Traitements formels et sémantiques des échanges et des documents textuels liés à des activités collaboratives

Ruslan Kalitvianski

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR ÈS SCIENCES DÉLIVRÉ PAR LA COMMUNAUTE UNIVERSITE GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

Ruslan KALITVIANSKI

Thèse dirigée par **Christian BOITET, Professeur émérite, Université Grenoble Alpes,** et
codirigée par **Valérie BELLYNCK, Maître de conférences, Grenoble INP**

préparée au sein du **GETALP-LIG (CNRS-INPG-UGA)**
dans l'**École Doctorale "Mathématiques, Sciences et Technologies de l'Information, Informatique"**

Traitements formels et sémantiques des échanges et des documents textuels liés à des activités collaboratives

Thèse soutenue publiquement le **20 mars 2018**,
devant le jury composé de :

Mme Marie-Christine ROUSSET
Professeur, Université Grenoble Alpes, Président

Mme Adeline NAZARENKO
Professeur, Université Paris 13, Rapporteur

Mme Anne VILNAT
Professeur, Université Paris-Sud, Rapporteur

Mme Violaine PRINCE
Professeur, Université de Montpellier, Examinateur

Mme Frédérique SEGOND
Professeur associé, INaLCO, Examinateur

M. Emmanuel MORIN
Professeur, Université de Nantes, Examinateur

Mme Valérie BELLYNCK
Maître de Conférences, Grenoble INP, Codirecteur

M. Christian BOITET
Professeur émérite, Université Grenoble Alpes, Directeur

Résumé

Cette thèse s'inscrit dans la problématique de l'extraction de sens à partir de textes et flux textuels, produits dans notre cas lors de processus collaboratifs. Plus précisément, nous nous intéressons aux courriels de travail et aux documents textuels objets de collaboration, avec une première application aux documents éducatifs. La motivation de cet intérêt est d'aider les utilisateurs à accéder plus rapidement aux informations utiles ; nous cherchons donc à les repérer dans les textes. Ainsi, nous nous intéressons aux tâches dans les courriels, et aux fragments de documents éducatifs qui concernent les thèmes de leurs intérêts. Deux corpus, un de courriels et un de documents éducatifs, principalement en français, ont été constitués. Cela était indispensable, car il n'y a pratiquement pas de travaux antérieurs sur ce type de données en français.

Notre première contribution théorique est une modélisation générique de la structure de ces données. Nous l'utilisons pour spécifier le traitement formel des documents, prérequis au traitement sémantique. Nous démontrons la difficulté du problème de segmentation, normalisation et structuration de documents en différents formats source, et présentons l'outil SEGNORM, première contribution logicielle de cette thèse. SEGNORM segmente et normalise les documents (en texte brut ou balisé), récursivement et en unités de taille paramétrable. Dans le cas des courriels, il segmente les messages contenant des messages cités en messages individuels, en conservant l'information du chaînage entre les fragments entremêlés. Il analyse également les métadonnées des messages pour reconstruire les fils de discussions, et retrouve dans les citations les messages dont on ne possède pas le fichier source.

Nous abordons ensuite le traitement sémantique de ces documents. Nous proposons une modélisation de la notion de tâche, puis décrivons l'annotation d'un corpus de plusieurs centaines de messages issus du contexte professionnel de VISEO et du GETALP. Nous présentons alors la deuxième contribution logicielle de cette thèse, un outil de repérage de tâches et d'extraction de leurs attributs (contraintes temporelles, assignataires, etc.). Cet outil, basé sur une combinaison d'une approche experte et d'apprentissage automatique, est évalué selon des critères classiques de précision, rappel et F-mesure, ainsi que selon la qualité d'usage.

Enfin, nous présentons nos travaux sur la plate-forme MACAU-CHAMILO, troisième contribution logicielle, qui aide à l'apprentissage par (1) structuration de documents pédagogiques selon deux ontologies (forme et contenu), (2) accès multilingue à du contenu initialement monolingue. Il s'agit donc de nouveau de structuration selon les deux axes, forme et sens.

(1) L'ontologie des formes permet d'annoter les fragments des documents par des concepts comme *théorème, preuve, exemple*, par des niveaux de difficulté et d'abstraction, et par des relations comme `élaboration_de`, `illustration_de`. L'ontologie de domaine modélise les objets formels de l'informatique, et plus précisément les notions de complexité calculatoire. Cela permet de suggérer aux utilisateurs des fragments utiles pour la compréhension de notions d'informatique perçues comme abstraites ou difficiles.

(2) L'aspect relatif à l'accès multilingue a été motivé par le constat que nos universités accueillent un grand nombre d'étudiants étrangers, qui ont souvent du mal à comprendre nos cours à cause de la barrière linguistique. Nous avons proposé une approche pour multilingualiser du contenu pédagogique avec l'aide d'étudiants étrangers, par *post-édition* en ligne de pré-traductions automatiques, puis, si besoin, amélioration incrémentale de ces post-éditions. (Nos expériences ont montré que des versions multilingues de documents peuvent être produites rapidement et sans coût.) Ce travail a abouti à un corpus de plus de 500 pages standard (250 mots/page) de contenu pédagogique post-édité vers le chinois.

Summary

This thesis is part of the problematics of the extraction of meaning from texts and textual flows, produced in our case during collaborative processes. More specifically, we are interested in work-related emails and collaborative textual documents, with a first application to educational documents. The motivation for this interest is to help users gain access to useful information more quickly; we hence seek to locate them in the texts. Thus, we are interested in the tasks referred to in the emails, and to the fragments of educational documents which concern the themes of their interests. Two corpora, one of e-mails and one of educational documents, mainly in French, have been created. This was essential because there is virtually no previous work on this type of data in French.

Our first theoretical contribution is a generic modeling of the structure of these data. We use it to specify the formal processing of documents, a prerequisite for semantic processing. We demonstrate the difficulty of the problem of segmentation, standardization and structuring of documents in different source formats, and present the SEGNORM tool, the first software contribution of this thesis. SEGNORM segments and normalizes documents (in plain or tagged text), recursively and in units of configurable size. In the case of emails, it segments the messages containing quotations of messages into individual messages, thereby keeping the information about the chaining between the intertwined fragments. It also analyzes the metadata of the messages to reconstruct the threads of discussions, and retrieves in the quotations the messages of which one does not have the source file.

We then discuss the semantic processing of these documents. We propose a modeling of the notion of task, then describe the annotation of a corpus of several hundred messages originating from the professional context of VISEO and GETALP. We then present the second software contribution of this thesis: the tool for locating tasks and extracting their attributes (temporal constraints, assignees, etc.). This tool, based on a combination of an expert approach and machine learning, is evaluated according to classic criteria of accuracy, recall and F-measure, as well as according to usage quality.

Finally, we present our work on the MACAU-CHAMILO platform, third software contribution, which helps learning by (1) structuring of educational documents according to two ontologies (form and content), (2) multilingual access to content initially monolingual. This is therefore again about structuring along the two axes, form and meaning.

(1) The ontology of forms makes it possible to annotate the fragments of documents by concepts such as *theorem*, *proof*, *example*, by levels of difficulty and abstraction, and by relations such as `elaboration_of`, `illustration_of`… The domain ontology models the formal objects of informatics, and more precisely the notions of computational complexity. This makes it possible to suggest to the users fragments useful for understanding notions of informatics perceived as abstract or difficult.

(2) The aspect related to multilingual access has been motivated by the observation that our universities welcome a large number of foreign students, who often have difficulty understanding our courses because of the language barrier. We proposed an approach to multilingualize educational content with the help of foreign students, by online *post-editing* of automatic pre-translations, and, if necessary, incremental improvement of these post-editions. (Our experiments have shown that multilingual versions of documents can be produced quickly and without cost.) This work resulted in a corpus of more than 500 standard pages (250 words/page) of post-edited educational content into Chinese.

Резюме

Эта диссертация является частью проблематики извлечения смысла из текстов и текстовых потоков, созданных в нашем случае в ходе совместной работы. В частности, мы заинтересованы в обработке электронных писем и текстовых документов, являющихся предметом совместной работы, с первой применемием к образовательным документам. Этот интерес мотивируется желанием помочь пользователям быстрее получать доступ к полезной информации; следовательно, мы стремимся обнаружать их в текстах. Таким образом, нас интересуют задачи в электронных письмах и фрагменты образовательных документов, которые затрагивают темы их интересов. Были созданы два корпуса, один из электронных писем и один из образовательных документов, в основном на французском языке. Это было необходимо, так как работ по этому типу данных на французском языке практически нет.

Наш первый теоретический вклад — это общее моделирование структуры этих данных. Мы используем его для определения формальной обработки документов, необходимого этапа для семантической обработки. Мы доказываем сложность проблемы сегментации, стандартизации и структурирования документов в разных исходных форматах и представляем утилиту SEGNORM, которая является первым программным вкладом этой диссертации. SEGNORM сегментирует и нормализует документы (в обычном тексте или размеченные), рекурсивно и в единицах произвольного размера. В случае электронной почты, он сегментирует сообщения, которые сами содержат сообщения, в отдельные сообщения, не теряя информацию о сцеплении между переплетенными фрагментами. Он также анализирует метаданные сообщений для восстановления потоков обсуждений и находит в цитатах сообщения, чьи исходные файлы мы не имеем.

Затем мы касаемся семантической обработки этих документов. Мы предлагаем моделирование понятия задачи, а затем описываем аннотацию корпуса из нескольких сотен сообщений из профессионального контекста VISEO и GETALP. Затем мы представляем второй программный вклад этой диссертации: инструмент для нахождения задач и извлечения их атрибутов (ограничения по времени, исполнители и т. д.). Этот инструмент, основанный на сочетании экспертного подхода и машинного обучения, оценивается в соответствии с классическими критериями точности, отзыва и F-измерения, а также в соответствии с качеством использования.

Наконец, мы представляем нашу работу на платформе MACAU-CHAMILO, которая является третьим программным вкладом — он помогает обучению : (1) структурированием образовательных документов в соответствии с двумя онтологиями (форма и содержание), (2) многоязычным доступом к первоначально одноязычному контенту. Поэтому речь снова идёт о структурировании по двум осям, форме и смыслу.

(1) Онтология форм позволяет аннотировать фрагменты документов понятиями, такими как *теорема, доказательство, пример*, уровнями сложности и абстракции, и такими отношениями, как **уточнение**, **иллюстрирование**... Онтология области моделирует формальные объекты информатики, а точнее, понятия вычислительной сложности. Это позволяет предлагать фрагменты пользователям, полезные для понимания понятий информатики, воспринимаемые как абстрактные или сложные.

(2) Аспект, связанный с многоязычным доступом, был мотивирован наблюдением, что наши университеты принимают большое количество иностранных студентов, которые часто испытывают трудности с пониманием наших курсов из-за языкового барьера. Мы предложили подход к многоязычению образовательного контента с помощью иностранных студентов, путём *пост-редактирования* онлайновых автоматических предварительных переводов и, при необходимости, постепенного улучшения этих пост-редактирований . (Наши эксперименты показали, что многоязычные версии документов могут быть получены быстро и без каких-либо затрат.) Эта работа привела к созданию корпуса из более 500 стандартных страниц (250 слов / страниц) постредактированного образовательного контента на китайском языке.

Remerciements

Мама, папа, Зарема, я вас люблю.

Elina, merci pour ton infaillible soutien.

Bien que j'aie baigné durant ces années dans le traitement de la langue naturelle, je peine à trouver les mots qui sauraient exprimer l'immensité de mon admiration et de ma gratitude envers les personnes qui m'ont encadré durant ce travail, à savoir Frédérique, Valérie, et Christian.

Je suis très reconnaissant aux membres de mon jury, à savoir les professeurs Adeline Nazarenko, Anne Vilnat, Emmanuel Morin, Violaine Prince et Marie-Christine Rousset, d'avoir accepté d'évaluer ce travail. Cela me rend très humble.

Mon aventure Viseo étant désormais arrivée à sa fin, je voudrais exprimer un remerciement tout particulier à Franck Priore, pour l'opportunité qu'il m'a offerte de travailler sur le projet SYNAPS. L'expérience a été des plus excitantes et instructives. Je n'oublierai pas les nombreuses discussions insolites, animées et stimulantes que nous avons eues les midis à la Brasserie, nous trois avec Pierre, grand-maître de Java et pilier porteur de l'architecture de SYNAPS, que je salue également.

Je voudrais aussi saluer Sébastien, auteur, et sorcier de JavaScript, qui a rejoint l'équipe SYNAPS le même jour que moi. (J'ai toujours la « petite Xbox » fabriquée par tes soins). Je n'oublie pas non plus mes collègues et amis de l'équipe R&D de Viseo, dont Cédric, Namrata, Nadia, Kévin, Pierre-Alain et Parantapa.

Au GETALP, mon équipe d'accueil, j'ai eu la chance de côtoyer des personnes remarquables. Mes chaleureuses salutations vont à Mutsuko, Jean-Claude, et Jean-Philippe, ainsi qu'à mes collègues, désormais docteurs : Ritesh, Claire, Lingxiao, Ying, et Andon.

Mes amis de longue date, Nadir et Denis, je vous salue et vous remercie cordialement aussi.

J'oublie nécessairement beaucoup d'autres personnes envers lesquelles j'ai une dette de gratitude. Sans doute me reviendront-elles dès que le dernier exemplaire de ce manuscrit aura été imprimé et relié.

Table des matières

Table des figures

Table des tableaux

Glossaire

Sigle / Terme	Développement en français / explication	Développement en anglais
Action item[1]	Souvent traduit incorrectement comme « élément d'action », ce terme, issu du domaine de la gestion, désigne un « évenement, tâche, activité ou action à effectuer ; une unité discrète d'action pouvant être faite par une seule personne. » Cela correspond souvent à un élément d'une *todo list*.	
Flux XML	Représentation dans Systran des formes successives d'un texte à traduire lors du processus de traduction automatique	
GT	Google Traduction	Google Translate
iMAG	Interactive Multilingual Access Gateway	Interactive Multilingual Access Gateway
Indice de Rand	Une mesure du pourcentage de décisions correctes par un outil de classification (ou de partitionnement).	Rand Index
MACAU	Multilinguïsation et Appropriation Contributive À l'Université (projet PedagoTICE mené depuis 2013 à l'UJF/UGA)	Multilingual Access and Contributive Appropriation for Universities
MaxEnt	Outil d'apprentissage basé sur la maximisation de l'entropie	Maximum Entropy
MT	Mémoire de Traductions (MT)	Translation Memory (TM)
Normalisation	Divers traitements en vue de la soumission à un système de TALN (tokénisation, décapitalisation, remplacement des balises et des hors-texte par des occurrences spéciales, etc.)	
PE	Post-édition	Post-Editing
SECTra	Système d'Exploitation de Corpus de Traductions	
Segmentation	Analyse d'un document en une hiérarchie de sections, jusqu'au paragraphe, puis en un graphe sans cycle de segments (phrases ou titres).	
SVM	Séparateur à vaste marge (outil d'apprentissage pour la classification)	Support Vector Machine
Systran	Firme et système de TA éponyme	**Sys**tem of **Tran**slation
TA	Traduction Automatique	Machine Translation (MT)
THAM	Traduction Humaine Aidée par la Machine	Machine-Aided Human Translation (or CAT, Computer-Aided Translation)

[1] https://en.wikipedia.org/wiki/Action_item

Introduction

La genèse de cette thèse résulte d'une histoire personnelle assez atypique. Arrivé en France à 13 ans en n'ayant pas tout à fait achevé mes études secondaires en Russie, j'ai réalisé l'importance d'avoir une compétence la plus haute possible dans la langue d'enseignement, et fait le maximum pour y arriver. Dix ans plus tard, étant dans une classe de L3-info où il y avait un groupe de Chinois, et faisant un stage sur le problème de la segmentation et de la normalisation des pages Web dans le projet IMAG, visant l'accès multilingue de qualité par traduction automatique (TA) puis post-édition (PE), j'ai proposé de construire avec cet outil un environnement d'aide à la compréhension du français, à l'intention des étudiants étrangers. Avec mon tuteur, devenu depuis mon directeur de thèse, nous avons monté ce projet, dit MACAU[2], et avons bénéficié de l'aide du pôle PEDAGOTICE de l'UJF.

Mon autre centre d'intérêt lors de mon entrée à l'université était les sciences, leur transmission, et leur vulgarisation. Après un début de parcours en communication, je me suis réorienté vers l'informatique, et me suis particulièrement intéressé à ses aspect fondamentaux (lien avec la logique mathématique, algorithmique, calculabilité, décidabilité, complexité, langages, grammaires, automates…). En TALN, je suis intéressé aussi bien par des problèmes de traitement de la forme et de la structure (structuration et segmentation des documents, analyse morphosyntaxique, traduction automatisée) que par ce qu'on appelle le « Web 3.0 », i.e. l'extraction de contenu, la compréhension explicite par interprétation dans une ontologie, et l'inférence.

Le sujet de thèse CIFRE qui a été mis au point avec l'entreprise VISEO après mon M2R était centré sur l'aide ou les aides qu'on pourrait mettre à disposition des utilisateurs du système SYNAPS, en cours de construction. Ce système très intéressant et novateur visait à supporter les activités collaboratives en général, et pas seulement les projets professionnels structurés. Il avait une interface similaire à celle de Trello, et un composant nouveau, la « carte sémantique », dans laquelle on prévoyait de représenter aussi bien les échanges entre participants d'une activité que les relations entre ces échanges et les composants de cette activité (tâches, participants, contraintes…). Mon sujet initial était : *Vers l'interprétation sémantique et pragmatique de l'évolution de la carte cognitive et des comportements des utilisateurs d'un logiciel basé sur une "carte cognitive"*. Il était aussi prévu que le projet MACAU soit utilisé comme base expérimentale, à cause de son caractère ouvert, et, en ce qui me concerne, de son caractère multilingue.

Malheureusement, SYNAPS a été abandonné au milieu de ma thèse (mai 2016) pour des raisons économiques. VISEO m'a alors proposé d'infléchir mon sujet en changeant le contexte tout en gardant le but initial d'aider les utilisateurs. Le contexte est devenu celui des échanges par courriel entre participants d'un projet, avec application au projet REUS[3] de gestion des réunions de travail dans des projets collaboratifs. Le sujet est alors devenu : *Traitement des textes et des flux textuels dans des activités collaboratives*. En travaillant sur plusieurs corpus réels de courriels[4], il est apparu qu'il y avait des différences notables entre la structure d'un ensemble d'échanges par courriel et celle d'un ensemble de documents liés à ces échanges, par exemple, la description d'un projet, avec ses lots, tâches et sous-tâches, ou encore la partie d'un cours de complexité calculatoire concernant les problèmes P, NP, et NP-complets. C'est pourquoi le titre

[2] Multilingual Access and Contributive Appropriation for Universities.
[3] Projet FUI, http://reus-project.com/
[4] Le fameux corpus ENRON, en anglais et « nettoyé », un corpus de Viseo, un corpus du GETALP, et les courriels de MACAU.

final de cette thèse est : *Traitements formels et sémantiques des échanges et des documents textuels liés à des activités collaboratives*.

