores peut évidemment présenter un intérêt pour suivre le fil des conversations, ce problème purement linguistique doit être traité séparement de la question de la synchronisation inter-participants.

Quand Poesio & Traum parlent de contexte, ils font référence à l'information que les participants doivent utiliser pour interpréter les énoncés d'une conversation. Ce contexte est caractérisé notamment par la notion, centrale, de *connaissances communes*, ou « terrain d'entente » (*common ground*) entre les participants. Cette information est cruciale pour pouvoir comprendre à quoi un énoncé fait référence, puisque c'est le contexte qui contient tous les référents disponibles, les référents étant ajoutés aux connaissances communes au travers des nouveaux actes de discours qui sont accomplis. Bien modéliser ces connaissances nécessite de bien modéliser les mises à jour du contexte, ce qui est l'objet des deux schémas d'annotation décrits en section 3.

4. Traum & Hinkelman ont utilisé des données tirées de (Gross *et al.*, 1993) avant leur publication, d'où l'apparente incohérence des dates.

3 Schémas d'annotation

L'annotation des conversations en termes d'actes de discours peut suivre deux approches. La première, ontologique, consiste à proposer une taxonomie spécifique au domaine ou à la tâche étudiée. La seconde, plus ambitieuse, cherche à atteindre une couverture plus générique du dialogue (Leech & Weisser, 2003). C'est le cas de deux schémas d'annotation largement utilisés : DAMSL et DIT++. Ce sont plutôt des actes de la conversation au sens de Traum & Hinkelman que de purs actes de discours qu'ils cherchent à modéliser. Ils apparaissent dans ces travaux sous le nom d'*actes de dialogue*.

3.1 DAMSL

Le schéma d'annotation DAMSL part du principe que les applications nécessitant une analyse automatique du dialogue doivent prendre en compte les modifications dynamiques du « terrain d'entente », et pour ce faire propose d'annoter les *fonctions communicatives* des actes de dialogue. Pour Core & Allen (1997), ces fonctions doivent représenter des manipulations directes du contexte informationnel d'une conversation. La première caractéristique de la taxonomie de DAMSL est donc qu'elle permet d'annoter les actes en tant qu'*opérations de mise à jour du contexte*.

Un deuxième aspect important du schéma DAMSL est sa multi-dimensionnalité. En effet, une des limites de la théorie des actes de discours d'Austin et Searle, qui a été souvent soulignée par les chercheurs, est son incapacité à prendre en compte la pluralité des intentions qu'un locuteur peut chercher à exprimer dans un seul énoncé. Comme préconisé par Traum & Hinkelman (1992), la taxonomie proposée par Core & Allen prend en compte ce problème et autorise l'application de plusieurs étiquettes à un seul énoncé.

Un troisième attribut important de DAMSL, et probablement celui qui a le plus contribué à sa popularité, est son caractère générique. Les annotations proposées sont toutes de suffisamment haut niveau pour pouvoir être appliquées à différents types de dialogues. Néanmoins, le schéma se focalise nettement sur les conversations fonctionnelles, et a d'ailleurs été d'abord développé autour du même corpus TRAINS que les travaux que nous avons détaillé en sous-section 2.2.

3.1.1 Quatre catégories d'étiquettes : fonctions prospectives, fonctions rétrospectives, niveau d'information et statut communicatif

Certains énoncés sont de toute évidence liés entre eux. Par exemple, prenons l'échange suivant :

1. Participant 1 : « *Le ciel commence à se dégager.* »
2. Participant 1 : « *Quelle heure est-il ?* »
3. Participant 2 : « *Il est bientôt midi.* »

Il est immédiatement apparent que l'énoncé 3 fait réponse à l'énoncé 2, et diffère en ce sens de l'énoncé 1, qui n'a pas été sollicité. Pourtant, les deux apportent une information factuelle au contexte, et pourraient être classés comme "informer". Core & Allen notent que si des travaux avaient déjà tenté de répondre à ce problème en proposant des sous-classes de type "informer-répondre" ou "informer-accepter", ce n'est pas satisfaisant car les actes d'acquitter, de répondre ou d'accepter un énoncé semblent appartenir à un genre d'actes bien distinct de "informer". Ils disent des fonctions de ces actes qu'elles sont *rétrospectives* (*backward-looking*), puisqu'elles sont orientées vers la partie antérieure de la conversation. Les autres fonctions (*e.g.* affirmer, ordonner, promettre, *etc.*) sont donc dites *prospectives* (*forward-looking*), puisqu'elles en impactent la partie ultérieure. Ces deux groupes de fonctions constituent les deux premières catégories de d'étiquettes [5] de la taxonomie DAMSL. Ce sont elles qui permettent d'étiqueter les énoncés par leur intention communicative.