Cette thèse s'inscrit donc dans la problématique de l'extraction de sens à partir de textes et de flux textuels, produits dans notre cas lors de processus collaboratifs. Plus précisément, nous nous intéressons aux courriels de travail et aux documents textuels objets de collaboration, avec une première application aux documents éducatifs. La motivation de cet intérêt est d'aider les utilisateurs à accéder plus rapidement aux informations utiles ; nous cherchons donc à les repérer dans les textes. Ainsi, nous nous intéressons aux tâches dans les courriels, et aux fragments de documents éducatifs qui concernent les thèmes de leurs intérêts. Par son aspect « traitement des échanges », cette recherche se place dans le cadre des recherches sur les « media sociaux ».

À cause de notre contexte, nous avons travaillé sur un seul type d'échanges, les *courriels*. Nous n'avons donc pas travaillé sur les tweets, les blogs, les microblogs, ni les tchats, sur lesquels il y a déjà de nombreux travaux[5]. Cependant, il s'agit le plus souvent de flux textuels plus simples que les échanges de courriels, et, par exemple, ce que nous avons fait pourrait être simplifié et adapté au cas des tweets et des blogs.

Nous avons aussi travaillé sur des *documents* souvent complexes. Cela nous a amené à progresser dans la modélisation des corpus en général. Aux notions maintenant classiques de macrostructure, microstructure et mésostructure, qui permettent de décrire, entre autres, un corpus de dialogues oraux interprétés puis transcrits (Wang L.X., 2015), nous ajoutons celles de *sous-document* et *sur-document*, pour rendre compte de la présence de citations de fragments de courriels dans d'autre courriels.

En ce qui concerne les traitements sémantiques, notre travail renforce l'idée qu'il faut utiliser comme « référentiels » non pas des ensembles d'énoncés en langue naturelle (comme des exigences ou des spécifications), mais plutôt des ontologies de domaine, relatives à différents aspects. Par exemple, nous utilisons une ontologie pour la forme et la structure des documents et sous-documents pédagogiques, et aussi une ontologie relative à leur contenu (ici, la théorie de la complexité calculatoire).

Dans le chapitre I, nous présentons le contexte scientifique général de notre domaine de recherche, en précisant le genre d'aide qu'il semble utile de fournir aux participants d'une activité collaborative, et en faisant ressortir les difficultés et les problèmes que cela pose.

Dans le chapitre II, nous abordons les traitements de la forme et de la structure, avec deux volets, (1) le traitement de *documents* en vue de la TA et de la PE, et (2) le traitement d'échanges par courriel. Pour cela, deux corpus, un de courriels et un de documents éducatifs, principalement en français, ont été constitués. Cela était indispensable, car il n'y a pratiquement pas de travaux antérieurs sur ce type de données en français. Nous présentons notre modélisation, et l'outil SEGNORM que nous développons depuis plusieurs années, non seulement pour segmenter des documents (dont des courriels), mais aussi pour les normaliser en vue de traitements ultérieurs.

Le chapitre III est consacré au repérage et au traitement de courriels relatifs à des tâches. Nous précisons le problème à résoudre et l'état de l'art[6], puis notre modélisation des informations à

[5] Voir par exemple le livre de (Farzindar & Inkpen. 2015), ou la thèse récente de (Shah, 2017) sur l'aide à la compréhension de tweets en langue étrangère.

[6] Nous avons découvert assez tard la remarquable thèse d'Andrew Lampert (2014), qui a surtout travaillé sur le corpus ENRON, mais nous inscrivons tout à fait en complément et en suite de ses travaux, pour ce qui est de la segmentation et de la normalisation d'une part, et des traitements sémantiques d'autre part.

attacher aux segments repérés. Nous décrivons alors la création de corpus de courriels annotés, principalement en français (il n'en existait pas). Enfin, nous présentons nos expériences de repérage. Nous comparons plusieurs méthodes, expertes et empiriques (apprentissage supervisé dans notre cas).

Le chapitre IV est consacré au traitement de documents complexes en contexte d'apprentissage actif. Nous décrivons d'abord le projet MACAU et son utilisation dans le contexte réel de l'apprentissage d'outils formels de l'informatique (OFI) en L3-info et M1-info par des étudiants chinois de l'UJF (intégrée depuis 2016 à l'UGA). Nous donnons les deux ontologies que nous avons développées pour la structure (à granularité fine, celle des sous-documents) et pour le contenu (ici, la théorie de la complexité calculatoire). Nous finissons en détaillant le traitement des documents textuels liés à MACAU-CHAMILO, l'application à la traduction en chinois (par TA suivie de PE par des étudiants chinois) d'environ 500 pages standard de cours, TD, exercices, examens et corrigés non seulement en complexité, mais aussi en logique, langages, grammaires, automates, décidabilité et calculabilité.

Nous terminerons par les traditionnelles « conclusions et perspectives ». Dans le texte, nous utilisons une fonte particulière pour les citations, ainsi que le « nous » traditionnel de modestie[7]. Nous reviendrons cependant au « je » à la fin, quand il s'agira de nos perspectives personnelles.

[7] Avec accord « sémantique », donc au singulier.
[8] Voir http://lig-membres.imag.fr/wang/downloads/index.html

Chapitre I Contexte scientifique

Introduction

Nous nous intéressons à la gestion et au support d'activités collaboratives, et plus particulièrement à la gestion des textes et des flux textuels liés à ces activités. Dans ce chapitre, nous présentons un bref panorama du vaste champ de la collaboration, des outils et des pratiques mises en œuvre pour coordonner les activités.

Nous commençons par présenter les activités collaboratives dans deux cadres : les projets d'entreprise et les collaborations hors du cadre commercial. Ensuite, nous proposons quatre types d'aide envisageables, en temps réel ou en différé. Enfin, nous mettons en évidence les obstacles et difficultés scientifiques qui y sont associés, et que l'on doit traiter afin d'aboutir à des aides utiles.

I.1 Gestion et support de différents types d'activités collaboratives

Nous distinguerons les activités menées par des entreprises ou autres organismes « professionnels » et les activités hors du cadre commercial.

I.1.1 Activités collaboratives dans le cadre de projets d'entreprise

Dans leur grande majorité, les projets d'entreprise sont collaboratifs au sens large, dans le sens où plusieurs personnes contribuent à l'atteinte d'un même but. Certaines collaborations sont internes à l'entreprise, d'autres se déroulent entre l'entreprise et d'autres acteurs. Il y a souvent des collaborations et échanges internes entre individus formant des équipes et des collaborations entre équipes. Nous examinons ici de plus près la variété des situations, ainsi que les moyens mis en œuvre pour la coordination et la gestion des collaborations.

I.1.1.1 Variété des situations

Les activités collaboratives dans les entreprises prennent différentes formes.

On trouve des collaborations complètement internes à l'entreprise, comme dans le cas de certaines phases de développement de logiciel. On trouve des activités qui impliquent des acteurs externes, par exemple l'évaluation de produits par les clients ou les phases de béta-tests. Il y a aussi des cas mixtes, comme le cas du développement d'une fonctionnalité en interaction avec un client.

Étudier la variété des situations nous intéresse, car les solutions mises en œuvre pour les gérer peuvent différer. Un outil peut être mis en place pour que les clients puissent signaler leurs besoins ou problèmes, comme les services clients ou les outils de *ticketing*[9]. En interne, d'autres outils, inaccessibles aux clients, sont utilisés, comme les outils de gestion de stocks ou de simulation.

Ce qui caractérise les grandes entreprises comme ATOS, SAP, MICHELIN, CAP SOGETI, c'est l'existence de *terminologies métier* qui leur sont propres, et qui peuvent être formalisées sous formes d'ontologies, servant de référentiels.

[9] (Définition de Wiktionary) Centre d'assistance, système de gestion de requêtes créant un ticket pour chacune, assigné à une seule personne à la fois jusqu'à sa résolution et son archivage.

Dans tous les cas, les activités collaboratives en entreprise requièrent une identification claire des tâches, de leurs contraintes temporelles et de leurs séquencements, et des personnes qui en sont responsables.

I.1.1.2 Contexte : communication textuelle peu variée mais multiplicité d'outils

I.1.1.2.1 Communication textuelle

Lorsque les participants d'une collaboration sont à distance (par exemple, pas dans la même pièce), la communication se fait principalement par écrit, sous forme de courriels (asynchrones), mais aussi de tchats (synchrones) et de commentaires dans les outils spécialisés (cf. section I.1.1.2.2). Dans certains cas, elle peut se faire à l'oral, soit en présentiel, soit par téléphone ou avec un outil comme SKYPE[10].

Ce qui est caractéristique des communications écrites est :

- **le volume.** Les participants reçoivent une centaine de courriels par jour. Certains de ces courriels concernent les tâches en cours ou à réaliser, d'autres sont des messages d'information.
- **l'hétérogénéité thématique.** Un courriel concerne en effet souvent plusieurs sujets différents.
- **la multiplicité des intervenants**, avec des rôles et des liens hiérarchiques.
- **l'utilisation de listes de diffusion** dédiées aux projets individuels, qui minimise le temps de sélection de destinataires et facilite le filtrage automatique.

On voit apparaître des problèmes liés au volume de ces courriels et aux interruptions qu'ils engendrent. Le premier problème est que la culture d'entreprise d'aujourd'hui exige un traitement du courriel aussi immédiat que possible. Le deuxième est que l'interruption du travail par les courriels induit un changement de contexte qui, outre le temps de traitement du courriel, requiert ensuite un temps (Jackson et al., 2001) de remise en contexte pour reprendre l'activité interrompue.

I.1.1.2.2 Outils spécialisés

Il existe sur le marché une multitude de logiciels et de plates-formes de gestion de projets, dont des solutions commerciales et gratuites. Ce point est illustré par la page dédiée de WIKIPEDIA[11], qui recense plus d'une centaine d'outils de gestion de projets. Certains ciblent un besoin particulier (GANTT[12] pour la planification), d'autres sont des solutions complètes intégrant plusieurs outils différents (CODENDI[13]).

Il y a des outils de communication spécialisés. Certains sont des solutions complètes, comme WORKPLACE par FACEBOOK[14], WRIKE[15], SLACK[16] et ses concurrents.

Par contre, certaines compagnies, comme SAP ou SALESFORCE, se spécialisent dans des solutions sur mesure pour leurs clients.

Il y a aussi de petites équipes qui travaillent sur un logiciel, des équipes moyennes distribuées pour le support, (recueil d'exigences comme référentiel) et parfois de grandes équipes (par exemple pour la gestion des applications de SANOFI par CAP GEMINI).

[10] https://skype.com
[11] https://en.wikipedia.org/wiki/Comparison_of_project_management_software
[12] http://www.gantt.com/
[13] https://www.codendi.com/
[14] https://www.facebook.com/workplace
[15] https://www.wrike.com/
[16] https://slack.com

I.1.2 Activités collaboratives hors du cadre commercial ou industriel

On constate également un intérêt croissant pour les activités collaboratives hors du contexte commercial, qui présentent des similitudes et des différences intéressantes avec les activités collaboratives commerciales.

I.1.2.1 Situations

L'absence d'impératifs commerciaux n'implique pas toujours que l'ampleur du projet soit moindre. Par exemple, WIKIPEDIA et des ONG comme PAX HUMANA, AMNESTY INTERNATIONAL et autres, gèrent des activités de grande ampleur. On peut citer la communauté DISTRIBUTED PROOFREADERS qui affiche plus de 300 utilisateurs actifs en 24h et plus de 1000 en un mois.

Il y a aussi des projets de portée moindre, pouvant requérir de la collaboration organisée, qui vont du très petit (par exemple l'organisation d'un événement) au relativement grand (les projets de développement et/ou localisation de logiciels comme DEBIAN[17], FIREFOX[18]).

Ce qui distingue ces activités des activités commerciales est l'absence d'obligation autre que morale ou personnelle pour les participants, et l'absence fréquente de contraintes temporelles rigides, ainsi que l'absence très fréquente d'ontologies ou de référentiels explicites.

Il est intéressant également de noter que certaines activités qui sont collaboratives peuvent ne pas être coordonnées du tout, comme la rédaction d'un article WIKIPEDIA. Des discussions prennent tout de même place pour résoudre les problèmes liés à l'article, mais elles impliquent rarement l'assignation de la responsabilité d'une tâche.

Un autre exemple d'activité collaborative non nécessairement coordonnée est la post-édition contributive de documents dans le projet MACAU (voir Chapitre IV).

I.1.2.2 Communication essentiellement textuelle, souvent avec beaucoup de participants

Dans les situations dont nous parlons, la communication est principalement textuelle, avec possibilité de quelques réunions bien organisées via téléconférence.

La communication textuelle se passe la plupart du temps par courriel, et implique souvent beaucoup de participants (de 3 à une vingtaine, par exemple pour la préparation du rapport d'un grand laboratoire comme le LIG).

Pour certains projets, par exemple dans le développement collaboratif de logiciels, la communication peut également avoir lieu sur un forum ou une plate-forme, comme c'est le cas dans GITHUB[19]. Tout comme avec le courriel, ce type de communication est asynchrone.

Il est courant également d'utiliser des outils de téléconférence comme Skype, qui permettent non seulement une communication audiovisuelle synchrone, mais aussi de la communication textuelle et du partage de fichiers en parallèle.

I.1.2.3 Outils utilisés

Les outils utilisés sont souvent de types différents, car les besoins, mais aussi les contraintes financières, ne sont pas les mêmes. Bon nombre de plates-formes payantes, telles que TRELLO[20], REDBOOTH[21] ou WRIKE[22], proposent des services en « freemium », avec des fonctionnalités réduites.

[17] https://www.debian.org
[18] https://www.mozilla.org/fr/firefox/
[19] https://github.com/
[20] https://trello.com/ (plate-forme collaborative de gestion de projets en ligne par Kanban simplifié)
[21] https://redbooth.com/ (plate-forme de gestion de projets sous formes de listes de tâches)
[22] https://www.wrike.com/fr/ (plate-forme collaborative de gestion de flux de travail en ligne)

Il existe des outils généralistes et des outils visant une activité particulière. Par exemple, pour la gestion de colloques scientifiques et des ateliers associés, il existe des outils dédiés, tels que EASYCHAIR[23] ou START[24] de SOFTCONF, conçus pour un processus très spécifique consistant en plusieurs étapes bien définies (soumissions, relecture, constitution de comité, etc.), avec des rôles différents attribués aux utilisateurs ; ces outils sont gratuits ou peu chers, et disposent d'un vrai support professionnel quel que soit le modèle commercial.

Les plates-formes généralistes visent au minimum à proposer de la gestion de tâches. La plate-forme TRELLO, par exemple, fonctionne sur le principe du KANBAN [25] simplifié, avec une visualisation de fiches correspondant à des tâches dans des colonnes correspondant à leur état (à faire, en cours, terminé). L'utilisateur peut affecter les fiches à des personnes, établir des dates limites, glisser les fiches de colonne en colonne, etc.

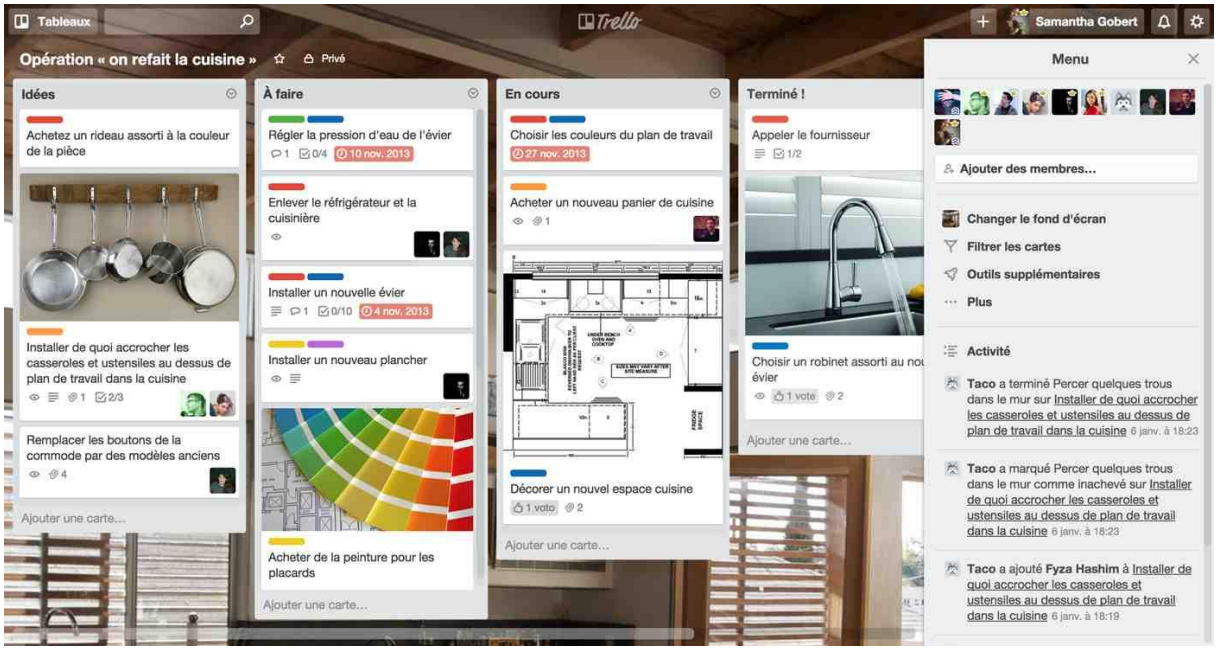


Figure 1 : capture d'écran de l'interface de TRELLO prise sur le site du logiciel

La généricité délibérée de ces outils implique qu'ils ne peuvent pas avoir de référentiel spécifique à un métier, et ne proposent pas de pouvoir en créer un. Par conséquent les outils génériques ne peuvent pas couvrir tous les besoins.

I.1.3 Le projet SYNAPS

A ce point, il est intéressant de décrire le projet SYNAPS, dans le cadre duquel cette thèse a été lancée.

I.1.3.1 Pourquoi en parler ?

Le projet SYNAPS, lancé par VISEO TECHNOLOGIES en 2013 par Franck Priore, avait pour but la création d'une plate-forme de collaboration qui puisse à la fois être générique et permettre aux utilisateurs d'exprimer leur vocabulaire métier via les moyens de la plate-forme.

La généricité de SYNAPS devait permettre de gérer à la fois des activités collaboratives professionnelles et personnelles.

[23] http://easychair.org/
[24] http://www.softconf.com/
[25] http://kanbanblog.com/explained/

Le développement de la plate-forme a malheureusement été arrêté courant 2016, malgré les nombreuses idées novatrices et prometteuses qui étaient déjà mises en œuvre dans la plate-forme prototype.

I.1.3.2 Caractéristiques visées

SYNAPS se présentait comme une plate-forme web permettant de créer et manipuler des objets de collaboration, de les organiser, de les typer, de les partager, et d'y travailler collaborativement. Il y avait également des moyens de communication et de notification.

SYNAPS était doté d'un graphe sous-jacent, nommé carte cognitive, qui regroupait tous les éléments de la plate-forme. C'est la première fois qu'on allait avoir une représentation (peut-être simplifiée) d'un référentiel dans un tel outil, *a priori* générique.

SYNAPS proposait une organisation d'objets en une hiérarchie d'espaces de partage et de collaboration, qui étaient un moyen de grouper les objets, d'y associer des droits d'accès et de les visualiser à l'aide de bureaux interactifs partagés. Les espaces pouvaient être partagés, importés et rattachés à d'autres espaces, formant un treillis, plutôt qu'une arborescence de fichiers classique.