Les deux autres catégories définies par DAMSL sont celles des traits énonciatifs (*Utterance Features*). Ces traits ne s'intéressent pas à la fonction communicative de l'énoncé, mais capturent les propriétés de son contenu. Ils indiquent sur quoi l'énoncé porte (si il porte directement sur la tâche, sur le processus de communication, du processus de résolution de la tâche, ou d'autre chose) : c'est la catégorie *niveau d'information* (*Information Level*). Ils permettent également d'identifier les énoncés qui peuvent être ignorés sans danger (parce qu'incompréhensibles ou interrompus) : c'est la catégorie *statut communicatif* (*Communicative Status*).

5. Ces super catégories sont appelées « couches » (*layers*) dans la documentation de DAMSL.

3.1.2 Taxonomie

Les quatre catégories de la taxonomie sont détaillées en figure 1 [6] :

Fonctions rétrospectives :
- *Agreement*
  - *Accept*
  - *Accept-Part*
  - *Maybe*
  - *Reject-Part*
  - *Reject*
  - *Hold*
- *Understanding*
  - *Signal-Non-Understanding*
  - *Signal-Understanding*
    - *Acknowledge*
    - *Repeat-Rephrase*
    - *Completion*
  - *Correct-Misspeaking*
- *Answer*
- *Information-Relation*

Fonctions prospectives :
- *Statement*
  - *Assert*
  - *Reassert*
  - *Other-Statement*
- *Influencing Addressee Future Action*
  - *Open-Option*
  - *Directive*
    - *Info-Request*
    - *Action-Directive*
- *Committing Speaker Future Action*
  - *Offer*
  - *Commit*
- *Performative*
- *Other Forward Function*

Niveau d'information :
- *Task*
- *Task Management*
- *Communication Management*
- *Other*

Statut communicatif :
- *Abandoned*
- *Uninterpretable*
- *Self-talk*

FIGURE 1 – Taxonomie DAMSL

Les classes situées au premier niveau des listes imbriquées sont appelées « dimensions » par Core & Allen. Les dimensions sont indépendantes les unes des autres. Ainsi par exemple un énoncé peut à la fois acquitter une question (*Acknowledge*) et y répondre (*Answer*). Toutes les dimensions sont optionnelles. Tous les énoncés n'ont pas non plus forcément une étiquette dans chaque catégorie (par exemple, il est possible d'avoir une fonction prospective mais aucune fonction rétrospective), à l'exception du niveau d'information qui doit toujours être indiqué.

DAMSL est le premier schéma d'annotation à implémenter une approche multidimensionnelle, permettant d'assigner de multiples étiquettes aux énoncés. L'utilité de la taxonomie qui y est décrite est prouvée par le nombre important de travaux s'appuyant dessus, ce qui en fait *de facto* un standard en analyse du dialogue. Cependant, comme souligné par Bunt (2006), les dimensions et les « couches » (les quatre catégories d'étiquettes) employées dans DAMSL ne sont pas discutées, manquent de signification conceptuelle, et ne s'appuient sur aucun fondement théorique. Comme nous le verrons dans la sous-section suivante, DIT++, lui, tente de proposer un système fondé sur des bases théoriques solides.

3.2 DIT++

DIT++, comme DAMSL, cherche à modéliser les actes de dialogue. Sa taxonomie est une extension de celle de la théorie de l'interprétation dynamique (*Dynamic Interpretation Theory*), originellement basée sur DAMSL (Bunt, 2009). Dans ce schéma, les actes sont interprétés comme des opérations de mise à jour appliquées à l'état informationnel des participants de la conversation. Dans cette perspective, Bunt définit les actes de dialogue comme la conjonction de deux éléments : leur *contenu sémantique* et leur *fonction communicative*. Le contenu sémantique spécifie les objets, propositions et toutes les choses sur lesquelles porte l'acte. La fonction communicative spécifie la manière dont l'acte est supposé impacter l'état informationnel de l'allocutaire. Par exemple, la phrase « *vous avez bientôt fini ?* » peut être interprétée comme une question littérale (le locuteur veut savoir si l'allocutaire est sur le point de finir une tâche), ou comme une expression d'exaspération (le locuteur est gêné par l'activité de l'allocutaire). C'est cette distinction qui doit être capturée par la notion de fonction communicative. Bunt formalise la notion d'acte de dialogue comme suit (Bunt, 2009, p. 13) :

> « *Un acte de dialogue est une unité de description sémantique du comportement communicatif dans le dialogue, précisant comment le comportement est supposé changer l'état informationnel d'un participant qui*

6. Nous avons choisi de conserver les noms originaux en anglais pour ne pas dénaturer la taxonomie.

aurait correctement compris et interprété le comportement. [...] Formellement, un acte de dialogue et un opérateur de mise à jour d'un état informationnel qui s'interprète en appliquant une fonction communicative à un contenu sémantique. [7] »

3.2.1 Dimensions

Dans son panorama des taxonomies d'annotation des actes de dialogue, Popescu-Belis (2005) note que les taxonomies multi-dimensionnelles semblent bénéficier d'une justification théorique au vu de la multiplicité des fonctions que les énoncés peuvent avoir. Néanmoins, le choix des dimensions à intégrer dans un schéma d'annotation devrait lui-même être justifié théoriquement. Il avance six aspects des énoncés qui devraient être pris en compte pour déterminer ces dimensions :

1. Actes de discours : cet aspect correspond à la catégorisation traditionnelle des actes de discours en cinq classes principales, qui bénéficie déjà de solides bases théoriques (Austin, 1975)
2. Tour de parole : les conclusions du champs de l'analyse du dialogue montrent que dans les conversations, des énoncés ont la fonction particulière de gérer les mécanismes de gestion du tour de parole (Shriberg *et al.*, 2004)
3. Paires adjacentes (*adjacency pairs*) : l'analyse de la conversation montre également que des couples d'énoncés sont souvent appareillés relativement à leurs fonctions communicatives, comme par exemple les énoncés de type « réponse »et les énoncés de type « question » (Levinson, 1983; Schegloff & Sacks, 1973)
4. Organisation thématique des conversations : les études en analyse de la conversation ont également démontré que les conversations sont structurées en successions d'épisodes thématiques, au cours desquels les sujets abordés sont amenés à évoluer, et que des énoncés servent à organiser cette évolution (Schegloff & Sacks, 1973)
5. Structure rhétorique : similairement à ce que Thompson & Mann (1987) ont montré pour les discours monologues à travers la théorie RST (*Rhetorical Structure Theory*), des relations discursives rhétoriques peuvent être établies entre les énoncés des conversations (Asher & Lascarides, 2003)
6. Politesse : les fonctions des énoncés en termes de politesse peuvent être formalisées en termes de gestion de la "face", chaque énoncé de ce type pouvant être analysé selon deux axes : d'abord, s'il s'agit de la face du locuteur ou de l'allocutaire, ensuite, si l'interaction vise à *sauver* ou à *menacer* la face (Brown & Levinson, 1983)

Bunt assoit la crédibilité théorique des dimensions qu'il choisit en les basant sur ces six aspects des actes de dialogue. Par ailleurs, il propose de définir précisément ce qu'est un ensemble de dimensions. Dans DIT++, chaque dimension regroupe des fonctions communicatives portant toutes sur un même aspect de ce que peut être la contribution d'un locuteur à la conversation, de manière à ce que : (1) les participants puissent communiquer autour de cet aspect, et (2) cette communication s'opère de façon indépendante des autres aspects, c'est-à-dire qu'un énoncé peut avoir une fonction communicative dans une dimension qui soit totalement indépendante de celles qu'il peut avoir dans d'autres dimensions.

Les dimensions retenues sont les suivantes : (1) *Task/Activity*, pour tout ce qui se rapporte à la tâche qui est l'objet de la conversation ; (2) *Auto-Feedback*, pour les actes signifiant le niveau de compréhension et d'interprétation du locuteur ; (3) *Allo-Feedback*, *idem* pour l'allocutaire ; (4) *Turn Management*, pour les actes portant sur la gestion du tour de parole ; (5) *Time Management*, pour les situations où il est nécessaire de signifier que le locuteur a besoin de plus de temps pour contribuer ou qu'il faut faire une pause ; (6) *Contact Management*, pour les actes qui servent à établir et maintenir la communication ; (7) *Own Communication Management*, pour les actes servant à indiquer que le locuteur prépare ou modifie sa contribution au dialogue ; (8) *Partner Communication Management*, pour les actes effectués par un participant endossant le rôle d'allocutaire, servant à assister son partenaire dans la formulation de sa contribution ; (9) *Discourse Structure Management*, pour les actes servant à structurer thématiquement la conversation ; et (10) *Social Obligations Management*, pour les actes de gestion sociale du dialogue. Les énoncés peuvent avoir au plus une fonction par dimension.

3.2.2 Fonctions

Le schéma d'annotation propose deux types de fonctions communicatives : les fonctions génériques (*general-purpose functions*), qui se retrouvent dans toutes les dimensions (*e.g. propositional question*, *address request*), et les fonctions spécifiques (*dimension-specific functions*), qui ne peuvent être appliquées qu'à une dimension particulière (*e.g. turn grabbing*, *greeting*). La table 2 fournit quelques exemples de fonctions spécifiques pour chaque dimension de la taxonomie.