Les bureaux étaient de deux types :

- le *whiteboard*, analogue au bureau d'un système d'exploitation, permettant de manipuler des objets qui pouvaient y être déposés.
- le *listboard*, une vue en colonnes de fiches analogue à celle de TRELLO, mais plus riche, permettant d'organiser des fiches suivant des dimensions prédéfinies par l'utilisateur, exprimées sous formes d'étiquettes sémantiques nommées *tags*. Ces dimensions pouvaient être des états d'une tâche correspondant à la fiche (par exemple « à faire ») ou son type (par exemple *bug* ou *feature* dans le cas d'un développement logiciel). Une fiche pouvait avoir plusieurs *tags* (p. ex. « *bug* » et « à faire »), et on pouvait alors afficher des listes de fiches correspondant à une expression logique sur les tags (par exemple lister toutes les *features* en cours de développement).

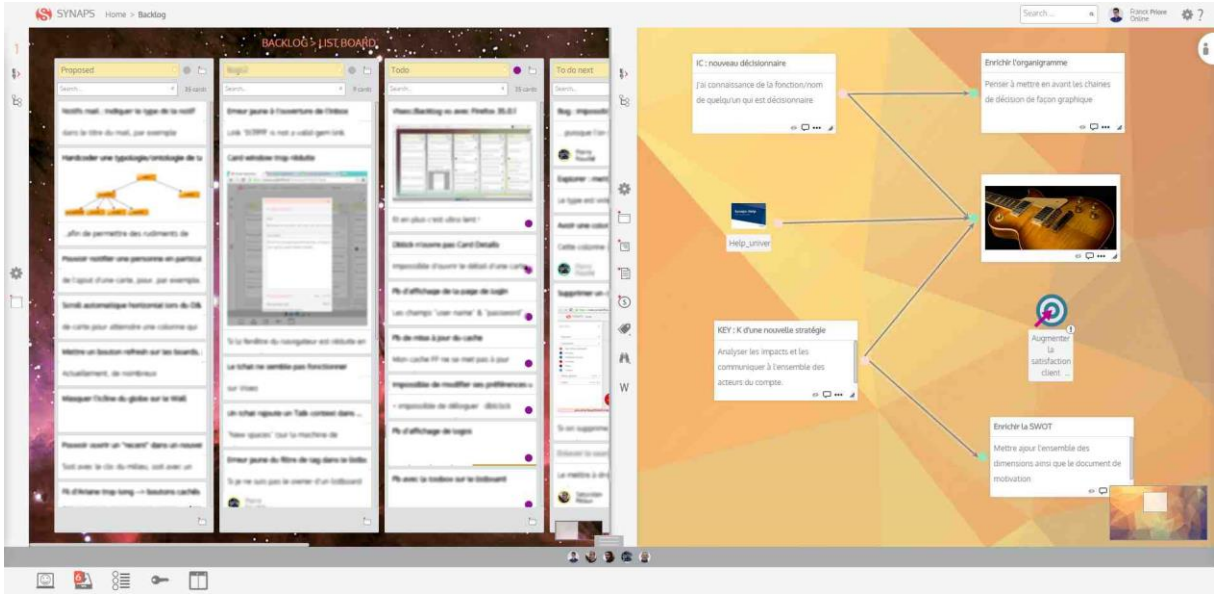


Figure 2 : capture d'écran de SYNAPS, illustrant les deux types de bureau

SYNAPS était doté d'autres fonctionnalités utiles pour la collaboration, comme le partage de fichiers, un outil de prise de notes, et un autre permettant la rédaction collaborative de documents. Les documents et fichiers partagés pouvaient être versionnés.

Les utilisateurs pouvaient être affectés à des groupes, et les groupes rattachés à des espaces, ouvrant à tout utilisateur du groupe le droit d'accéder aux éléments de l'espace.

Afin d'avoir une passerelle avec la gestion de la collaboration effectuée en dehors de la plate-forme, un plugin Outlook avait été développé pour transférer les messages relatifs à la collaboration dans la plate-forme, et le but de la thèse était alors de doter SYNAPS de fonctionnalités d'assistance intelligente, dont l'extraction de tâches à partir de ces correspondances électroniques, et le rattachement automatique du message aux contextes auxquels il devait appartenir.

I.1.3.3 Évolution du sujet à partir de ce projet

Après l'arrêt du développement de la plate-forme, le sujet de la thèse a conservé le thème de l'analyse des textes liés à la collaboration, mais le contexte applicatif a inévitablement changé, la plate-forme d'implémentation n'étant plus utilisée.

Le contexte applicatif final de cette thèse est celui du projet FUI REUS, dont VISEO est membre.

Bilan provisoire

La gestion des activités collaboratives est une problématique qui suscite de nombreux travaux. Il y a un désir de construire un outil de gestion de projets collaboratifs qui soit à la fois assez générique et assez expressif pour supporter à la fois des activités fortement structurées et dotées d'un référentiel, et des activités moins structurées, mais il est très difficile de construire un tel outil, comme on l'a vu avec SYNAPS.

Nous avons vu qu'en pratique les activités collaboratives souffrent du problème de surcharge de courriels. De nombreuses personnes mènent plusieurs projets en parallèle, ce qui augmente d'autant les distractions, la charge cognitive et temporelle.

C'est pourquoi nous nous sommes concentré sur un type d'aide plus précis, l'assistance à l'utilisation de courriels, tout en gardant l'ambition de le faire dans les deux cadres (en et hors entreprise). Par conséquent, il faut que cette aide puisse être fournie, qu'il y ait ou non un référentiel, ou une ou des ontologies.

I.2 Aider des participants à traiter des textes et des flux textuels liés à des activités collaboratives

En partant des constats de la section précédente, nous décrivons dans cette section les aides envisageables pour rendre les collaborations plus efficaces. Cette section présentera les types d'assistance envisagés, un bref état de l'art, ainsi que nos désidérata.

I.2.1 Types d'assistance envisagés

I.2.1.1 Aide en temps réel

Le type d'aide le plus immédiat est l'aide en temps réel, car elle s'attaque au problème de la surcharge d'information dans un contexte de fortes contraintes temporelles. Cela implique de traiter les textes et les flux textuels dès leur arrivée, et ainsi par exemple, d'analyser les courriels pour en extraire les informations les plus importantes, les trier par priorité, et si possible les mettre en relation avec un référentiel.

I.2.1.1.1 Aide à l'accès à l'information pertinente

Un des besoins évidents est l'aide à l'accès à l'information pertinente. Il peut s'agir de tâches dans les courriels, ou de fragments de documents utiles dans le contexte d'apprentissage.

Quelques solutions non automatiques existent déjà. On peut citer le plugin ACTIVEINBOX[26] pour GMAIL, ou bien TASKFORCEAPP[27], ou encore GOOGLE INBOX[28].

De plus en plus de clients de messagerie proposent une classification de courriels en `perso, pro, spam`.

Office Outlook classifie les courriels en `prioritaires` et `pêle-mêle`. LIVE.FR a récemment commencé à classifier en `Prioritaire` et `Autres`. L'équipe WATSON D'IBM collabore avec le Crédit Mutuel[29] pour analyser des courriels des clients et suggérer des réponses pré-formulées, apprises sur des correspondances passées.

I.2.1.1.2 Aide à la mise en relation de fragments textuels avec un référentiel ou des ontologies

Nous proposons aussi un nouveau type d'aide qui consiste à mette en relation des fragments textuels avec un référentiel (tiré par exemple à partir d'un document décrivant un projet) ou des ontologies.

Il s'agirait, par exemple, de relier des fragments d'un courriel avec des tâches décrites dans un lot de travail d'un document de projet.

I.2.1.2 **Aide à l'analyse a posteriori de courriels liés à un événement majeur**

Les collaborations engendrant beaucoup de données textuelles, on pourrait vouloir les analyser a posteriori afin d'identifier des problèmes survenus.

Par exemple, dans le cadre de l'enquête sur les fraudes financières de la compagnie ENRON, un immense corpus contenant les correspondances internes de 150 employés a été rendu public[30].

I.2.2 Bref état de l'art

L'intérêt pour la gestion d'activités à travers les courriels date au moins de 1986, quand Terry Winograd proposa son système et sa modélisation du cycle de vie de tâches. Bien que l'idée fût intéressante et novatrice, elle n'eut pas de succès, car son implémentation aurait requis des efforts de spécification trop prohibitifs de la part des utilisateurs, qui préfèrent exprimer leurs tâches avec du texte simple.

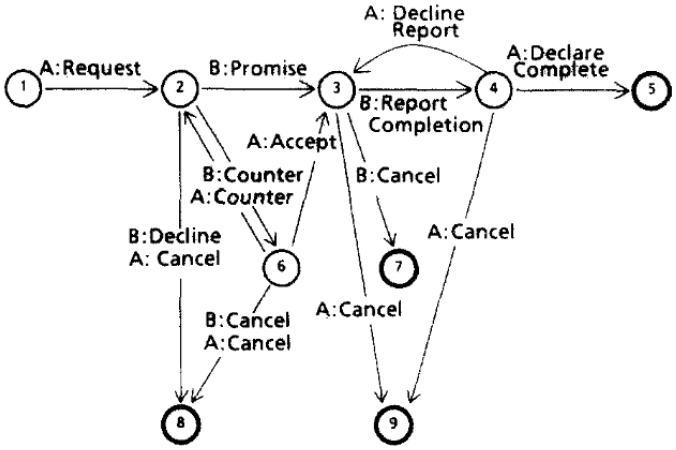


Figure 3 : schéma de conversation autour d'une tâche, repris de (Winograd et Flores, 1986)

[26] http://www.activeinboxhq.com/
[27] https://www.taskforceapp.com/
[28] https://www.google.com/inbox/
[29] http://www.lemonde.fr/entreprises/article/2017/04/20/le-credit-mutuel-deploie-le-robot-d-intelligence-artificielle-watson-dans-son-reseau_5114032_1656994.html
[30] https://www.cs.cmu.edu/~./enron/

I.2.2.1 État de l'art très synthétique

Le livre *Natural Language Processing for Social Media* d'Atefeh Farzindar et Diana Inkpen (2015) fait un état de l'art du TAL pour des textes issus de médias sociaux. On y constate une absence notable d'outils pour les courriels.

IBM propose l'outil VERSE[31], en le positionnant comme une messagerie repensée et épurée, qui met en évidence les courriels et documents importants, ainsi que les obligations de l'utilisateur (et celles des autres utilisateurs envers lui).

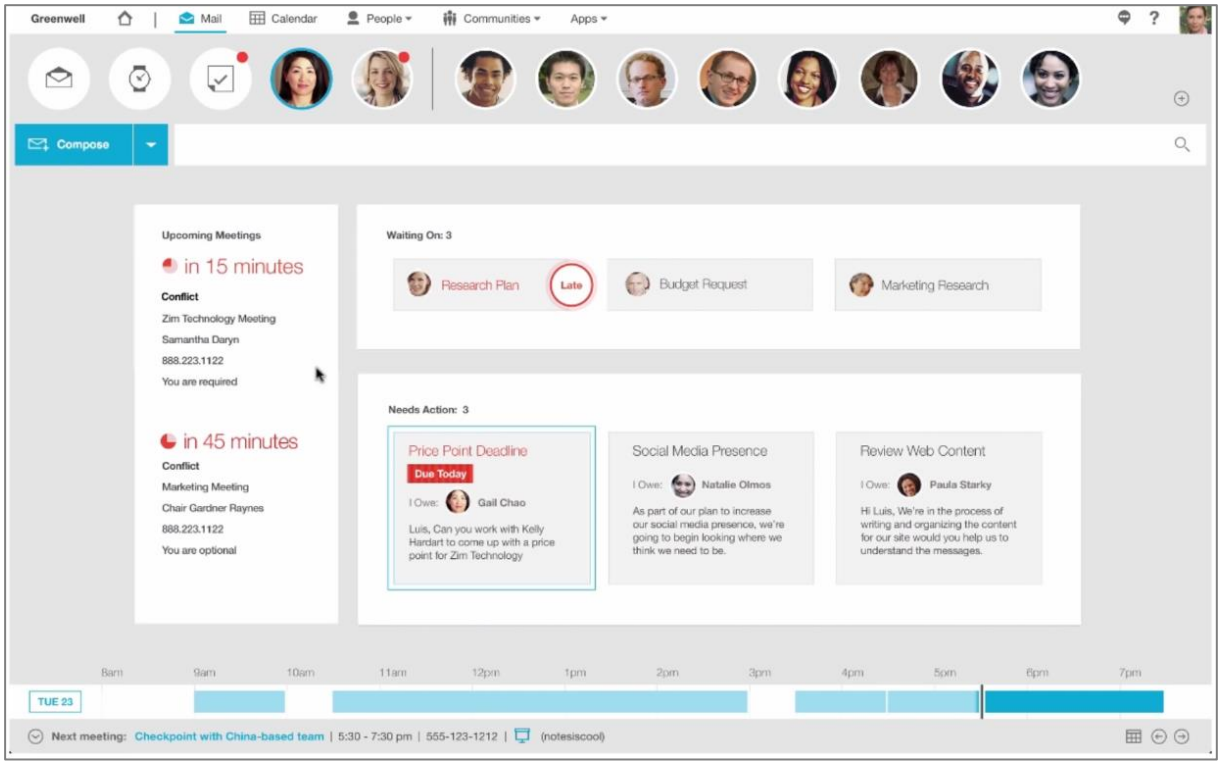


Figure 4 : image d'écran d'IBM VERSE issue d'une vidéo promotionnelle

Microsoft propose OFFICE DELVE[32], un outil de recherche d'information dans les divers services de la suite OFFICE. DELVE construit automatiquement un réseau de collaborateurs de chaque utilisateur et fait émerger les documents pertinents issus de courriels, de SHAREPOINT, etc.

[31] https://www.ibm.com/fr-fr/marketplace/business-email-platform
[32] https://products.office.com/en-us/business/intelligent-search?tab=Discovery

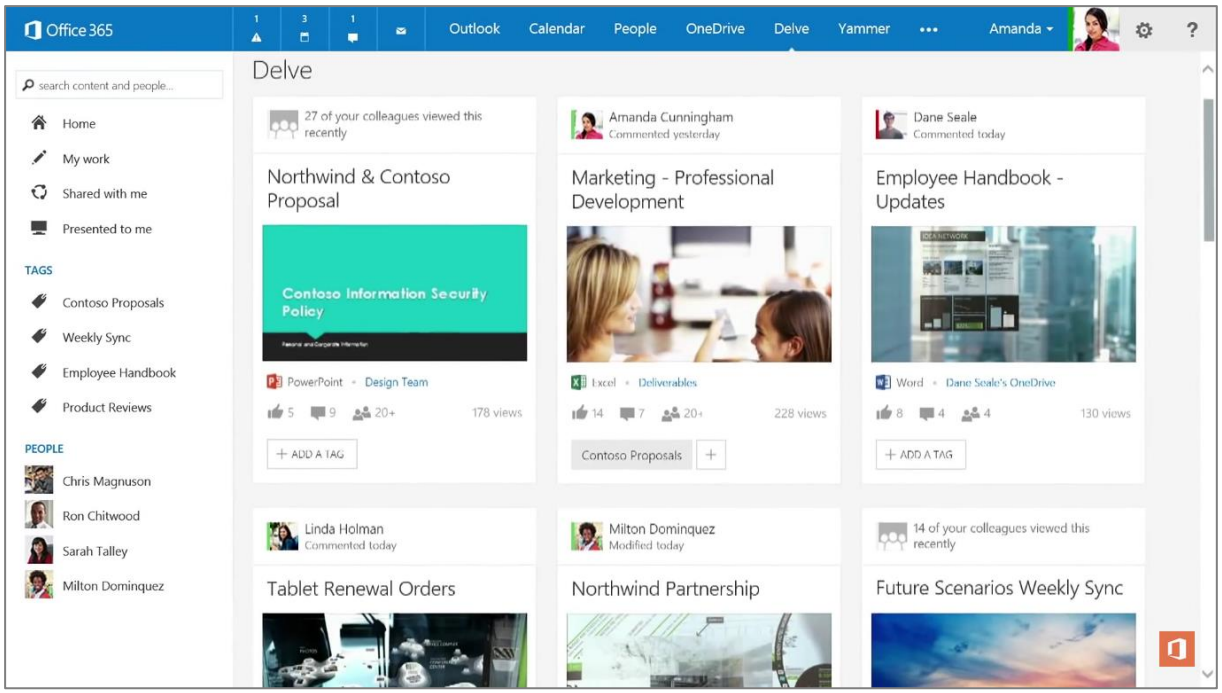


Figure 5 : image de OFFICE DELVE issue d'une vidéo promotionnelle par MICROSOFT

Il existe aussi de nombreux outils non automatiques qui s'intègrent dans les clients de courriel. Des plugins pour OUTLOOK d'OFFICE comme WUNDERLIST[33] (cf. capture d'écran ci-dessous) ou TASKS IN A BOX[34] permettent de transformer un courriel en « chose à faire » par un clic sur un bouton.

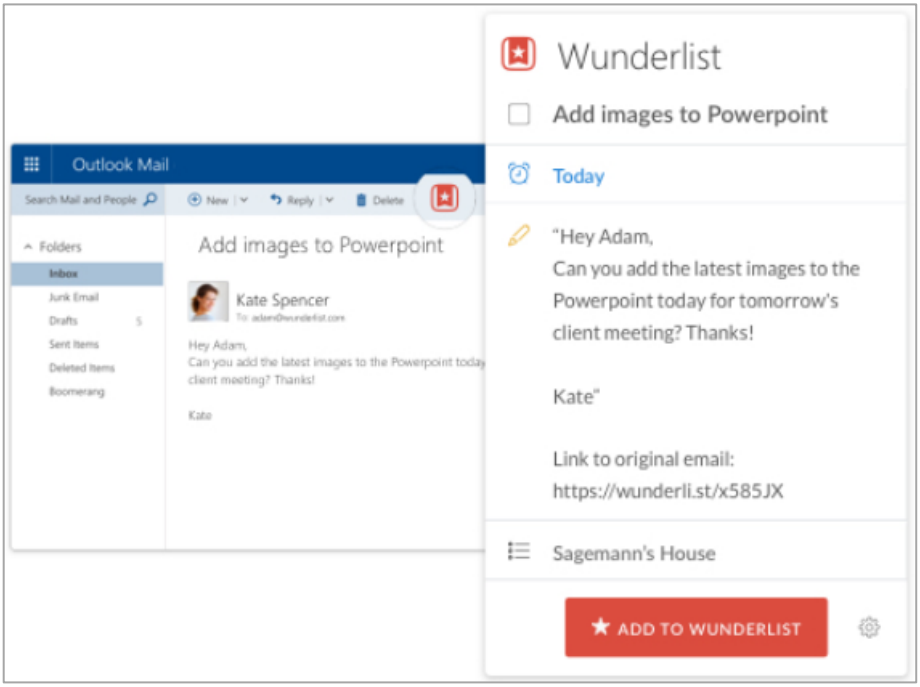


Figure 6 : image promotionnelle du plugin WUNDERLIST pour OUTLOOK

Il existe des plugins similaires pour GMAIL ; tous ont pour particularité d'être manuels, c'est-à-dire de ne pas détecter automatiquement les courriels qui contiennent des choses à faire.

[33] https://www.wunderlist.com/fr/
[34] https://tasksinabox.com/

Certains, comme SORTD[35], proposent une interface à la TRELLO[36], permettant l'affectation à un courriel d'une catégorie relative à des activités, et la visualisation de ces activités en mode KANBAN simplifié.

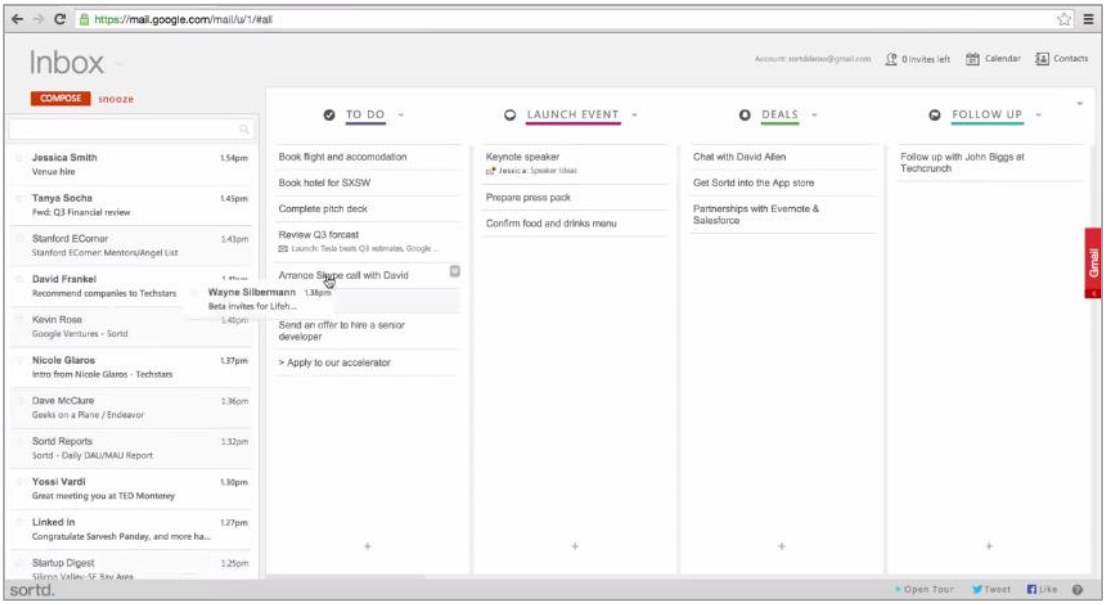


Figure 7 : capture d'écran de la vidéo promotionnelle de Sortd, un plugin pour Gmail.

En ce qui concerne l'extraction automatique de choses à faire, un plugin pour OUTLOOK, intégré par défaut à OUTLOOK 2016, se distingue. Il repère les phrases en anglais qui contiennent une demande d'action (*action item*[37]) et propose d'associer un indicateur de suivi au message entier.

Nous savons d'autre part qu'une équipe de Microsoft a travaillé sur le résumé de courriels centré sur les tâches *(task-focused email summary)* (Ringger et al. 2004), et Eric Ringger nous a informé dans une correspondance privée que le prototype verrait le jour sous forme d'un plugin OUTLOOK, donc nous pouvons supposer qu'il s'agit du résultat de leurs travaux.

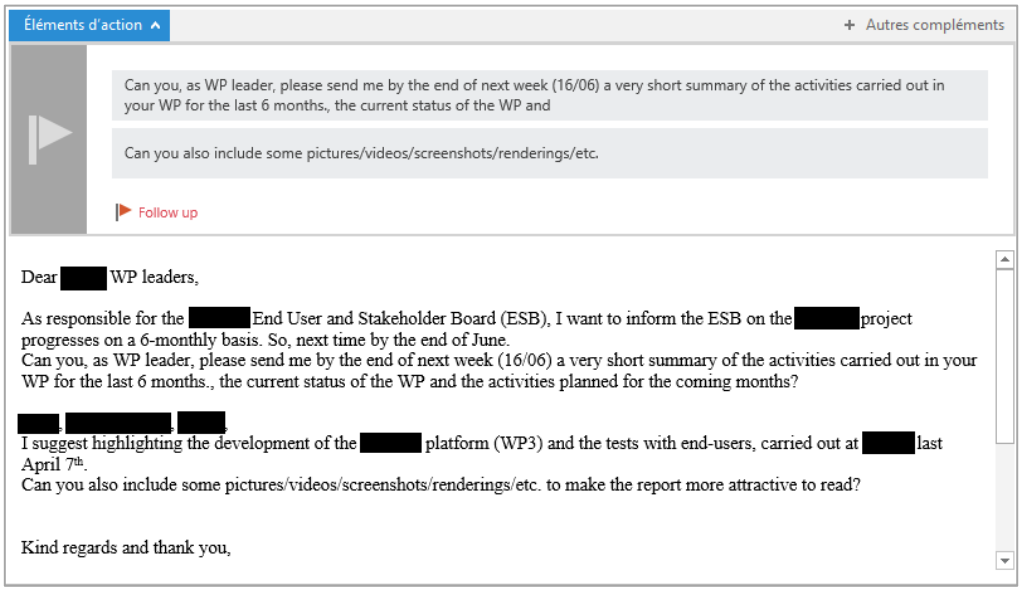


*Figure 8 : le plugin Outlook en cours de repérage d'*action items.

[35] https://www.sortd.com/
[36] https://trello.com/
[37] Il s'agit bien d'une demande d'action, et pas d'un « élément d'action », comme c'est parfois traduit.

Ce plugin est discret dans sa façon de présenter les résultats : lors de la lecture des courriels où des éléments d'action ont été repérés, un bouton « Éléments d'action » est visible dans un bandeau au-dessus du texte du message. Un clic fait déplier la liste des phrases repérées. Il n'y a aucun autre indicateur ailleurs, par exemple dans la liste des messages, où il pourrait être utile de l'afficher pour attirer l'attention de l'utilisateur.

L'exemple ci-dessous illustre un problème auquel on se heurte lors du traitement automatique du texte des courriels : les signatures peuvent contenir des phrases formulées comme des demandes.

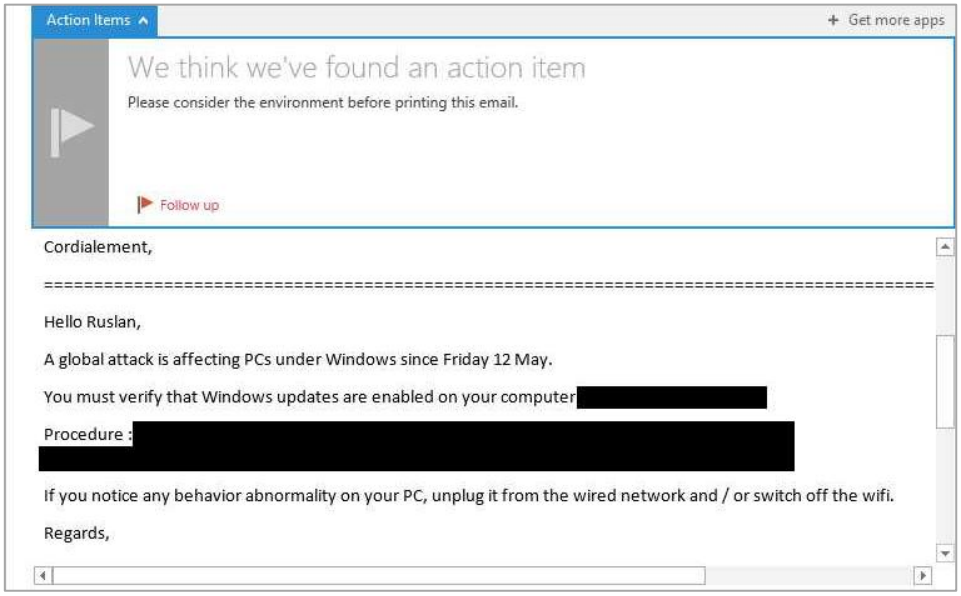


Figure 9 : le plugin OUTLOOK a repéré une phrase issue de la signature du message, mais n'a pas repéré la tâche principale, qui est de vérifier que les mises à jour sont activées.

L'affichage de la liste d'éléments d'action est relativement lent, ce qui est probablement dû au fait que le plugin la télécharge depuis le serveur à la demande. Cette lenteur limite l'utilité du plugin, puisqu'on a souvent le temps de lire le message entre temps, surtout lorsqu'il est relativement court.

I.2.2.2 Bilan

Nous avons vu que, bien que la gestion du flux des tâches pose des problèmes que beaucoup abordent, il n'existe aucun outil qui convienne vraiment :

- il n'y a pratiquement pas d'outils automatiques. Celui qui existe n'extrait pas les attributs des tâches (dates limites, personnes attributaires).
- il n'y a pas non plus de mise en relation des fragments de texte avec un référentiel ou entre eux. Il n'y a pas non plus de modèle de citation ou de conversation.
- surtout, il n'existe aucun outil pour le français.

I.2.3 Ce qu'on voudrait pouvoir faire

I.2.3.1 Sur des courriels

Nous voudrions développer un outil (sous forme de plugin s'intégrant dans un outil de messagerie) qui fonctionnerait sur un principe semblable au plugin OUTLOOK décrit ci-dessus, mais qui le généraliserait et l'améliorerait. Plus particulièrement, il s'agirait d'un plugin multilingue, qui repérerait non seulement les phrases porteuses de demandes, mais également les attributs de ces demandes, comme les contraintes temporelles et les personnes impliquées.

Après repérage de phrases, il les typerait en fonction du type d'action qu'elles exigent. Un score d'urgence et/ou d'importance serait éventuellement associé au courriel.

Enfin, si le courriel contenait plusieurs demandes, il faudrait identifier leur séquencement, s'il est indiqué par le texte (par des articulateurs temporels, comme « ensuite », « cela fait », « après avoir *ParticipePassé* »…).

On peut envisager d'aller plus loin, en essayant de relier les tâches à un référentiel du projet, donné sous forme d'une ontologie sous PROTEGE[38] ou SIGMA[39].

I.2.3.2 Sur des documents objets de l'activité collaborative

On voudrait pouvoir également analyser les documents qui sont l'objet de l'activité collaborative.

Dans le cas de la post-édition, il s'agit (dans notre contexte) de documents pédagogiques, Ils sont souvent très complexes, comme on le verra, d'où la nécessité d'introduire une modélisation adaptée.

I.2.3.3 Sur d'autres flux textuels

Le traitement de « verbatims » produits par les utilisateurs ne se limite pas qu'aux courriels. On pourrait vouloir analyser des blogs (il y a une thèse CIFRE en cours à la SNCF sur l'extraction d'itinéraires à partir de blogs), des microblogs, des forums, des tchats, des dialogues oraux transcrits, des tweets… Tous ces textes présentent des caractéristiques qui leur sont propres ; les analyser serait certainement intéressant, mais aurait nécessité au moins un an de plus de travail de thèse.

I.3 Difficultés et problèmes

Dans cette section, nous montrerons qu'arriver à implémenter les aides décrites dans la section précédente impose de se confronter à des problèmes difficiles liés à la modélisation et au traitement des documents.

I.3.1 Modéliser puis traiter les flux de courriels et les documents

I.3.1.1 Difficultés de modélisation

Comment modéliser les conversations écrites ? Elles forment des fils de discussion avec une structure arborescente. Les messages peuvent citer d'autres messages, en modifiant leur texte.

Pour ce qui est des courriels, ce sont (dans le contexte d'activités collaboratives) des dialogues multiparties, et ils forment un *graphe de citations*. Le texte d'un courriel peut être intentionnellement légèrement modifié par son auteur, ou un autre participant, ce qui nous pousse à rechercher une séquence minimale de modifications menant au fragment cité.

Quant aux documents techniques ou pédagogiques (cf. Ch. 2 et 4), ils sont interconnectés structurellement ou sémantiquement, et leur structure interne peut être complexe.

Dans les deux cas, il faut tenir compte de la nature possiblement « réticulaire » de ces documents.

I.3.1.2 Difficultés de traitement

Il y a plusieurs obstacles techniques importants au traitement immédiat du contenu textuel : (1) la nécessité de segmentation et de normalisation, (2) la reconstruction des messages et des

[38] https://protege.stanford.edu/
[39] http://articulatesoftware.com/

discussions, (3) les problèmes liés à la multiplicité des formats, et enfin (4) la conception logicielle de plugins opérationnels.

I.3.1.2.1 Segmentation, normalisation

Avant de pouvoir analyser le texte, il faut le préparer. Cela inclut la *segmentation*, c'est-à-dire le *découpage* en unités de traitement, et la *normalisation*, qui ramène si nécessaire les éléments textuels à une forme standard. Cela inclut la reponctuation, la substitution des émoticons par des occurrences spéciales, etc. Des détails sont donnés au Chapitre II.

Ces processus peuvent être interdépendants, car une bonne segmentation peut dépendre d'une normalisation préalable.

I.3.1.2.2 Reconstruction (pour les courriels)

Un courriel issu d'une conversation contient souvent les corps des messages qui l'ont précédé et qui sont chacun une réponse au précédent. Avant de pouvoir être analysé, un tel courriel doit être segmenté en messages individuels, qui sont alors chaînés dans un *fil de discussion*.

Cette tâche n'est pas triviale à cause de la présence de déformations non signalées comme telles, du besoin d'identification des en-têtes, de l'entremêlement de plusieurs messages.

De plus, le texte d'un message cité peut être modifié par la personne qui le cite, ce qui implique que le texte original ne puisse pas être reconstitué.

Enfin, lorsqu'on dispose de plusieurs messages (au format `eml`[40]) issus d'une conversation, il peut être possible de reconstruire le fil (ou l'arbre) de discussion à l'aide de métadonnées contenues dans les fichiers. Un exemple de cela se trouve dans la Figure 23 ci-dessous.

I.3.1.2.3 Conversions

On est inévitablement confronté à la nécessité de gérer des formats de fichiers différents. La méthode qui nous semble la meilleure est de convertir les différents formats vers un ou plusieurs autres, plus simples à traiter. Par exemple, pour la post-édition de documents pédagogiques, nous convertissons les formats DOC(X)[41], PPT(X)[42], LATEX[43], ODT[44], PDF[45] vers le format HTML[46], pour lequel sont développés les outils de post-édition et de traitement.

Pour les courriels, le format de partage le plus commun est EML. Ces fichiers contiennent des données textuelles en HTML ou en texte non balisé, qui sont le plus souvent encodées dans un encodage spécifique, comme BASE64[47] ou QUOTED-PRINTABLE[48] (et pas UTF-8[49]). Dans ce cas, avant de traiter ces données, nous les convertissons en UTF-8.

[40] https://www.loc.gov/preservation/digital/formats/fdd/fdd000388.shtml
[41] https://en.wikipedia.org/wiki/Docx
[42] https://msdn.microsoft.com/en-us/library/dd926741(v=office.12).aspx
[43] https://www.latex-project.org/
[44] https://en.wikipedia.org/wiki/OpenDocument
[45] https://en.wikipedia.org/wiki/PDF
[46] https://html.spec.whatwg.org/multipage/
[47] https://en.wikipedia.org/wiki/Base64
[48] https://en.wikipedia.org/wiki/MIME
[49] http://www.utf-8.com/

I.3.1.2.4 Réalisation de plugins opérationnels

Une fois les algorithmes implémentés, il faut les mettre en œuvre dans un outil qui ou bien soit autonome, ou bien s'intègre avec des outils existants, par exemple un plugin dans un client de courriel (OUTLOOK[50], THUNDERBIRD[51], MAIL[52], etc.).

La mise en production est un moment-clé pour un logiciel, car il est pour la première fois mis à disposition de l'utilisateur final.

Ainsi, un plugin doit préalablement être analysé, si possible prouvé, et testé pour vérifier qu'il n'introduit pas d'instabilités ou de vulnérabilités.

Il doit également être maintenu en permanence, afin de garantir son bon fonctionnement suite aux mises à jour du logiciel-hôte. Il doit répondre aux critères d'utilisabilité (réactivité, ergonomie de l'interface). Enfin, on doit permettre à l'utilisateur de donner des retours, qu'on utilisera pour améliorer le plugin.

I.3.1.3 **Difficultés de construction des outils linguistiques**

I.3.1.3.1 Constituer et éventuellement préparer des corpus représentatifs anonymisés

Une des difficultés inévitables est la constitution de corpus représentatifs, les données étant nécessaires quelle que soit la méthode (experte ou empirique[53]) utilisée pour le développement des outils.

Dans le cas des courriels se pose de plus le problème de la confidentialité des échanges et de la protection de la vie privée[54]. Il faut donc que ces corpus représentatifs soient anonymisés avant de pouvoir être traités par les spécialistes.

I.3.1.3.2 Choisir la « meilleure méthode »

Une deuxième difficulté est le choix de l'approche à utiliser pour concevoir l'outil. En effet, il y a toujours plusieurs approches possibles, et elles diffèrent en termes de délai de mise en route, de taille et type données nécessaires (brutes, annotées), etc.

De manière générale, une approche experte requiert moins de données linguistiques qu'une approche fondée sur l'apprentissage automatique. Les règles peuvent être développées de façon à être facilement interprétables par des humains. Cependant, les approches empiriques ont l'avantage d'être plus facilement automatisables.

Un outil doit pouvoir s'adapter à de nouvelles données, donc un mécanisme d'amélioration à partir de retours d'utilisateurs doit être mis en place.

Il y a aussi beaucoup de contraintes « opérationnelles », en particulier le délai de mise en route pour que ce soit utile, la possibilité ou non d'utiliser du travail parcellaire[55] pour annoter les corpus sélectionnés (ou construits), ainsi que la possibilité d'amélioration incrémentale ultérieure, la plus automatique possible et donc fondée sur des « retours naïfs ».

---

[50] https://outlook.com
[51] https://www.mozilla.org/fr/thunderbird/
[52] https://support.apple.com/fr-fr/mail
[53] On regroupe ici les méthodes qui utilisent directement des données brutes ou annotées, ou encore qui reposent sur de l'apprentissage automatique.
[54] http://eduscol.education.fr/internet-responsable/ressources/legamedia/correspondance-privee-et-monde-numerique.html
https://www.editions-tissot.fr/actualite/droit-du-travail-article.aspx?secteur=PME&id_art=2387&titre=E-mails+au+travail+%3A+les+limites+du+secret+de+la+correspondance
https://www.arobase.org/entreprise/email-personnel.htm
[55] Nous préférons ce terme à « myriadisation » et « crowdsourcing », ce dernier impliquant trop l'idée de tâches ultra simples et de paiement à coups de lance-pierre.

I.3.1.3.3 La réaliser et la mettre en œuvre

La réalisation des outils nous confronte à des difficultés informatiques.

Afin de pouvoir réutiliser des outils déjà existants développés par des tiers (par exemple des outils d'analyse morphosyntaxique), on doit maîtriser de multiples langages informatiques.

Les outils peuvent parfois poser des problèmes de licence. Cela est particulièrement pertinent dans le contexte d'un développement dans un but commercial. Certains logiciels deviennent obsolètes et ou peuvent poser des problèmes de fiabilité. Enfin, on peut ne pas disposer d'outils pour certaines langues.

I.3.2 Traiter le multilinguisme

I.3.2.1 Multilinguisme des documents et des échanges, « code switching »

On est naturellement confronté au multilinguisme sur le web, et les échanges électroniques ne font pas exception. Plusieurs cas de figure se présentent.