7. *« A dialogue act is a unit in the semantic description of communicative behaviour in dialogue, specifying how the behaviour is intended to change the information state of a dialogue participant who understands the behaviour correctly. [...] Formally, a dialogue act is an information-state update operator construed by applying a communicative function to a semantic content.* »

Dimension	Exemples de fonction
Task / Activity	*Open Meeting, Appoint, Hire*
Auto-Feedback	*Perception negative, Evaluation positive*
Allo-Feedback	*Interpretation Negative, Evaluation Elicitation*
Turn Management	*Turn Grab, Turn Take, Turn Keep*
Time Management	*Stalling, Pausing*
Contact Management	*Contact Check, Contact Indication*
Own Communication Management	*Self-Correction*
Partner Communication Management	*Completion, Correct Misspeaking*
Discourse Structure Management	*Opening, Topic Introduction*
Social Obligations Management	*Return Greeting, Apology, Thanking*

TABLE 2 – Exemples de fonctions spécifiques

Les fonctions génériques sont elles mêmes réparties en deux catégories principales : les fonctions de transfert d'information et les fonctions de discussion d'action. La première catégorie comporte les fonctions de sollicitation et de procuration d'information. La seconde comporte les fonctions servant à gérer la planification d'actions, correspondant typiquement aux actes commissifs et directifs. La liste complète est fournie en table 3 :

Type	Catégorie	Fonctions
Information	Sollicitatifs	*Propositional Question*, *Set Question*, *Alternatives Question*, *Check Question*, *etc.*, et équivalents indirects (*e.g. Indirect Check Question*)
	Procuratifs	*Inform*, *Agreement*, *Disagreement*, *Correction*, *Propositional Answer*, *Set Answer*, *Confirmation*, *Disconfirmation*, autres variantes dotées de fonctions rhétoriques, comme l'élaboration et la justification, ou de fonctions attitudinales, comme les avertissements
Action	Commissifs	*Offer*, *Promise*, *Address Request*, autres expressifs exprimables via verbes performatifs
	Directifs	*Instruction*, *Address Request*, *Indirect Request*, *Request*, *Suggestion*, autres directifs exprimables via des verbes performatifs, comme les conseils, les encouragements *etc.*

TABLE 3 – Fonctions communicatives génériques de la taxonomie DIT++

3.2.3 Réception et extension

La taxonomie DIT++ a été utilisée pour un ensemble d'applications variées, notamment dans le cadre d'annotation de conversations, de l'analyse théorique du dialogue, de la modélisation des phénomènes conversationnels, et du développement de systèmes de dialogue. Elle peut être étendue pour prendre en compte plus finement certains phénomènes, notamment au travers de la notion de *qualifieurs de fonctions*. Les qualifieurs, introduits par Petukhova & Bunt (2010), sont utilisés en conjonction avec les fonctions communicatives pour décrire l'énoncé plus précisément. Ils proposent une représentation fine du comportement des participants selon différents critères : la modalité, qui spécifie la conviction du locuteur ; la conditionnalité, qui représente la capacité du locuteur à effectuer une action ; la partialité, qui limite la portée de l'énoncé à une partie seulement de l'acte auquel il fait réponse ; et le mode, qui est censé capturer l'attitude et l'état émotionnel du locuteur. Ainsi, le schéma DIT++, facilement extensible et appuyé par un vaste ensemble de travaux antérieurs, a été le socle d'un standard international pour l'annotation dialogique : ISO 24617-2 (Bunt *et al.*, 2012).

4 Classification des énoncés dans les conversations écrites en ligne

L'idée de chercher à classifier les énoncés des conversations écrites en ligne n'est pas nouvelle. En particulier, plusieurs travaux ont cherché à développer des méthodes pour parvenir à classifier automatiquement les énoncés (ou à défaut les phrases, voire les messages entiers) de courriels, de forums et de chats en termes d'actes de dialogue. Dans le domaine du traitement automatique du langage, l'approche supervisée domine les tâches de classification. C'est à cet aspect que

s'intéressent Tavafi *et al.* (2013) : ils étudient les travaux antérieurs du domaine de la modélisation d'actes de dialogue et proposent un panorama des techniques de classification supervisée. Ils concluent que le modèle SVM-HMM, qui prédit les étiquettes séquentiellement, est le plus performant (comparé aux champs conditionnels markoviens (CRFs) et à un SVM multi-classes) sur des corpus de courriels, de forums, de réunions et de conversations téléphoniques. Le travail de Tavafi *et al.* a néanmoins l'inconvénient de ne pas définir ce qu'est un acte de dialogue, et se contente d'utiliser les taxonomies utilisées par les corpus dont ils se servent. C'est problématique parce que ces taxonomies sont généralement très spécifiques à leurs domaines. Dans cette section, nous ne nous intéresserons pas aux médiums retranscrits, mais uniquement aux courriels, aux forums et aux chats.