- Une conversation est multilingue, mais chaque message est monolingue.
- Des messages contiennent des fragments en différentes langues, mais ces fragments constituent des « unités autonomes », par exemple des paragraphes.
- Des phrases contiennent des portions en différentes langues (« bascule de code » ou *code-switching*).

Le premier cas est le plus facile à traiter, car l'identification automatique de la langue fonctionne le mieux sur (ce qu'on peut espérer être) des blocs textuels homogènes et de taille suffisante. Le deuxième cas est plus difficile, car les fragments homogènes sont plus petits, et donc il y a plus de chances d'erreur pour un outil d'identification de langue. Le troisième cas est naturellement le plus problématique.

I.3.2.2 Nécessité d'outils pour le français (au moins) comme pour l'anglais

Afin de traiter des documents en français, nous devons disposer d'outils d'analyse linguistique fiables. Cela concerne toutes les phases du traitement, de la normalisation à l'analyse sémantique. Or l'anglais est bien mieux doté que le français de ce point de vue. Il possède non seulement des ressources linguistiques informatisées de grande taille, dont il n'y a pas d'équivalent en français[56], mais aussi des outils d'analyse éprouvés (ex. STANFORD CORENLP).

I.3.2.3 Analyse linguistique et nécessité de bons outils d'analyse ouverts à au moins 80 langues

Avec l'extension de l'accès à l'informatique et à Internet dans les pays et régions jusqu'ici peu équipés, il est nécessaire de prévoir des outils capables de traiter les langues émergentes de ces régions. Par exemple, les pays de la francophonie comportent au moins 320 langues, dont la plupart sont peu ou pas du tout dotées.

Cette réalité est reconnue par les grands éditeurs de logiciels. Par exemple, à sa sortie, WINDOWS 10 était disponible en 111 langues, dont le luxembourgeois.

Le multilinguisme croissant du Web est illustré par le fait qu'actuellement,13 langues disposent de plus de 1 million d'articles chacune dans WIKIPEDIA, et 45 autres ont plus de 100 000 articles chacune. En témoigne aussi le fait que MOZILLA FIREFOX est disponible en 90 langues.

[56] Par comparaison, les analyseurs du français ont une couverture lexicale 2 ou 3 fois moindre que Stanford Parser ou, dans le passé, PEG, ou l'analyseur « slot grammars » de MacCord dans LMT (IBM, 300K entrées dans le dictionnaire anglais), qui a été utilisé pour traiter 1M livres dans l'expérience Watson-Jeopardy. https://www.ibm.com/watson/tech.html

Néanmoins, des langues très parlées, comme le swahili (une estimation récente[57] donne environ 100 millions de locuteurs) sont aujourd'hui sous-représentées sur le Web, bien qu'on s'attende à leur émergence dans un avenir proche.

Il est illusoire de penser que tout le monde utilise et utilisera de plus en plus l'anglais. Au contraire, la proportion de l'anglais ne cesse de baisser[58].

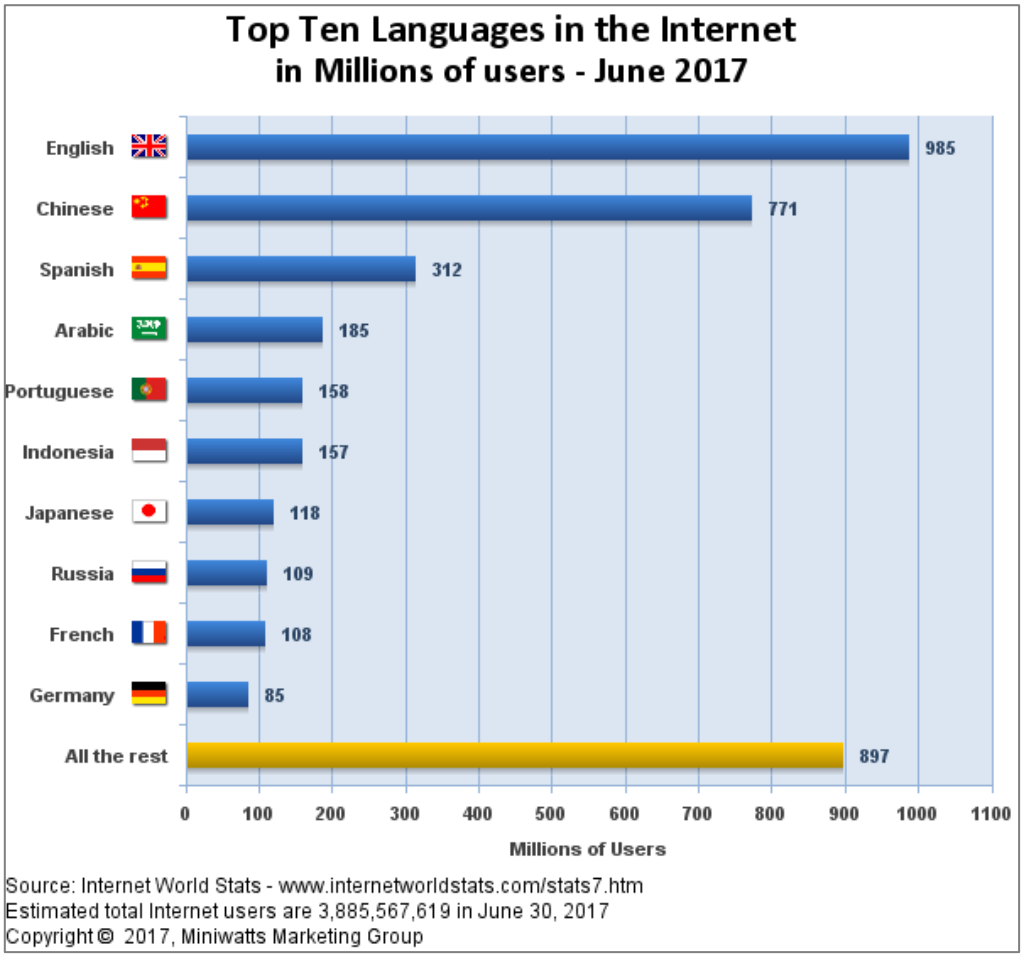


Figure 10 : langues sur le Web, en millions d'utilisateurs

I.3.3 Donner du sens aux textes

On veut pouvoir interpréter documents et courriels écrits en langue naturelle, c'est-à-dire les relier à un référentiel, qui peut être une modélisation du contenu ou du domaine. Pour cela il est nécessaire de construire cette modélisation, si possible sous forme d'ontologie, et de fabriquer des outils d'analyse capables de faire le lien entre cette modélisation et le texte.

La modélisation du contenu permettrait d'interpréter le contenu selon sa structure. Il s'agirait, par exemple, d'identifier les différentes parties d'un document ou d'un courriel, comme l'introduction, les phrases de salutation, le corps de texte, les fragments cités, etc.

I.3.3.1 Documents

Comprendre les documents liés aux projets peut permettre d'interpréter les échanges. Les projets menés en entreprise sont souvent guidés par des documents. Par exemple, les documents définissent des lots de travail (*work packages*), leur décomposition en tâches, et leurs

[57] https://www.ethnologue.com/language/swh
[58] https://internetworldstats.com/stats7.htm

dépendances et séquencements (GANTT). D'où l'intérêt de pouvoir interpréter les documents pour pouvoir comprendre les échanges autour des projets.

Pour la modélisation du domaine, il y a trois dimensions à considérer :

- ce dont parle le texte (le thème).
- les types et sous-types d'énoncés (demande d'action, d'information, interdiction, et autres).
- la structuration discursive du texte ou de la conversation.

On peut imaginer de passer par un étiquetage en acceptions interlingues, via les UW UNL (Uchida, 2006) (lexèmes interlingues). Cela aurait pour avantage de permettre de concevoir des systèmes plus génériques, puisqu'on pourrait effectuer l'analyse sémantique non plus sur la langue source, mais sur le résultat de l'enconversion vers UNL.

Dans sa thèse, D. Rouquet a utilisé cette méthode pour le système OMNIA de recherche interlingue d'images, en utilisant leurs légendes annotées par des UW. On trouvera les détails dans (Falaise et al., 2010), et (Rouquet, 2012).

I.3.3.2 **Courriels**

Donner du sens aux courriels veut dire les interpréter par rapport à un ou plusieurs référentiels.

Voici ceux qui sont le plus souvent envisagés.

- Description des participants.
- Graphe temporel des échanges.
- Modélisation des tâches.
- Organisation du projet en cours, si possible en tant qu'instance d'un modèle générique des projets collaboratifs, i.e. décrivant ses lots et tâches.
- Evolution des sous-courriels (citations, éventuellement modification d'un courriel dans un autre.

I.3.4 Trouver comment bien évaluer pour améliorer de façon utile

I.3.4.1 **Trouver des méthodes d'évaluation pertinentes**

Il y a différentes façons d'évaluer la qualité d'un système et nous voulons aller au-delà des classiques « Précision—Rappel—F-mesure » et trouver des critères qui reflètent l'utilité réelle du système pour l'utilisateur (qualité d'usage).

Par exemple, un système qui ne trouverait que 50% des énoncés renvoyant à une certaine tâche, mais qui ne « raterait » aucune tâche, serait très utile.

I.3.4.2 **Trouver comment capitaliser sur les résultats courants pour améliorer les outils (« boule de neige »)**

L'utilisateur peut être amené à évaluer le système par le biais d'un retour, et cette évaluation peut servir à améliorer incrémentalement et automatiquement le système.

Par exemple, on peut s'inspirer du tri « courrier pêle-mêle » de OUTLOOK, qui demande des commentaires sur son fonctionnement ou de la différenciation en « courrier prioritaire » et « autres » dans LIVE.FR.

Dans le même ordre d'idées, BING et GOOGLE demandent des évaluations grossières de qualité de leurs résultats de traduction avec [👍] ou [👎].

Synthèse et plan de la suite

On a montré que vouloir apporter de l'aide dans le traitement des textes des échanges et des documents objets de collaboration implique de résoudre un ensemble de problèmes linguistiques et informatiques relatifs au traitement de la structure et du contenu de ces objets.

On montre maintenant comment structurer ces documents suivant des représentations formelles de forme et de sens : les ontologies.

Dans la suite, nous décrivons la spécification, le développement et l'évaluation des outils :

- de traitement de la structure des documents,
- d'extraction d'informations,
- d'organisation d'unités issues de documents (sous-documents, citations, sur-documents).

Chapitre II Traitements de la forme et de la structure

Dans ce chapitre, nous nous intéressons aux traitements de la forme et de la structure de documents, visant à les préparer à des traitements linguistiques. En particulier, nous abordons le problème de la segmentation, normalisation et structuration de documents (parfois multiples) en différents formats.

II.1 Traitement de documents en vue de la TA et de la PE

Cette section décrit les différentes étapes des traitements de documents, en particulier en vue de la traduction automatique et de la post-édition.

II.1.1 Position du problème et état de l'art

Il existe une multitude de logiciels de traduction automatique. Certains (GOOGLE TRANSLATE[59], BING TRANSLATE[60], PROMT[61], etc.) proposent au moins une partie de leurs services gratuitement via une interface Web, tandis que d'autres sont des logiciels qui s'installent sur une machine et sont utilisés localement.

Ces logiciels diffèrent par le type et la taille des documents qu'ils prennent en entrée, ainsi que par la taille des unités de traduction qu'ils sont capables de traiter. Certains (GT, BING, les systèmes basés sur MOSES…) ont historiquement travaillé au niveau de la phrase, en traitant les phrases indépendamment les unes des autres. D'autres (PROMT, par exemple) peuvent faire des traitements sur plusieurs phrases (résolution d'anaphores, par exemple).

Les systèmes diffèrent également par le format des unités de traduction. GOOGLE TRANSLATE est capable de traiter le format HTML, alors qu'un système MOSES de base travaille sur du texte brut (cependant, des scripts comme MOSESWEB sont disponibles pour traiter du HTML).

Nous sommes amené à utiliser différents systèmes de TA, ce qui implique de pouvoir adapter le document à traduire aux différentes exigences de ces systèmes. Cela inclut la segmentation en unités de traduction de taille optimale, la normalisation, et la conversion vers les formats d'entrée les plus adaptés.

Une phase supplémentaire nécessaire est l'identification de la langue. Les textes que nous sommes amené à traiter contiennent souvent des passages en langues différentes. Ils peuvent être à l'intérieur d'un segment (phrase ou titre), ou bien c'est toute l'unité qui change de langue.

La normalisation implique de ramener l'unité à une représentation standard acceptable pour tel ou tel système.

II.1.1.1 Systèmes de TA

A mesure que les documents à traduire passaient du simple format texte brut à des documents balisés, les systèmes de TA ont été adaptés à traiter ce type de documents complexes.

Une des motivations de cela est la nécessité de préserver dans la traduction la mise en page du document source. Aujourd'hui, les systèmes comme SYSTRAN[62], GOOGLE TRANSLATE et BING (entre autres) sont capables de le faire.

[59] https://translate.google.fr
[60] https://www.bing.com/translator/
[61] http://www.online-translator.com
[62] http://www.systransoft.com/

SYSTRAN crée un flux XML à partir d'un document balisé, en plusieurs passes de traitement, incluant la segmentation, la tokénisation (ou mieux, « itémisation »), la normalisation des tokens et la reconnaissance d'entités nommées. Voici un exemple, pris dans (Sénellart et al., 2003).

Texte en entrée	`A $100 wrist-watch.`
	`<html><hr>A <b>$100</b> wrist-watch</html>`

Première représentation en XML	`<?xml version="1.0"?>`
	`<document original_format="html">`
	`<tag>&<lt;hr></tag><par id="1">A <ts face="bold">$100</ts> wrist-watch.</par>`
	`</document>`

Texte tokénisé après détection d'éntités	`[...]`
	`<par id="p1" xml:lang="en">`
	`<token type="word" capit="first" id="t1">A</token>`
	`<ts face="bold"><entity type="monval" id="t2">$100</entity></ts>`
	`<token type="word" norm="wrist-watch" id="t3">wristwatch</token>`
	`<token type="punct" id="t4">.</token>`
	`</par>`

Tableau 1 : exemple de flux XML de Systran construit itérativement

Une autre illustration de cela est l'ensemble de scripts MOSESWEB[63], conçus pour permettre aux systèmes MOSES de traiter des pages HTML. Ces scripts segmentent le texte en phrases à l'aide de quelques règles heuristiques basées sur les ponctuations, et normalisent les balises HTML en les remplaçant par des occurrences spéciales parenthétiques (MOSESOPENTAG1, MOSESCLOSETAG1, MOSESOPENTAG2, MOSESCLOSETAG2…). Ces scripts extraient également les valeurs textuelles des attributs rencontrés dans les balises.

II.1.1.2 Traitements généraux de conversion entre formats et traitements plus spécifiques

Il faut distinguer entre les traitements généraux de conversion entre formats et les traitements plus spécifiques.

Les traitements généraux peuvent être plus ou moins élaborés :

- Conversion de WORD en HTML, souvent directement par une API, par exemple celle de MICROSOFT.
- Conversion de POWERPOINT (PPT ou PPTX) en HTML.
- Conversion de PDF en HTML (beaucoup plus difficile).
- Conversion de PDF-IMAGE en DOC ou HTML, nécessaire quand il y a des textes et des formules dans des images, comme des manuels d'informatique.

Pour le THAM (les systèmes de type TRADOS[64], TM, DEJA VU[65], TRANSIT[66]), on est amené à mettre en œuvre des traitements plus spécifiques, par exemple pour trouver des phrases voisines d'une phrase, on peut être amené à passer par des « classes », c'est-à-dire des représentations normalisées dans lesquelles des instances de dates, nombres, catégories ont été substitués par des occurrences spéciales génériques. Cette approche est décrite dans la thèse d'E. Planas (Planas, 1998) sur les mémoires de traductions à étages.

[63] http://www.statmt.org/moses/?n=Moses.WebTranslation
[64] http://www.sdltrados.com/products/trados-studio/
[65] https://atril.com/
[66] http://www.star-ts.com/technology/translation-memory-transit-nxt/

II.1.1.3 **Autres applications spécifiques**

La segmentation peut être guidée par un alignement d'un texte avec une représentation formalisée de son contenu. Par exemple, Cong-Phap Huyhn a segmenté 25 articles de EOLSS[67] à l'aide de fichiers UNL[68] « compagnons » qui contenaient la représentation UNL de chaque phrase.

```
[S:01.029]
{org:en}
Although there are differences between countries, people who live in an
area located at easy access to aquatic environments consume more fish
than people without this
facility.
{/org}
{unl}
agt(live(agt>thing):1R,people(icl>person):1G.@topic.@pl)
agt(consume(agt>thing,obj>thing):3I.@entry,people(icl>person):1G.@topic.@
pl)
aoj(easy(icl>not difficult(aoj>thing)):2I,access(icl>opportunity):2N)
aoj(aquatic(icl>connected with
water(aoj>thing)):2X,environment(icl>natural world):35.@pl)
aoj(exist(aoj>thing):0H,difference(icl>way):0L.@pl)
aoj(located(aoj>thing):27,area(icl>place):22.@indef)
aoj(without(aoj>thing,obj>thing):4C,people(icl>person):45.@pl)
aoj(between(aoj>thing,obj>thing):0X,difference(icl>way):0L.@pl)
bas(fish(icl>animal):3V,people(icl>person):45.@pl)
con(consume(agt>thing,obj>thing):3I.@entry,although(obj>thing):02)
mod(facility(icl>equipment):4P,this(mod<thing):4K)
obj(between(aoj>thing,obj>thing):0X,country(icl>region):15.@pl)
obj(consume(agt>thing,obj>thing):3I.@entry,fish(icl>animal):3V)
obj(without(aoj>thing,obj>thing):4C,facility(icl>equipment):4P)
obj(access(icl>opportunity):2N,environment(icl>natural world):35.@pl)
obj(although(obj>thing):02,exist(aoj>thing):0H)
plc(live(agt>thing):1R,area(icl>place):22.@indef)
qua(fish(icl>animal):3V,more(bas>thing,qua<thing):3Q)
scn(located(aoj>thing):27,access(icl>opportunity):2N)

{/unl}
[/S]
```

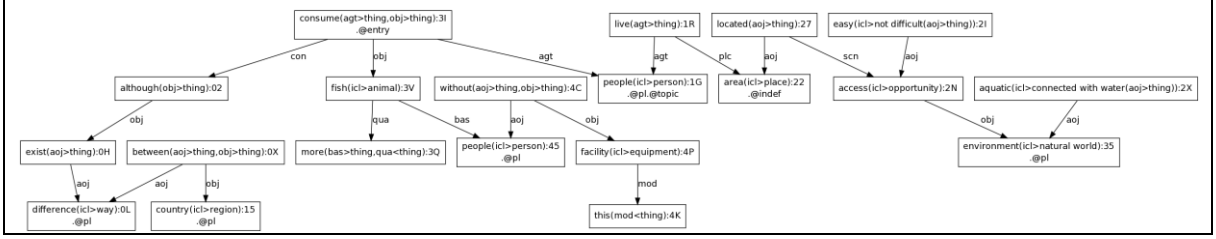


Figure 11 : exemple d'entrée pour un segment source anglais du corpus EOLSS

La segmentation intervient aussi dans le traitement de dialogues multilingues et autres documents complexes. On peut prendre l'exemple des dialogues interprétés dans ERIMM (Fafiotte et Boitet, 2003). Pour deux langues, on a au moins 3 documents (composés chacun de plusieurs parties, comme métadonnées, fichier son `.wav`, et éventuellement transcription écrite).

Enfin, nous traitons l'exemple de courriels relatifs à des projets, organisés en un graphe de citations.

[67] http://www.undlfoundation.org/eolss/index.htm
[68] http://www.undlfoundation.org/undlfoundation/

II.1.2 Éléments de modélisation

Dans cette section, nous décrirons la modélisation de documents usuels et des unités qui les composent : les segments, les fragments, les éléments de structuration. Nous parlerons également de la modélisation et du traitement de documents plus complexes.

II.1.2.1 Document usuel

Nous nous intéressons d'abord aux formats de documents qui cherchent à préserver le contenu aussi bien que la forme du document. Nous ne nous intéressons donc pas ici à des formats apparentés au PDF, dont la vocation est de préserver la présentation typographique du document, la conservation du contenu étant secondaire.

Un document usuel est généralement composé d'un fichier texte maître, et d'un ensemble de fichiers satellites.

Par exemple, un fichier DOCX est une archive contenant un fichier XML avec le contenu textuel structuré, ainsi qu'un ensemble de fichiers contenant des feuilles de style, les figures, et même des notes en bas de page.

La situation est analogue lorsqu'on exporte un tel document vers le format HTML : nous obtenons une page principale, et des fichiers satellites dans un répertoire.

En TEX, il est courant d'avoir un fichier de bibliographie séparé du texte principal.

Il est aussi possible qu'un document logique soit divisé en une hiérarchie de plus petits fichiers. C'est souvent le cas avec des documents TEX, ainsi qu'avec des livres en HTML, divisés en fichiers individuels correspondant aux chapitres ou aux sections.

Les documents peuvent être accompagnés d'annotations, qui ne font normalement pas partie du document, mais servent à l'étude, à l'indexation ou à la traduction (cas d'annotations par des graphes UNL).

II.1.2.1.1 Segments

Dans cette thèse, nous opérons avec la définition de segment donnée dans la thèse de Cong-Phap Huyhn (Huyhn, 2008) :

Définition 1 : *un* ***segment*** *est l'unité de base des traducteurs humains. Il s'agit d'une phrase, d'un titre, ou d'un terme dans une nomenclature.*

En effet, on cherche à découper le texte à traiter en plus petites unités autonomes et faisant sens.

Cependant, on peut considérer des (super)segments qui contiennent plusieurs phrases, voire plusieurs paragraphes. La notion de segmentation n'est donc pas limitée à l'identification de frontières entre phrases.

Il est possible de rencontrer des segments qui contiennent des mots ou fragments en langues différentes[69]. Nous voulions utiliser le terme « segment multilingue » pour décrire ce type de textes, mais ce terme existe déjà et est défini différemment par Cong-Phap Huyhn dans sa thèse, comme un ensemble de N segments équivalents en N langues, et non pas comme un segment qui contient des fragments en langues différentes. Ce type de segments se rencontre dans les documents multilingues tels que les brevets où chaque traduction fait foi. Nous allons donc parler plutôt de segment à bascule de code (*code switching*).

Définition 2 : *un* ***segment à bascule de code*** *est un segment qui contient des fragments dans au moins deux langues.*

[69] Par exemple, on met souvent entre parenthèses l'équivalent anglais d'un terme français, ou réciproquement.

Définition 3 : *un* ***segment multilingue*** *est un ensemble de segments monolingues exactes traductions l'un de l'autre.*

II.1.2.1.2 Fragments

D'après la thèse de C.-Ph. Huyhn, un fragment est « une sous-chaîne (connexe) d'un segment ». Nous étendons cette définition en permettant d'appeler fragment des images et des balises.

II.1.2.1.3 Éléments de structuration et de présentation (balises…)

Les éléments de structuration et de présentation peuvent être des balises souvent parenthétiques et verbeuses (cas des langages apparentés au XML), de balises « légères » (MARKDOWN[70]) ou des instructions (LATEX, RTF, …). Ces éléments se trouvent souvent à l'intérieur des unités logiques de traitement linguistique (segments, paragraphes, etc.) et doivent être pris en compte lors des traitements.

II.1.2.1.4 Difficultés inévitables

Les difficultés que l'on rencontre en traitant des documents usuels peuvent avoir des causes linguistiques et structurelles.

Une des difficultés est la *gestion des encodages*. Cette difficulté est moindre aujourd'hui, car UTF-8 est devenu la norme, mais jusqu'à récemment il existait une multitude d'encodages pour des langues telles que le russe ou le japonais, et la reconnaissance automatique de l'encodage, lorsqu'il n'est pas spécifié, n'est pas parfaitement fiable.

Une autre est la *multiplicité de formats*. Nous avons à traiter du texte brut, mais également des formats balisés, comme le HTML.

Une autre est la *gestion des balises de formatage*. Dès qu'on veut traiter des formats balisés, on est confronté à la gestion des balises, qui doivent être repérées et manipulées correctement par le système de segmentation. Or il peut y avoir un conflit entre les segments linguistiques et les fragments balisés, avec chevauchement.

Ceci est une première phrase. Ceci est une deuxième phrase.

Figure 12 : exemple de chevauchement de balises sur une frontière de phrases

Il est donc parfois nécessaire de procéder à un rebalisage afin que chaque segment soit valide du point du format traité.

Ceci est une première phrase. Ceci est une deuxième phrase.

Figure 13 : exemple de rebalisage qui rend chaque segment valide du point de vue de HTML

Une autre est la difficulté de segmentation multiniveau. Il y a des ambiguïtés pour la division en segments, ainsi que des conflits possibles entre balisages et segments linguistiques. Dans le cas des listes à puces, par exemple, une phrase qui amorce une liste peut ne pas être complète d'un point de vue linguistique, et de même pour les éléments de la liste, comme illustré dans l'exemple ci-dessous :

Pouvez-vous m'envoyer : -Votre numéro de passeport -La date de fin de validité -Votre date de naissance

Tableau 2 : exemple de structure énumérative

Cet exemple est assez élémentaire, car la liste n'est qu'à un niveau, et les éléments de la liste ne contiennent pas plusieurs phrases. Les segmenteurs usuels produiront un segment par ligne

[70] https://daringfireball.net/projects/markdown/

dans cet exemple ; or, pour une analyse linguistique il faudrait produire trois phrases, en collant la phrase d'amorce à chacun des éléments. Mais pour cela, il faudrait savoir repérer ce genre de structures énumératives.

Il y a aussi des phénomènes de récursion (citations dans le texte, incises autonomes, notes en bas de page, explications dans des balises, etc.). Voici un exemple de texte contenu dans un attribut alt en html :

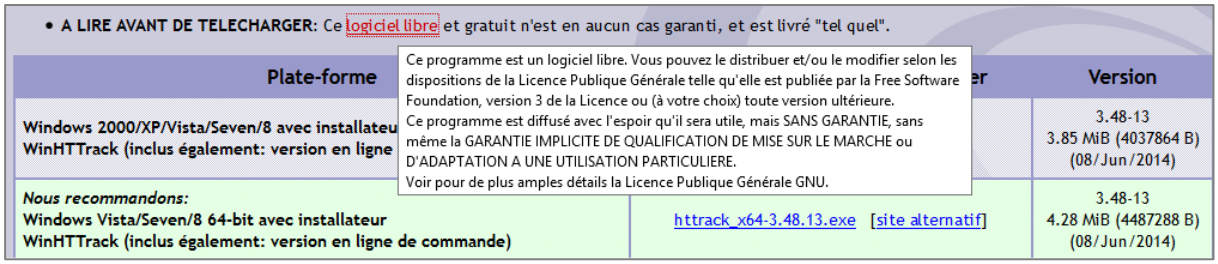


Figure 14 : exemple de texte récursif contenu dans l'attribut `alt` d'une balise en HTML

II.1.2.2 **Document complexe**

Nous parlerons ici de documents plus complexes, tels que les courriels, et de leur structure compositionnelle. Nous introduisons les notions de sur-document et de sous-document, pour décrire la composition hiérarchique de documents qu'on rencontre souvent.

II.1.2.2.1 Sur-document

Définition 4 : *un* ***sur-document*** *est un document qui contient un ou plusieurs autres documents.*

Par exemple, une édition des actes d'un colloque est un sur-document qui contient les articles individuels comme sous-documents. Elle a des éléments qui lui sont propres, comme l'introduction, la table des matières, les numéros de page globaux, etc.

Dans le cas de courriels, un fichier qui contient plusieurs courriels imbriqués est un sur-document pour chacun de ces courriels, qui à leur tour, sont des sur-documents pour les courriels plus bas dans la hiérarchie.

On peut aussi prendre l'exemple d'un document de cours généré à partir de fragments de documents. Dans ce cas, c'est un sur-document qui contient des fragments qui sont des sous-documents non seulement de ce cours mais aussi des documents individuels dont ils sont issus.

II.1.2.2.2 Sous-document

Définition 5 : *un* ***sous-document*** *est un document ou une partie (non nécessairement connexe) d'un document qui peut être inclus dans un autre document.*

Par exemple, les paragraphes 2 et 4 d'un document peuvent constituer un sous-document d'un document où ils sont cités, en principe dans le même ordre, de façon connexe ou non.

Un sous-document peut lui-même être un sur-document pour d'autres documents, comme un article scientifique en LaTeX qui importe une bibliographie, et qui à son tour est importé par l'édition des actes de colloques.

Enfin, chaque document est constitué de segments et fragments, qui sont eux aussi des sous-documents.

II.1.2.2.3 Modélisations possibles des pages Web

Il est courant de rencontrer des pages Web dont la structure générale est statique, mais dont certaines parties sont de nature à évoluer. Un exemple de cela est la page principale d'une université, qui contient généralement une section « Actualités » qui est modifiée fréquemment.

Pour pouvoir modéliser ce type de documents nous utilisons la notion de pseudodocument de (Huynh, 2010).

II.1.2.3 Représentations possibles

Au niveau logique, il est clair que la structuration hiérarchique des documents, ainsi que les références croisées entre eux, décrivent une structure réticulaire, ou en graphe.

Du point de vue de la représentation concrète, on est amené à manipuler des structures linéaires (de type flux XML ou graphe en XML ou…)

II.1.2.4 État de l'art de la segmentation

Il existe deux approches à la segmentation de texte en phrases : l'approche par règles et l'approche par apprentissage automatique. La première consiste à définir des heuristiques décrivant les fins de phrases potentielles. La deuxième utilise des corpus annotés manuellement pour entraîner le segmenteur. Nous décrivons ici quelques outils de segmentation représentatifs de ces deux approches et en donnons une première évaluation.

Un exemple notable de l'approche par règles est le standard SRX[71] (*Segmentation Rules eXchange*), défini par l'association LISA pour permettre à la communauté du TALN de créer et échanger des règles de segmentation. Un fichier de règles SRX est un fichier XML contenant des expressions régulières décrivant des fins de phrase potentielles. Lors de la segmentation, les règles sont appliquées en cascade à chaque position dans le texte, jusqu'à ce qu'une des expressions coïncide avec le patron de la règle.

`<rule break="no">`
`  <beforebreak>\b[Mm]lle\.</beforebreak>`
`<afterbreak>\s</afterbreak>`
`</rule>`
	Règle indiquant que si la position courante est précédée par "Mlle." ou "mlle." et est suivie par un caractère d'espace, alors ce n'est pas une frontière de phrase. Autrement dit, une phrase ne peut se terminer par "Mlle." ou "mlle."

Figure 15 : exemple commenté de règle SRX

Le standard permet de préciser des paramètres de segmentation dans l'en-tête du fichier, par exemple d'indiquer si l'on doit segmenter les attributs `TITLE` et `ALT` des balises, ou si l'on inclut ou non les balises enveloppant un segment textuel dans le segment, ce qui implique que les SRX peuvent être utilisés pour segmenter du HTML. Cependant, l'utilisation des SRX requiert un programme-moteur qui les applique. Il est du ressort du moteur de gérer le HTML et de tenir compte des paramètres de segmentation.

```
Cette phrase contient la citation "Première phrase. Deuxième phrase." et les
SRX ne peuvent la segmenter correctement.

Segment 1 : Cette phrase contient la citation "Première phrase.

Segment 2 : Deuxième phrase." et les SRX ne peuvent la segmenter correctement.
```

Figure 16 : exemple de texte que les SRX ne peuvent segmenter correctement sans traitement de la récursivité

Une autre approche par règles est utilisée par LINGPIPE[72], un ensemble d'outils pour le TALN. LINGPIPE permet au développeur de définir quels symboles seront des frontières potentielles, quelles chaînes ne peuvent précéder une fin de phrase (typiquement, une phrase ne

[71] http://okapiframework.org/wiki/index.php?title=SRX
[72] http://alias-i.com/lingpipe/

peut pas se terminer par "Mr."), et quels débuts de phrase sont impossibles (une parenthèse fermante par exemple). Ces trois ensembles constituent la base de règles pour la segmentation.

Un apprentissage automatique supervisé est implémenté par l'outil SPLITTA (Gillick, 2009), qui propose des classificateurs SVM ou bayésiens naïfs initialement entraînés sur les corpus du WALL STREET JOURNAL (Marcus et al., 1993) et sur le BROWN CORPUS (Francis et Kucera, 1979), qui sont censés être représentatifs de l'anglais. Les problèmes de cette approche sont les suivants :
(1) il est nécessaire de refaire le travail d'annotation pour chaque nouvelle langue, mais aussi pour chaque nouveau genre de texte (littérature médicale, romans, etc.).

(2) cette approche semble peu adaptée à la segmentation de texte balisé.

(3) il n'y a pas de standard d'échange de données d'entraînement.

Voici une brève évaluation de ces outils. Pour des raisons techniques, nous n'avons pas pu tester l'outil PUNKT. Pour les SRX, nous utilisons le moteur SRX d'OKAPI et deux fichiers SRX, l'un provenant d'OKAPI et l'autre de SWORDFISH III, couvrant au total 20 langues.

LINGPIPE dispose de règles pour deux types de texte : les textes biomédicaux et les articles du WALL STREET JOURNAL. Nous incluons également le segmenteur de GOOGLE TRANSLATE que nous utilisons implicitement dans nos IMAG.

Voici un récapitulatif des systèmes que nous avons évalués.

Outil	**Approche à la segmentation**	**Langues disponibles**	**Formats en entrée**	**Sorties**
Okapi-SRX	Règles	20 langues dont l'anglais et le français	Texte brut, HTML	Texte brut, HTML
Splitta	Apprentissage supervisé	Anglais	Texte brut	Texte brut
LingPipe	Règles	Anglais	Texte brut, XML	XML
Google Translate	Inconnu	71 langues dont l'anglais et le français	Texte brut, HTML, XML, doc(x), odt, pdf	HTML

Tableau 3 : différents segmenteurs

Dans cette première expérience, pour avoir une idée de la performance des systèmes dans le meilleur des cas, nous avons pris l'article anglais de WIKIPEDIA sur la complexité calculatoire, et l'avons mis sous forme de texte brut, contenant 241 phrases.

Cet article a l'avantage de contenir une quantité suffisante de texte, et aussi d'être riche en formules et autres éléments susceptibles de causer des erreurs de segmentation, comme les citations ou les numéros de références entre crochets, qui ont été laissés dans le texte.

Outil	**Vrais positifs**	**Faux positifs**	**Faux négatifs**	**Précision**	**Rappel**	**F-mesure**
SRX-Swordfish	190	36	51	0,84	0,79	0,81
SRX-Okapi	194	19	47	0,91	0,80	0,85
Splitta-SVM	192	22	49	0,90	0,80	0,85
Splitta-NB	201	19	40	0,91	0,83	0,87
LingPipe-Méd	198	19	43	0,91	0,82	0,86
LingPipe-News	173	31	68	0,85	0,72	0,78
Google Translate	186	25	55	0,88	0,77	0,82

Tableau 4 : performance des segmenteurs testés

En analysant les erreurs, on s'aperçoit que, comme prévu, les SRX ne peuvent tenir compte de l'équilibre des parenthèses dans les structures parenthésées. On voit aussi qu'aucun des systèmes n'est capable de segmenter les phrases qui se terminent par les références WIKIPEDIA, comme celle-ci :

> *The time and memory consumption of these alternate models may vary.[1] What all these models have in common is that the machines operate deterministically.*

ce qui cause le plus grand nombre d'erreurs.

- GOOGLE TRANSLATE, SPLITTA et les SRX ne détectent pas la fin de phrase lorsqu'elle se termine par une seule lettre, comme ici :

  > *If a problem X is in C and hard for C, then X is said to be complete for C. This means that X is the hardest problem in C.*

  sans doute parce que "C." est interprété comme une abréviation.

- De même, lorsqu'une phrase se termine par "no.", LINGPIPE l'interprète comme abréviation de "number", et ne détecte pas la frontière.
- Les SRX d'OKAPI ne coupent pas entre deux phrases si la deuxième commence par une parenthèse.
- On remarque aussi que GOOGLE TRANSLATE normalise certaines occurrences dans les segments, transformant par exemple "e.g." et "i.e." en "eg" et "ie".

II.1.3 Conception, implémentation et évaluation de SegNorm

II.1.3.1 Conception

II.1.3.1.1 Ambitions par rapport aux outils et méthodes existants

En premier lieu, nous voulons que notre segmenteur puisse traiter une multitude de langues. Cela implique d'avoir des règles de segmentation, car des corpus d'apprentissage ne sont pas disponibles pour un grand nombre de langues. Cela implique aussi de savoir identifier automatiquement la langue et l'encodage d'un texte[73].

Nous voulons que le segmenteur puisse opérer aussi bien sur du texte brut que sur des documents balisés (a minima en HTML). Il faut donc que l'outil sache extraire le texte d'un document balisé tout en préservant la structure de ce document. Ainsi, nous voulons produire des fichiers-squelettes, qui conservent les balises, mais où les segments ont été substitués par des identifiants.

La multiplicité de formats d'entrée implique une multiplicité de formats de sortie, aussi bien pour les fichiers-squelettes que pour les fichiers contenant les segments. Nous voulons, par exemple, pouvoir produire des fichiers XML avec les segments dotés d'identifiants et chaînés, mais aussi du texte brut, avec un segment par ligne.

Lorsque plusieurs segmentations sont possibles, nous voudrions construire un graphe de ces segmentations.

II.1.3.1.2 Segmentation multiple et récursive

La multiplicité de la segmentation mérite une mention à part entière. Puisque les traitements subséquents à la segmentation peuvent opérer sur des unités allant au-delà de la phrase, nous voudrions pouvoir produire des segmentations en unités de taille paramétrable. Ainsi, un texte

[73] Ces problèmes ont été étudiés par Vo-Trung Hung dans sa thèse soutenue en 2004 à l'INP-G.

serait segmenté en paragraphes ou fragments, et ces unités en phrases, avec des liens de chaînage entre elles.

Lorsqu'un document ou un texte présentent de la récursivité, nous voudrions la traiter. Ainsi, le segmenteur doit pouvoir identifier les segments récursifs, en extraire le sous-texte, le segmenter et le relier au segment englobant.

II.1.3.1.3 Normalisation (selon tel ou tel système)

Quel que soit le système qui traitera le texte segmenté, on vise à identifier les éléments qu'on sait a priori gêner l'analyse : les balises, les formules mathématiques, dont l'identification en amont de la segmentation est nécessaire, les émoticons, etc.