4.1 Courriels

Un bref examen de n'importe quel corpus de courriels permet de constater que les messages présentent plusieurs caractéristiques qui les rend très différentes des transcriptions tirées de conversations parlées. D'abord, le fait que les messages ne soient pas entièrement constitués de contenu « neuf ». Pour Lampert *et al.* (2009), les courriels peuvent être découpés en trois zones : les zones de locution (*sender zones*), qui contiennent le texte écrit par l'expéditeur ; les zones de contenu cité (*quoted conversation zones*), qui contiennent à la fois le contenu retransmis d'autres conversations et le contenu cité du message auquel l'auteur répond ; et les zones d'encadrement (*boilerplate zones*) qui contiennent le contenu réutilisé sans modification dans plusieurs messages, comme la signature ou les coordonnées de l'auteur. Si l'analyse de la conversation peut profiter de l'extraction d'informations tirées des zones d'encadrement, et si les zones de citations peuvent aider des systèmes à lier les messages entre eux, ce sont surtout les zones de locution qui nous intéressent si l'on cherche à analyser les conversations en termes d'actes de dialogue.

Quelques travaux cherchent à identifier ces actes. Cohen *et al.* (2004) emploient des techniques de classification supervisée pour classer chaque courriel dans une ontologie d'« actes de discours des courriels ». Chaque élément de cette ontologie est composé d'un nom et d'un verbe (*e.g. negotiate meeting* ou *request information*). Cette ontologie ne s'intègre pas nécessairement au paradigme d'analyse introduite par les théories des actes de dialogue que nous avons vu, notamment parce que ce sont les messages et non les énoncés qui sont classés. Lampert *et al.* (2006) tentent, eux, d'annoter les *énoncés* présents dans les courriels. Les classes utilisées sont basées sur les VRM (*Verbal Response Modes*), une taxonomie d'actes de discours qu'ils utilisent pour capturer à la fois l'aspect littéral et pragmatique des énoncés : chaque énoncé est donc classé deux fois, une par aspect, chaque fois avec la même taxonomie. Les résultats encourageants d'une machine à vecteurs de support (SVM) poussent Lampert *et al.* à penser que cette approche est crédible, néanmoins les classes de la taxonomie VRM (*Disclosure*, *Edification*, *Advisement*, *Confirmation*, *Question*, *Acknowledgement*, *Interpretation* et *Reflection*) ne sont pas satisfaisantes pour nos besoins, ni en termes de finesse ni en termes d'exhaustivité. Ainsi, par exemple, elles ne permettent pas de distinguer les fonctions commissives des fonctions expressives, qui tomberaient toutes dans la classe *Disclosure*, ni de faire la différence entre une demande d'action et une demande d'information, ni de capturer les formes de politesse, etc.

4.2 Forums

Qadir & Riloff (2011) font la distinction entre actes de discours et texte expositif (qui apporte de l'information factuelle), et considèrent que les messages de forums diffèrent des documents monologues en ce qu'ils contiennent un mélange des deux, formant ainsi un genre « hybride ». Ils cherchent à classer les phrases de ces messages, d'abord entre actes de discours et phrases expositives, et ensuite entre quatre catégories tirées de la taxonomie de Searle (1976) : les commissifs, les directifs, les expressifs et les représentatifs (ils ignorent les déclaratifs parce qu'ils sont trop rares dans leur corpus). Cette distinction entre texte expositif et actes de discours semble surprenante, puisque d'une certaine façon, en retirant les énoncés expositifs à la catégorie des représentatifs, les auteurs nient leur caractère illocutoire. Néanmoins leur travail rapporte des résultats intéressants, d'autant plus que la construction des traits utilisés par leur classifieur (un SVM) ne demande pas d'analyse linguistique, et se prête bien aux phrases peu grammaticales que l'on peut trouver sur des forums.

Un autre travail qu'il est pertinent de mentionner est celui de Kim *et al.* (2010b). Ils s'intéressent également aux messages des forums, mais cette fois dans une perspective d'aide à la résolution des problèmes. Le problème qu'ils se donnent, annoter automatiquement les conversations s'opérant sur les forums du site CNET, se découpe en deux tâches : (1) la classification des messages, et (2) la classification des *liens* entre les messages. La taxonomie qu'ils adoptent pour identifier la structure du contenu rappelle le concept de paires adjacentes vu en 3.2.1. Les classes sont divisées en deux groupes : *Question* et *Answer* (*e.g. Question-Add*, *Answer-Objection*), plus trois classes isolées qui sont utiles pour parler

de problèmes et de solutions (*Resolution*, *Reproduction* et *Other*). Cette taxonomie paraît très utile pour analyser des conversations porteuses de demandes d'assistance. Néanmoins il ne s'agit pas vraiment d'analyse du dialogue mais plutôt d'analyse de structure de conversation ; le fait de classer les messages et les conversations ignore complètement le concept d'énoncé. L'information apportée se situe à un niveau différent, et on peut parfaitement imaginer effectuer les deux analyses (l'analyse "macro" de Kim *et al.* et une analyse plus proche de l'énoncé) en parallèle.