Ces éléments doivent être substitués par des occurrences spéciales qu'on aura définies, et un fichier-dictionnaire avec les correspondances entre ces occurrences et les entités qu'elles représentent sera construit. Nous obtiendrons ainsi une représentation normalisée, à partir de laquelle nous saurons générer des représentations correspondant aux formats d'entrée de tel ou tel système de traitement.

II.1.3.1.4 Représentations visées

En représentation en mémoire, nous visons un graphe d'unités liées par des liens de chaînage séquentiel. Ainsi, pour chaque unité, il sera possible de consulter son élément englobant, son prédécesseur, son suivant, sa forme brute et ses formes normalisées.

La représentation sur disque sous sa forme complète sera sous forme de fichier XML contenant les différentes unités chaînées via des références aux identifiants uniques dont chaque unité sera dotée. Une représentation simplifiée sous forme de fichier texte avec un segment par ligne (éventuellement numéroté) pourra aussi être générée.

On produira également un ou plusieurs fichiers-squelette, avec différentes « résolutions » de tailles des segments.

Enfin, puisque nous normalisons le texte segmenté, nous produirons un fichier-dictionnaire avec l'association occurrence-entité.

II.1.3.2 **Implémentation**

L'implémentation de l'outil, appelé SEGNORM, a été réalisée en JAVA. Ce choix a été guidé par le souci de portabilité, par la disponibilité d'une grande quantité de bibliothèques utiles, et par la facilité d'intégration avec nos autres outils, dont certains ont également été développés en Java.

L'outil s'appuie essentiellement sur des bibliothèques open-source, dont des outils d'APACHE, la suite OKAPI, et d'autres.

SEGNORM se lance en ligne de commande. Il prend un certain nombre de paramètres en entrée de ligne, mais aussi dans des fichiers de paramétrage. L'Annexe 1 donne un extrait d'un de ces fichiers.

En sortie, SEGNORM produit un fichier contenant les segments (en XML ou en texte) et un fichier squelette. Si on a demandé une normalisation, un fichier normalisé et un fichier-dictionnaire des occurrences sont également produits. L'Annexe 2 donne des exemples de sortie.

La segmentation du HTML peut être guidée par les balises, et peut produire deux résultats :

- une segmentation selon les balises "dures" qui délimitent des sections dans le document (balises `<p>` et `<div>`, par exemple), effectuant ainsi un découpage en paragraphes.

- une segmentation en phrases. Un fichier de paramètres permet de spécifier les balises à considérer comme "dures", ainsi que les balises à ignorer, dont le contenu ne doit pas être segmenté, par exemple la balise `<code>`.

La segmentation en phrases s'effectue par défaut à l'aide du moteur SRX de la suite OKAPI. Nous disposons de règles SRX pour 20 langues.

Puisque ce moteur ne traite pas le format HTML et que les SRX sont définies pour du texte brut, nous effectuons un « débalisage » et une normalisation du texte à segmenter. Nous extrayons également les valeurs des attributs pouvant contenir du texte (`title`, `alt`, ...) et les segmentons récursivement. Les balises enlevées et leurs positions dans le texte sont mémorisées. La normalisation consiste aussi à remplacer les entités HTML (telles que `&amp;` ou ` `) par leur équivalents texte. La position de ces éléments est également mémorisée.

Le texte brut ainsi obtenu est fourni au moteur de segmentation SRX, qui produit une liste de segments, où l'on réintroduit les éléments enlevés lors de la normalisation. L'utilisateur peut spécifier si, dans le cas où il y avait des balises qui enveloppaient un segment, elles doivent être incluses dans le segment ou pas.

Cette transformation réversible en texte brut permet d'utiliser d'autres systèmes de segmentation qui n'opèrent que sur ce type de données.

SEGNORM est conçu pour être très paramétrable, grâce à trois fichiers de paramétrage. On peut ainsi définir les listes de balises XML qui tombent dans chaque catégorie (dure, molle, à ignorer), et contrôler la longueur des fragments à envoyer aux systèmes de TA, qui ont tous une limite (par exemple 250 segments ou 30000 caractères pour GOOGLE TRANSLATE). Ils permettent également de spécifier les banques de règles à utiliser pour telle ou telle langue, des options de normalisation, et d'autres options.

On peut aussi contrôler le reparenthésage nécessaire pour que les segments « découpés » (par exemple, au milieu d'un `<span>` ou même d'une zone en gras `<b>`) soient en html bien formé.

SEGNORM est rapide sur une machine de milieu de gamme : le traitement d'un fichier HTML de 500 Ko prend environ 1 seconde.

II.1.3.3 **Déploiement**

Le code de SEGNORM est actuellement sur la forge du LIG (scm.forge.imag.fr). Il y 13 classes propres à SEGNORM (10 pour le segmenteur et 3 pour le normaliseur), et 50 venant d'OKAPI et d'autres bibliothèques.

II.2 Traitement d'échanges par courriel

A cause de la complexité des courriels, nous avons dû développer dans SEGNORM des modules spécifiques au traitement des courriels. Pour cela, nous avons proposé une modélisation originale, et implémenté des traitements spécifiques, que nous avons évalués, en les comparant à NLTK et à LINGPIPE, sur le français et l'anglais.

II.2.1 Modélisation

Après avoir décrit les éléments spécifiques des courriels tels qu'on les rencontre, nous en proposons une modélisation en « microstructure » (celle d'un courriel pris isolément) et « macrostructure » (en graphe d'inclusion et chaînage de fragments).

II.2.1.1 Éléments spécifiques

Les courriels que nous avons à traiter se présentent sous forme de fichiers EML. Ces fichiers contiennent plusieurs sections : ensemble de métadonnées, et texte du courriel (souvent encodé en QUOTED-PRINTABLE ou en BASE64). Cette section peut contenir du texte brut, du HTML ou les deux. À la fin, on trouve éventuellement des pièces attachées, encodées en général en BASE64. Ces sections sont séparées par une chaîne séparatrice unique, définie dans les métadonnées.

D'un point de vue logique, un courriel est un élément d'une conversation à deux ou à plusieurs. Chaque courriel possède un expéditeur et un ou plusieurs destinataires, qui peuvent être des destinataires directs, en copie ou en copie cachée. Ce sont donc les participants de la conversation.

Définition 6 : *les* ***participants d'une conversation*** *par courriel sont l'ensemble des personnes figurant comme expéditeur ou destinataire d'au moins un courriel de la conversation.*

Chaque courriel possède un identifiant unique (MESSAGE-ID), qu'on trouve dans les métadonnées. Lorsqu'un courriel est envoyé à plusieurs personnes, chaque personne reçoit une copie du courriel. Un courriel peut être une réponse à un courriel précédent ou un transfert. Dans ces deux cas, la plupart du temps le courriel cite un courriel précédent. La plupart du temps aussi, les clients de messagerie incluent des références aux courriels précédents dans les champs IN-REPLY-TO et REFERENCES des métadonnées. À l'aide de ces champs, on peut reconstruire un fil de discussion et les relations entre les messages.

Enfin, puisqu'un message peut en citer un autre, on peut retrouver des fragments des textes cités et nouveaux entremêlés. Cela est décrit plus en détail dans la section suivante.

II.2.1.2 Microstructure

Dans cette section, nous décrivons les structures typiques du texte d'un courriel. Une des difficultés du traitement de courriels vient du fait qu'un courriel peut en citer un autre. Ces citations peuvent se trouver en-dessous du corps de texte principal, ou bien peuvent être entremêlés avec ce texte. L'exemple ci-dessous illustre un cas relativement simple d'entremêlement.

Définition 7 : *la* ***microstructure*** *d'un ensemble de courriels, et plus généralement de documents, est la description formalisée des structures possibles des documents de cet ensemble.*

```
Sujet : Re: Dossier
De : Michel Lefèvre
Date : 16/01/2011 23:04
Pour : Yohann Treuillot <ytreuillot@gmail.fr>
Bonjour,
On 16 January 2016 at 13:27, Yohann Treuillot <ytreuillot@gmail.fr>
wrote:
> Bonjour, pourriez-vous faire suivre le dossier aux personnes
concernées ?
C'est fait.
Cordialement,
Michel
> Merci !
> Cdlt,
> Yohann Treuillot
Michel Lefèvre - PhD
Assistant Professor, HDR
Laboratoire ABCD, Equipe EFGH  - Tel: (+33) 4 00 00 00 00
```

Figure 17 : exemple de deux messages entremêlés

Nous voyons ici deux messages courts avec des fragments entremêlés. Nous reconnaissons deux en-têtes, et le message cité, qu'on identifie à l'aide des chevrons de citation. Le niveau d'imbrication ici est 1, mais on rencontre très souvent des messages avec des imbrications plus profondes. Pour un traitement correct de ce type de message, il faut identifier les en-têtes et les portions citées, en essayant d'identifier correctement le niveau de citation.

II.2.1.3 **Macrostructure**

Définition 8 : *la* ***macrostructure*** *d'un ensemble de documents pouvant contenir des sous-documents est la description formalisée des relations entre documents et sous-documents.*

Les messages individuels s'organisent en fils de discussion. Ces fils forment une structure arborescente et peuvent être reconstruits à l'aide d'attributs souvent présents dans les métadonnées des fichiers EML. Voici un exemple de discussion reconstruite par MOZILLA THUNDERBIRD[74].

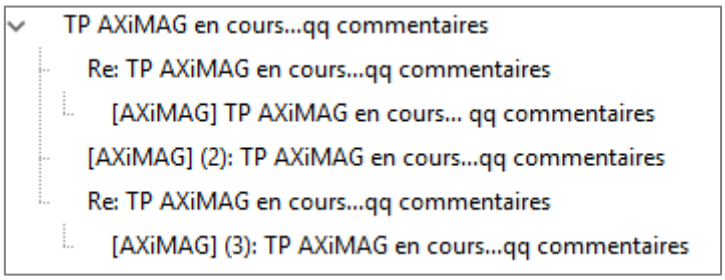


Figure 18 : exemple de fils de discussion reconstruits par MOZILLA THUNDERBIRD

De plus, un message peut citer des fragments d'autres messages, présents ou non dans le fil de discussion. Ainsi, on obtient un graphe de citations, qui représente en quelque sorte la structure discursive d'une conversation.

En reprenant l'exemple des deux messages entremêlés ci-dessus, un graphe de citation pourrait être le suivant :

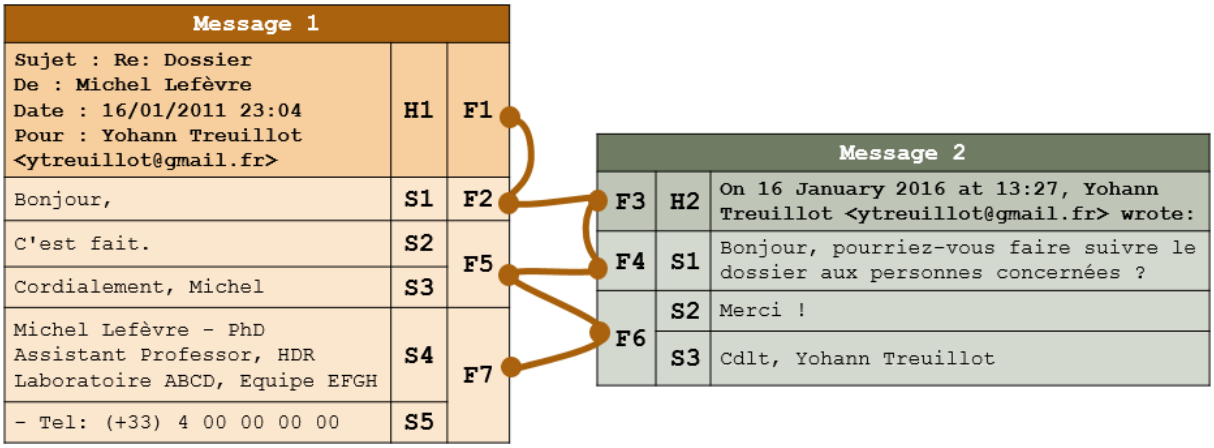


Figure 19 : exemple de graphe de citation pour deux messages entremêlés

Dans cet exemple, nous voyons les courriels individuels reconstruits et des fragments numérotés F1, F2, etc., chaînés par des arcs représentant leur séquence dans le courriel exemple.

II.2.2 Traitement de fichiers contenant des conversations

Après un bref un état de l'art, nous décrivons le traitement structurel de fichiers EML et des conversations.

[74] https://www.mozilla.org/fr/thunderbird/

II.2.2.1 État de l'art

Divers travaux sur le traitement structurel des courriels ont été publiés au fil des années. Les travaux sur la reconstruction des fils de discussion en l'absence de métadonnées se basent sur une mesure de similarité entre messages cités et messages originaux (Yeh & Harnly 2006, Dehghani et al., 2013).

Des travaux sur la reconnaissance de la structure des textes des messages ont été menés par plusieurs équipes.

- (Carvalho et Cohen, 2004) entraînent un classifieur à identifier les lignes correspondant à des signatures et des zones citées dans les messages.
- (Lampert et al., 2009) vont plus loin et entraînent un classifieur à identifier dans les messages les « zones fonctionnelles », à savoir les salutations, les phrases de fin de courriel, les signatures, les textes ajoutés automatiquement, et les portions citées.
- (Couillaut et al., 2014) utilisent une grammaire développée à l'aide de l'outil GRAMLAB[75] (semblable à NOOJ[76]) pour repérer les amorces de reprise, c'est-à-dire les portions citées dans les messages.
- (Hernandez et Salim, 2014) utilisent les marqueurs de citation dans des paires de courriels pour pré-annoter les fragments avec des annotations discursives. Ils exploitent le fait que des courriels sont fragmentés par les personnes qui y répondent.

En conclusion, il y a eu des travaux sur le traitement des courriels, mais il existe peu d'outils opérationnels mis à la disposition des chercheurs.

II.2.2.2 Segmentation en courriels individuels

Le problème qu'on cherche ici à résoudre est de segmenter des courriels en fragments de différents niveaux de granularité, en vue de les préparer à l'extraction d'informations à partir du texte. En particulier, nous visons à identifier les tâches mentionnées dans les énoncés constituant ces courriels, ainsi que leurs caractéristiques et leurs relations, ainsi que le type pragmatique de ces énoncés (annonce, demande, rappel, etc.).

Dans la plupart des outils de messagerie, lorsqu'on répond à un courriel, ce courriel est placé en citation en dessous de la réponse qu'on rédige. Nous devons traiter ce type de suites de messages, et cherchons à les découper en messages individuels, puis en phrases.

II.2.2.2.1 Repérage des en-têtes

Pour segmenter une conversation en messages individuels, nous devons nous baser sur des marqueurs de séparation entre les sous-segments de ces messages. Un de ces marqueurs est le caractère de citation '>'. Un autre est l'en-tête (sur plusieurs lignes).

Le rôle d'un en-tête est, au minimum, de rappeler l'expéditeur du message qui le suit. Dans la plupart des cas, les en-têtes indiquent aussi la date du message, et, dans le cas des en-têtes multiligne, l'objet et les destinataires.

L'en-tête est un marqueur important de séparation entre messages car, d'une part, le symbole de citation '>' n'est pas toujours présent dans les conversations, et d'autre part, des informations importantes et utiles pour la suite des traitements peuvent être extraites des en-têtes.

Nous distinguons deux types d'en-têtes : multiligne et monoligne. Dans l'exemple 1 ci-dessus, nous avons un aperçu de chacun. L'exemple 2 ci-dessous montre 4 en-têtes multiligne (anonymisés) pour en illustrer la variété linguistique et structurelle.

[75] http://unitexgramlab.org/fr
[76] http://www.nooj-association.org/index.php

Da : FR BORDEAUX Online [mailto :donotreply@carlsonwagonlit.com] **Inviato** : Martedì 12 gennaio 2016 09 :15 **A** : Jean Dupont <jean.dupont@mail.fr> **Cc** : service@carlsonwagonlit.com **Oggetto** : Confirmation/Invoice Réservation Hôtel **Importanza** : Grande
Sender : messages-noreply@bounce.linkedin.com **From** : John Doe via LinkedIn <member@linkedin.com> **Reply**-To : John Doe <John.Doe@gmail.com> **To** : My Name <My.Name@gmail.com>
Von : Pierre Dupond [mailto :Pierre.Dupond@mail.de] **Im Auftrag von** Jean Dupont **Gesendet** : Mittwoch, 12. Oktober 2016 20 :55 **An** : Int'l E-learning Association <elearning-list@ielas.com>; elearning@ehub.no Betreff : Shared task proposition
Sujet : Acte2i : votre état des lieux (mission n◦28544) **Date** de renvoi : Tue, 2 Jun 2015 09 :58 :38 -0700 **De** (renvoi) : jdupont@gmail.fr **Pour** (renvoi) : jean.dupont@mail.fr **Date** : Tue, 02 Jun 2015 18 :58 :34 +0200 **De** : Acte2i <nepasrepondre@acte2i.com> **Pour** : jdupont@gmail.fr

Tableau 5 : 4 en-têtes de structures et de langues différentes

Souvent, des champs sémantiquement identiques possèdent différents libellés dans une langue donnée, selon le client de courriel qui les génère. Par exemple, le champ '`Date`' peut aussi apparaître comme '`Envoyé`', ou bien '`Sujet`' peut être '`Objet`'. L'ordre et le nombre des champs varie considérablement : rien que pour le français, nous avons identifié 17 structures d'en-tête différentes (cf. Annexe 3). Le format des dates présente aussi une certaine variabilité.

Le Tableau 6 ci-dessous présente des exemples d'en-têtes monoligne observés dans notre corpus. Ici aussi, nous observons la variété linguistique et structurelle de ces en-têtes : présence ou non d'une « adressielle »[77] , d'un format de date (lorsqu'elle est présente), etc.

Il giorno 21 ott 2015, alle ore 14 :46, Jean Dupont <jean@mail.fr> ha scritto :
Вторник, 19 января 2016, 12 :03 +01 :00 от Jean D. <jean@mail.fr> :
Am 22.02.2016 um 13 :44 schrieb Jean Dupont :
10/26/2015 12 :18 PM(e)an, Jean Dupont igorleak idatzi zuen :
Le 06.03.2008 09 :38, le perspicace Jean DUPONT s'exprimait en ces termes :
Jean Dupont wrote :

Tableau 6 : exemples d'en-têtes monoligne

Pour détecter les en-têtes des messages quelles que soient leurs langues et leurs structures, nous devons identifier des traits discriminants linguistiquement invariants.

Nous repérons les en-têtes à l'aide d'expressions régulières couplées à une recherche de traits. Les expressions en elles-mêmes sont relativement peu discriminantes, et servent seulement à trouver des zones candidates.

Pour les en-têtes multiligne, on cherche des blocs de trois à sept lignes, chaque ligne commençant par un à sept mots, suivis du caractère ':', suivi lui-même par d'autres mots et enfin par un passage à la ligne.

[77] Nous utilisons ce néologisme bien formé, de préférence à « adresse de courriel », trop lourd, et à l'horriblement mal formé « adresse de mél » — en effet, la phonétique et l'orthographe françaises interdisent absolument une terminaison en « -él ». C'est d'ailleurs pour cela qu'on a les termes « péritel », « minitel », et pas *péritél, *minitél.