4.3 Chats

Les chats représentent également un canal fréquemment utilisé pour communiquer autour de la résolution de problèmes, et se distinguent des courriels et forums par leur caractère synchrone. Dans le domaine de l'assistance, Stede & Schlangen (2004) notent qu'il existe des différences fondamentales entre les chats de nature exploratoire et axés vers la recherche d'information (*information-seeking chats*), comme par exemple ceux où un client s'adresse à un agent pour obtenir plus d'informations sur un produit, et ceux orientés autour de l'accomplissement d'une tâche. Ils observent notamment que si les conversations tournées vers l'obtention d'information sont articulées autour d'une série de topiques et sous-topiques liés au domaine, les autres sont plutôt mues par un ensemble de sous-objectifs.

Ha *et al.* (2013) s'intéressent à ces dernières, dans le cadre du développement de systèmes de dialogue. Ils choisissent une approche basée sur la classification supervisée d'actes de dialogue à la fois pour identifier les énoncés de l'utilisateur et pour choisir le type de réponse du système. Bien qu'ils n'évoquent pas directement le terme « chat », ils se basent sur un corpus composé de conversations textuelles entre participants humains communiquant de manière synchrone via une interface web, ce qui revient au même. Néanmoins, ces conversations sont plus des exemples de tutorat que d'assistance à la résolution de problème, ce qui éloigne ce travail de notre objectif. Cet aspect se retrouve dans la taxonomie d'actes de dialogue employée, qui fait la distinction entre ceux du tuteur (*e.g. hint*, *positive feedback*) et de l'élève (*e.g. request for feedback*), certaines classes pouvant être appliquées aux énoncés des deux (*e.g. statement*, *question*). Ils parviennent à prédire à la fois le timing des interventions du tuteur, mais également la type d'intervention. Leurs résultats dépassent l'état-de-l'art en confiant chacune de ces sous-tâches (timing et type) à un classifieur différent.

Ivanovic (2005a) constate que les messages envoyés lors de chats peuvent contenir plus d'un énoncé, et il note donc qu'avant de chercher à classer les actes de dialogue, il faut d'abord identifier leurs frontières textuelles. Il avance que les chats peuvent être découpés en séquences de *tours*, au cours desquels un participant peut envoyer un ou plusieurs messages avant d'attendre une réponse de la part d'un autre participant, chacun de ces messages pouvant contenir un certain nombre d'énoncés. Ses expériences sur un corpus composé de conversations extraites d'un service de support client montre que les techniques statistiques d'apprentissage machine sont très efficaces pour réaliser cette segmentation. Dans un travail séparé, Ivanovic (2005b) cherche à classer ces énoncés dans une taxonomie dérivée de DAMSL. Ils atteint une précision de 80% en combinant un classifieur naïf de Bayes et un modèle de n-grammes d'actes de dialogue. Kim *et al.* reprennent la taxonomie de Ivanovic pour l'appliquer d'abord à un corpus de conversations bipartites (Kim *et al.*, 2010a) puis multipartites (Kim *et al.*, 2012). Dans le premier cas, en utilisant un ensemble de traits lexicaux, structurels et capturant les dépendances entre actes de dialogue, et en adoptant une approche d'apprentissage par champs conditionnels markoviens (CRFs), ils parviennent à des résultats proches de 97%. Si les traits structuraux et les traits de dépendance fonctionnent moins bien pour des conversations multipartites, les CRFs se révèlent de nouveau extrêmement performants avec des résultats de respectivement 97.80% et 99.03% suivant le corpus.

5 Conclusion et perspectives de recherche

Nous avons vu que la recherche théorique en matière d'analyse des conversations en termes d'actes de dialogue est extrêmement fournie, les origines de cette tradition remontant aux années 60 avec l'introduction du concept d'acte de discours (Austin, 1975; Searle, 1969). Néanmoins, on remarque que les apports théoriques plus récents adaptés aux conversations fonctionnelles reposent souvent sur l'étude du même échantillon de dialogues, le corpus TRAINS. C'est le cas de Traum & Hinkelman (1992); Poesio & Traum (1997); Core & Allen (1997); Bunt (2009). Les dialogues contenus dans ce corpus correspondent à un type bien particulier de conversations fonctionnelles, avec leurs spécificités (comme, par exemple, le fait que tous soient bi-agents plutôt que poly-agents). Ces dialogues ont aussi un focus important sur la planification temporelle et géographique des tâches, ce qui n'est pas nécessairement une propriété commune à toutes les situations de résolution de problème, en particulier celles rencontrées sur le plate-formes d'entraide. Toujours est-il que ces fondements théoriques ont pu donner naissance à des schémas d'annotation bien établis et à l'utilité démontrée.

Nous avons également montré que l'idée d'appliquer ce type d'analyse aux conversations en ligne, et en particulier aux conversations construites autour de la réalisation d'une tâche, a déjà été concrétisée, y compris dans certains cas pour l'aide à la résolution des problèmes. Cependant, on ne peut pas dire que les travaux s'intéressant à ces applications soient particulièrement homogènes en termes de taxonomie, leurs auteurs préférant souvent choisir des classes appropriées à leurs domaines que s'appuyer sur des fondements théoriques généraux. C'est un problème puisque, d'une part, les corpus utilisés pour leurs tâches sont donc annotés de façon *ad hoc*, ce qui limite leur exploitabilité pour d'autres applications, et, d'autre part, parce que cela rend les méthodes employées difficiles à comparer et à généraliser.

Une première perspective de recherche intéressante qui nous apparaît est celle du développement d'une théorie de la conversation asynchrone riche. Si les messages instantanés rentrent assez bien dans le cadre de ce que peut être un énoncé tel que résumé par Popescu-Belis (2005), ce n'est pas nécessairement le cas des courriels et des messages postés sur les forums de discussion, pour lesquels un certain nombre de questions peuvent se poser. Pouvons nous, sur ces médiums, parler de gestion de tour de parole, quand tout est géré techniquement par la plate-forme ? Comment se passe la gestion des connaissances communes ? *Quid* des comportements non-verbaux écrits (citations, liens vers de ressources externes, inclusion de contenu multimédia, émoticônes, points d'exclamation multiples *etc.*) ? Une autre idée qui mérite investigation est de savoir s'il est possible de spécialiser des schémas d'annotation comme DAMSL ou DIT++ pour le genre des conversations écrites en ligne orientées vers la résolution de problème, et si la taxonomie qui en résulterait peut être employée pour une tâche de classification automatique des énoncés. Enfin, nous prévoyons également de définir formellement ce qu'est un problème et ce qu'est une solution, et comment ces objets apparaissent et évoluent au travers des conversations. Ce travail de formalisation aura pour but de permettre, à terme, leur modélisation automatique.

Remerciements

Ce travail, qui s'inscrit dans le cadre du projet ODISAE (www.odisae.com), a bénéficié du soutien du fond unique interministériel (FUI) 17. Nous remercions nos relecteurs pour leurs commentaires constructifs.

Références

ASHER N. & LASCARIDES A. (2003). *Logics of Conversation*. Cambridge University Press.

AUSTIN J. L. (1975). *How to Do Things With Words*. Oxford University Press, second edition.

BERNETT H. (2000). E-Commerce, Customer Service, and the Web-Enabled Call Center.

BROWN P. & LEVINSON S. C. (1983). *Politeness: Some Universals in Language Use*. Cambridge University Press.

BRUNER J. S. (1975). From Communication to Language - A Psychological Perspective. *Cognition*, **3**(3), 255–287.

BUNT H. (2006). Dimensions in Dialogue Act Annotation. In *Proceedings of the 2006 International Conference on Language Resources and Evaluation (LREC 2006)*, p. 919–924, Genoa, Italy.

BUNT H. (2009). The DIT++ taxonomy for functional dialogue markup. In *Proceedings of the AAMAS 2009 Workshop "Towards a Standard Markup Language for Embodied Dialogue Acts" (EDAML 2009)*, p. 13–24, Budapest, Hungary.

BUNT H., ALEXANDERSSON J., CHOE J.-W., FANG A. C., HASIDA K., PETUKHOVA V., POPESCU-BELIS A. & TRAUM D. R. (2012). ISO 24617-2: A semantically-based standard for dialogue annotation. In *Proceedings of the 2012 International Conference on Language Resources and Evaluation (LREC 2012)*, p. 430–437, Istanbul, Turkey.

COHEN W. W., CARVALHO V. R. & MITCHELL T. M. (2004). Learning to Classify Email into "Speech Acts". In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, p. 309–316, Barcelona, Spain.

CORE M. & ALLEN J. (1997). Coding Dialogs with the DAMSL Annotation Scheme. In *Working Notes of AAAI Fall Symposium on Communicative Action in Humans and Machines*, p. 28–35, Boston, MA, USA.

FILLMORE C. J. (1971). Some Problems for Case Grammar. *Monograph Series on Languages and Linguistics*, **24**, 35–56.

GROSS D., ALLEN J. & TRAUM D. (1993). *The TRAINS 91 Dialogues*. Rapport interne, University of Rochester.

HA E. Y., MITCHELL C. M., BOYER K. E. & LESTER J. C. (2013). Learning Dialogue Management Models for Task-Oriented Dialogue with Parallel Dialogue and Task Streams. In *Proceedings of the 14th SIGdial Workshop on Discourse and Dialogue (SIGdial 2013)*, p. 204–2013, Metz, France.

IVANOVIC E. (2005a). Automatic utterance segmentation in instant messaging dialogue. In *Proceedings of the Australasian Language Technology Workshop 2005 (ALTW 2005)*, p. 241–249, Sydney, Australia.

IVANOVIC E. (2005b). Dialogue act tagging for instant messaging chat sessions. In *Proceedings of the ACL Student Research Workshop (SRW 2005)*, p. 79–84, Ann Arbor, MI, USA.

JAKOBOVITS L. A. & GORDON B. (1974). *The Context of Foreign Language Teaching*. Newbury House Publishers.

KIM N. S., CAVEDON L. & BALDWIN T. (2010a). Classifying dialogue acts in one-on-one live chats. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, p. 862–871, Cambridge, MA, USA.

KIM N. S., CAVEDON L. & BALDWIN T. (2012). Classifying dialogue acts in multi-party live chats. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation (PACLIC 2012)*, p. 463–472, Bali, Indonesia.

KIM S. N., WANG L. & BALDWIN T. (2010b). Tagging and Linking Web Forum Posts. In *Proceedings of the 14th Conference on Computational Natural Language Learning (CoNLL 2010)*, p. 192–202, Stroudsburg, PA, USA.

LAMPERT A., DALE R. & PARIS C. (2006). Classifying Speech Acts Using Verbal Response Modes. In *Proceedings of the 2006 Australasian Language Technology Workshop (ALTW 2006)*, p. 34–41, Sydney, Australia.

LAMPERT A., DALE R. & PARIS C. (2009). Segmenting Email Message Text into Zones. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, p. 919–928, Singapore.

LEECH G. & WEISSER M. (2003). Generic speech act annotation for task-oriented dialogues. In *Proceedings of the 2003 Corpus Linguistics Conference (CL 2003)*, p. 441–446, Lancaster, UK.

LEVINSON S. C. (1983). *Pragmatics*. Cambridge University Press.

MORELLI R., BRONZINO J. & GOETHE J. (1991). A computational speech-act model of human-computer conversations. In *Proceedings of the 17th IEEE Northeast Bioengineering Conference*, p. 263–264, Hartford, CT, USA.

OHMANN R. (1971). Speech Acts and the Definition of Literature. *Philosophy & Rhetoric*, **4**(1), 1–19.

PETUKHOVA V. & BUNT H. (2010). Introducing communicative function qualifiers. In *Proceedings of the 2nd International Conference on Global Interoperability for Language Resources (ICGL 2010)*, p. 123–133, Hong Kong, China.

POESIO M. & TRAUM D. R. (1997). Conversational Actions and Discourse Situations. *Computational Intelligence*, **13**(3), 309–347.

POPESCU-BELIS A. (2005). Dialogue Acts: One or More Dimensions? *ISSCO Working Papers*, (62).

QADIR A. & RILOFF E. (2011). Classifying Sentences As Speech Acts in Message Board Posts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, p. 748–758, Edinburgh, UK.

SADOCK J. M. (1974). *Toward a Linguistic Theory of Speech Acts*. Academic Press.

SCHEGLOFF E. A. & SACKS H. (1973). Opening Up Closings. *Semiotica*, **8**(4), 289–327.

SEARLE J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

SEARLE J. R. (1976). *A Taxonomy of Illocutionary Acts*. Linguistic Agency University of Trier.

SHRIBERG E., DHILLON R., BHAGAT S., ANG J. & CARVEY H. (2004). The ICSI Meeting Recorder Dialog Act (MRDA) Corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue (SIGdial 2004)*, p. 97–100.

STEDE M. & SCHLANGEN D. (2004). Information-Seeking Chat: Dialogues Driven by Topic Structure. In *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2004)*, p. 117–124, Barcelona, Spain.

TAVAFI M., MEHDAD Y., JOTY S., CARENINI G. & NG R. (2013). Dialogue Act Recognition in Synchronous and Asynchronous Conversations. Master's thesis, University of British Columbia.

THOMPSON S. A. & MANN W. C. (1987). Rhetorical Structure Theory: A Framework for the Analysis of Texts. *IPrA Papers in Pragmatics*, **1**(1), 79–105.

TRAUM D. R. & HINKELMAN E. A. (1992). Conversation Acts in Task-Oriented Spoken Dialogue. *Computational Intelligence*, **8**(3), 575–599.

TWITCHELL D. P., ADKINS M., NUNAMAKER J. F. & BURGOON J. K. (2004). Using Speech Act Theory to Model Conversations for Automated Classification and Retrieval. In *Proceedings of the 9th International Working Conference on the Language-Action Perspective on Communication Modelling (LAP 2004)*, p. 121–129, New Brunswick, NJ, USA.

VANDERVEKEN D. (1992). La théorie des actes de discours et l'analyse de la conversation. *Cahiers de Linguistique Française*, p. 9–62.