Nous associons un score, valant 0 au départ, à chacun de ces blocs. Ce score est calculé en cherchant la présence :

- d'une année ou d'une heure (les mois étant souvent écrits avec des mots, cette information dépend donc de la langue) ;
- d'une adressielle ;
- d'un changement de niveau d'imbrication (nombre de caractères '>' commençant une ligne) entre la ligne qui précède le bloc et la ligne qui le suit ;
- de noms de personnes, en détectant des suites de mots commençant par des majuscules;
- de chaînes « `re :` » ou « `fwd :` », indicatrices d'un champ « `objet` » ;
- d'une ligne immédiatement précédente contenant une séquence de '—', comme dans «–– `Message original` ––».

Pour chaque trait présent, nous incrémentons le score de 1. Si le score atteint un certain seuil (2 ou plus), nous considérons que la zone est bien un en-tête. Nous présentons dans la partie 3 une évaluation quantitative de cette approche pour le repérage.

Il est difficile de s'abstraire entièrement de la langue, car les deux derniers traits ne sont pas tout à fait indépendants de la langue : il n'y a pas de distinction entre majuscules et minuscules dans les scripts coréen, japonais et chinois, et nous ne savons pas si « `re` : » et « `fwd` : » sont des préfixes universels.

Pour les en-têtes monoligne, nous cherchons des lignes contenant une année ou une heure, et qui se terminent par un ' :'.

II.2.2.2.2 Démêlage de fragments enchevêtrés

La segmentation en messages individuels est guidée par le repérage des en-têtes des messages et par les caractères d'imbrication.

Dans les lignes du texte qui commencent par des caractères d'imbrication ('>', mais parfois aussi des tabulations ou des '|'), nous associons un niveau à chaque ligne, correspondant au nombre de ces caractères d'imbrication en début de ligne. On part de l'hypothèse qu'entre deux en-têtes toutes les lignes d'un même niveau appartiennent au même message. Cette hypothèse est largement vérifiée dans les corpus que nous avons collectés.

Ensuite, les suites de lignes au même niveau sont groupées dans des fragments. Ainsi, on passe d'une séquence de lignes à une séquence de fragments. À chaque fragment est associé un niveau, et chaque niveau est associé à l'en-tête précédent (ou parfois suivant, lorsqu'il n'y a pas d'ambiguïté) le plus proche.

Une fois les fragments construits, on associe à chacun un lien vers son prédécesseur et un lien vers son successeur. On obtient un chaînage de fragments, qui peuvent alors être séparés en messages individuels.

Reprenons l'illustration précédente :

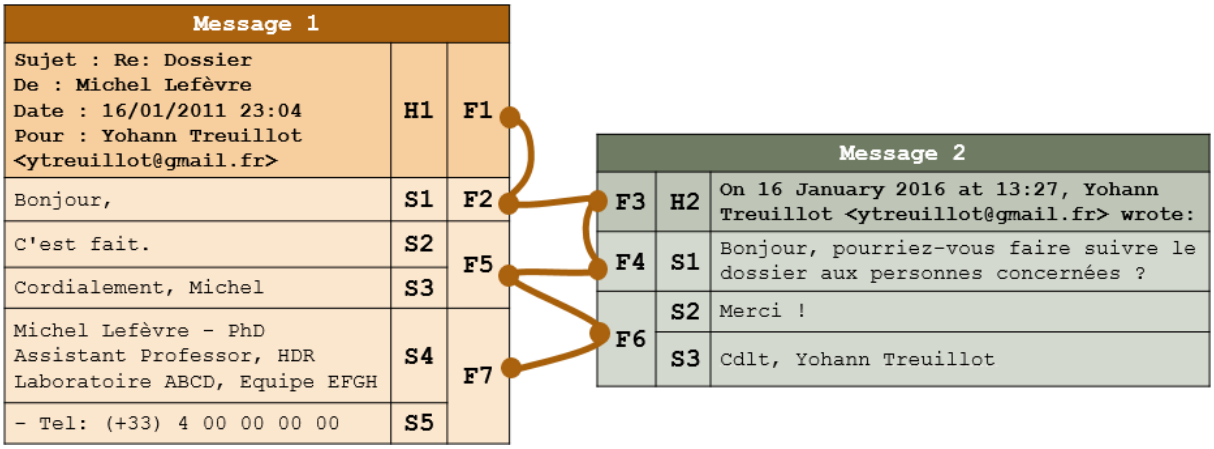


Figure 20 : illustration du démêlage de deux messages entremêlés

Nous voyons dans la Figure 20 deux messages reconstruits, chapeautés par des en-têtes, identifiés par H1 et H2.

On y voit aussi une numérotation des fragments qui correspond à leur séquence d'occurrence dans le courriel entremêlé.

De plus, chaque fragment a été segmenté en phrases, ce qui est représenté par les identifiants de segments S1, S2, etc.

II.2.2.3 Normalisation

Avant d'appliquer les règles de segmentation, nous passons par une étape de normalisation afin de minimiser les erreurs.

À cette étape, on remplace les émoticônes (*smileys*) par des hors-textes spéciaux, car la ponctuation contenue dans les émoticônes peut causer une fausse segmentation.

On détecte aussi des phrases tronquées par des passages à la ligne : certains messages peuvent avoir été encodés en QUOTED-PRINTABLE[78] ou en MIME[79], des encodages qui limitent la longueur des lignes à 76 ou 78 caractères, insérant des passages à la ligne en cas de dépassement.

Le passage à la ligne étant un marqueur potentiel de fin de phrase, nous voulons éliminer ceux qui n'en sont pas. Par conséquent si une ligne est d'une longueur comprise entre 75 et 78 (sans compter les caractères de citation) et ne se termine pas par une ponctuation de fin de phrase, alors nous remplaçons le caractère de passage à la ligne par une espace.

II.2.2.3 **Segmentation en phrases**

II.2.2.3.1 Méthode

Nous utilisons ici aussi une approche par règles SRX[80].

Une fois que les messages ont été identifiés, et « démêlés » si nécessaire, nous effectuons la segmentation en phrases, en trois étapes.

(1) D'abord, nous identifions les paragraphes dans le texte, séparés par des passages à la ligne. (2) Ces paragraphes sont soumis à l'identification de la langue par l'outil APACHE TIKA[81]. Cela nous permet de choisir le bon jeu de règles de segmentation en phrases.

[78] https://tools.ietf.org/html/rfc2045
[79] https://tools.ietf.org/html/rfc2822
[80] *Segmentation Rules eXchange* : https://www.gala-global.org/srx-20-april-7-2008
[81] https://tika.apache.org/

(3) Chaque paragraphe est segmenté en phrases en utilisant des règles SRX.

Lors de la segmentation, les règles sont appliquées en cascade à chaque position dans le texte, jusqu'à ce qu'une des expressions concorde.

II.2.2.3.2 Exemples commentés de résultats

La figure ci-dessous montre le résultat de la segmentation du message de la Figure 17, en messages individuels et en phrases. C'est en fait une représentation XML du graphe représenté dans la Figure 20.

Nous voyons que certaines informations, comme le préfixe indicatif de citation, ont été mémorisées dans l'attribut `citPrefix`.

L'attribut `nextId` permet de lier chaque fragment à son successeur dans le graphe.

Les segments extraits sont stockés isolément, et sont de plus rattachés par l'attribut `id` aux fragments dont ils sont issus.

```
<Message id="0">
    <MessageHeader>
        <Fragment citPrefix="" id="C1M0F0" nextId="C1M0F1">
        Sujet : Re: Dossier
        De : Michel Lefèvre
        Date : 16/01/2011 23:04
        Pour : Yohann Treuillot &lt;ytreuillot@gmail.fr&gt;</Fragment>
    </MessageHeader>
<MessageBody>
    <Fragment citPrefix="" id="C1M0F1" lang="fr" nextId="C1M1F2">
    Bonjour,</Fragment>
    <Fragment citPrefix="" id="C1M0F4" lang="fr" nextId="C1M1F5">
    C'est fait.
    Cordialement,
    Michel</Fragment>
    <Fragment citPrefix="" id="C1M0F6" lang="it" nextId="-1">
    Michel Lefèvre - PhD
    Assistant Professor, HDR
    Laboratoire ABCD, Equipe EFGH  - Tel: (+33) 4 00 00 00
    00</Fragment>
</MessageBody>
<Segments>
    <Segment id="C1M0F1S0" lang="fr">Bonjour,</Segment>
    <Segment id="C1M0F4S1" lang="fr">C'est fait.</Segment>
    <Segment id="C1M0F4S2" lang="fr">Cordialement, Michel</Segment>
    <Segment id="C1M0F6S3" lang="fr">Michel Lefèvre - PhD Assistant
    Professor, HDR Laboratoire ABCD, Equipe EFGH</Segment>
    <Segment id="C1M0F6S4" lang="fr">- Tel:(+33) 4 00 00 00
    00</Segment>
</Segments>
</Message>
```

```
<Message id="1">
    <MessageHeader>
        <Fragment citPrefix="" id="C1M1F2" nextId="C1M1F3">
        On 16 January 2016 at 13:27, Yohann Treuillot
        &lt;ytreuillot@gmail.fr&gt; wrote:
        </Fragment>
    </MessageHeader>
<MessageBody>
    <Fragment citPrefix="&gt; " id="C1M1F3" lang="fr"
    nextId="C1M0F4">
    Bonjour, pourriez-vous faire suivre le dossier aux personnes
    concernées ?
    </Fragment>
    <Fragment citPrefix="&gt; " id="C1M1F5" lang="fr"
    nextId="C1M0F6">
    Merci !
    Cdlt,
    Yohann Treuillot
    </Fragment>
</MessageBody>
<Segments>
    <Segment id="C1M1F3S0" lang="fr">Bonjour, pourriez-vous
    faire suivre le dossier aux personnes concernées ?</Segment>
    <Segment id="C1M1F5S1" lang="fr">Merci !</Segment>
    <Segment id="C1M1F5S2" lang="fr">Cdlt, Yohann
    Treuillot</Segment>
</Segments>
</Message>
```

Figure 21 : fichier XML avec courriels identifiés et fragments chaînés

Nous nous sommes rapidement aperçu de l'intérêt de reconnaître les signatures dans les messages. En effet, les signatures peuvent contenir non seulement des noms, des adressielles, mais aussi des phrases qu'il faut éviter d'analyser car elles peuvent induire des erreurs d'analyse.

Nous avons repris l'outil CIRANDA[82] de (Cohen et Carvalho, 2004), qui utilise un perceptron entraîné à classifier des séquences de lignes d'un courriel pour y identifier les signatures.

CIRANDA caractérise chaque ligne par un ensemble de traits que l'on trouve dans ce type de zones, comme la présence de mots capitalisés, de grades académiques, de numéros de téléphone, etc.

CIRANDA prend également en compte la position de la ligne et les traits des lignes précédente et suivante.

[82] https://www.cs.cmu.edu/%7Evitor/codeAndData.html

À partir de ces informations, CIRANDA associe un score à chaque ligne. Un score positif indique l'appartenance de la ligne à une signature.

Les modèles ont été entraînés sur des courriels en anglais du début des années 2000. Nous avons étendu CIRANDA en y ajoutant l'extraction de traits que l'on rencontre dans les signatures des courriels en français, tels que des mots relatifs aux adresses physiques (cours, rue, chemin, etc.), des grades académiques (HDR, Prof…), des lieux (laboratoire, université, école, etc.) et des numéros de téléphone au standard français.

Voici le résultat de l'identification de signature dans un des courriels de l'exemple précédent.

Score Ciranda	Ligne du courriel
-6431502.0	`Bonjour,`
-7905192.0	`C'est fait.`
-6822303.0	`Cordialement,`
-3456937.0	`Michel`
1.7669746E7	`Michel Lefèvre - PhD`
2.1788574E7	`Assistant Professor, HDR`
1.6358615E7	`Laboratoire ABCD, Equipe EFGH - Tel: (+33) 4 00 00 00 00`

Figure 22 : exemple d'identification de signatures avec un Ciranda adapté au français

II.2.2.3.3 Rattachement des fragments ou segments de courriels identifiés aux courriels originaux

Nous pouvons utiliser les informations issues des métadonnées des fichiers EML pour structurer un ensemble de fichiers en fils de discussion. En effet, les fichiers EML incluent souvent des références aux messages précédents dans la conversation, à l'aide d'identifiants uniques de courriels. Nous utilisons ces identifiants afin d'associer à chaque identifiant mentionné dans les métadonnées d'un fichier EML tous les messages qui y font référence.

On obtient ainsi des structures arborescentes représentant les fils de discussion. Voici un exemple de conversations reconstruites.

```
Tue Aug 17 15:42:08 CEST 2010 [AXiMAG] étendre les langues  (malais en particulier)?
-<1E6B98A4-AA55-4D1F-AD2A-AFE491FE2DDD@imag.fr>
--Wed Aug 18 16:31:16 CEST 2010 [AXiMAG] merci: ajout de malais, thai, indonésien
--Wed Aug 18 17:43:25 CEST 2010 [AXiMAG] appel de Systran, + annonce contact Chai (SysSTeC-EBMT)

<a06240810c7fede3eld83@[192.168.0.101]>
-<8EC5388B-28E2-4BD5-B35E-2E2992666074@imag.fr>
--<4BD95BCC.80609@free.fr>
---Thu Apr 29 15:27:53 CEST 2010 Re: [AXiMAG] iMAG-lametro: quelques points à améliorer
----Fri Apr 30 04:37:53 CEST 2010 [AXiMAG] iMAG-v2: Bugzilla, CCH, partenariats,  démos liglab et lametro

<g2ge0f39f1310040209091j9c7c3e0cy560d30a2d487df75@mail.gmail.com>
-<BCAC818A-BFCE-4E39-83EC-F2FBC8CCCF98@imag.fr>
--<85D8950B-AF06-44FA-B462-4A79DB2CF5CB@imag.fr>
---<1BBA0717-7E9E-41DB-8A36-2C1EE938694D@imag.fr>
----Sat Apr 03 12:50:38 CEST 2010 Re: [Urgent] Le problème de spam à SECTra_w
----Sat Apr 03 17:26:25 CEST 2010 [Urgent] Le problème de spam à SECTra_w : proposition d'actions
-----Sat Apr 03 21:15:23 CEST 2010 Re: [Urgent] Le problème de spam à SECTra_w : proposition  d'actions
------Mon Apr 05 15:05:25 CEST 2010 [Urgent]  Problème de spam à SECTra_w +/- réglé  , et Text&Tone
```

Figure 23 : illustration de reconstruction de fils de discussion à partir d'une banque de fichiers EML

Dans cette sortie produite par notre module `ThreadStructureRecognizer`, nous voyons trois conversations.

Lorsque nous disposons d'un fichier EML pour un message, nous représentons ce message par une date et son objet. Lorsque nous savons qu'un message a existé (puisque son identifiant est cité dans les métadonnées d'un ou plusieurs autres messages), mais que nous n'avons pas trouvé

de ficher EML correspondant, nous le représentons par son identifiant, et nous le reconstruisons à partir du témoin le plus proche (temporellement) dans le fil dont nous disposons, en appliquant le segmenteur en messages individuels décrit dans ce chapitre.

Voici un diagramme des traitements effectués par SEGNORM pour les courriels.

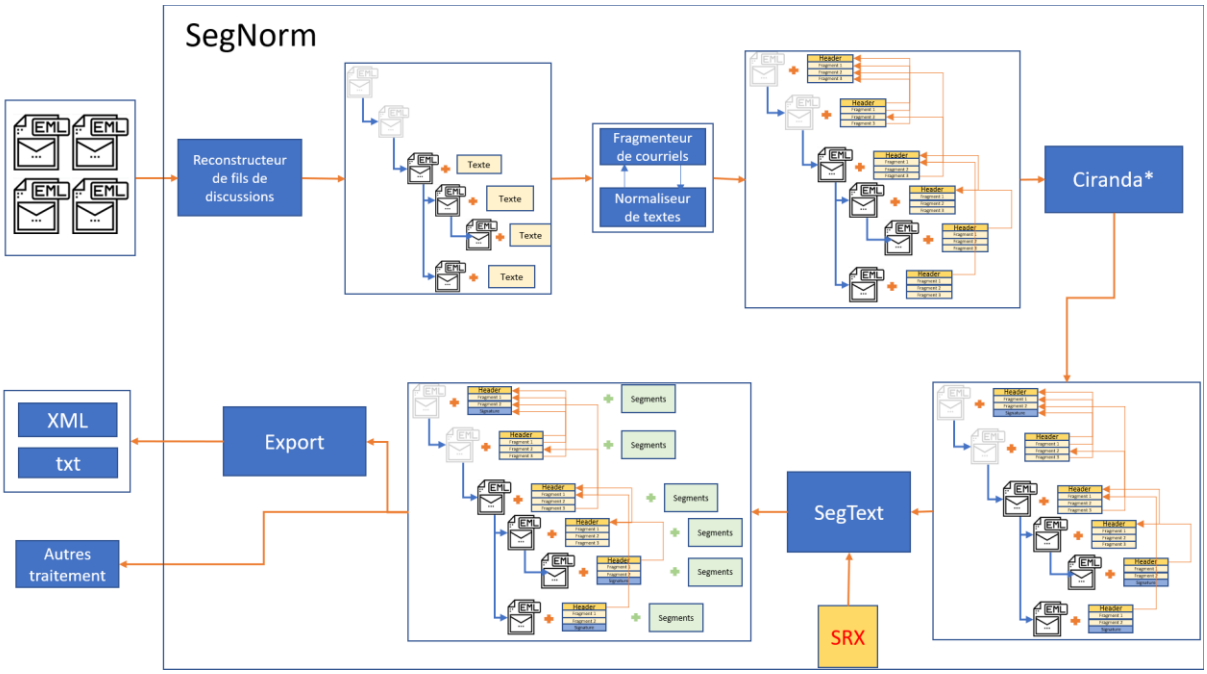


Figure 24 : diagramme des traitements effectués par SEGNORM pour les courriels

II.2.3 Évaluation

II.2.3.1 Critères et méthode

Passons maintenant à l'évaluation des outils de segmentation qui ont été développés pour le traitement de collections de courriels.

Notre but est ici d'évaluer l'identification des en-têtes, la reconstruction des messages individuels, et la segmentation de courriels en phrases.

Pour préparer cette évaluation, nous avons annoté structurellement environ 500 courriels pour identifier les en-têtes et les frontières de phrases.

II.2.3.2 Évaluation subjective

Nous avons cherché à trouver une « bonne » méthode d'évaluation subjective, c'est à dire une méthode faisant appel à des jugements humains. Malheureusement, en essayant nous-même de le faire, nous avons constaté que nous n'arrivions pas à évaluer la segmentation et la structuration d'un message indépendamment de notre compréhension globale de ce message.

En effet, une segmentation incorrecte ne nous empêche souvent pas de comprendre un message, et peut ou non entraîner des problèmes pour la suite des traitements, selon leur nature.

C'est pourquoi nous nous sommes limité à des évaluations objectives.

II.2.3.3 Évaluation objective

Nous avons constitué un corpus de correspondances dans lequel nous avons manuellement identifié 495 en-têtes. En appliquant nos règles de repérage, sur 495 en-têtes annotées, 453 ont été correctement repérées. 25 fragments de texte ont été incorrectement classés comme en-têtes.

	Précision	Rappel	F-mesure
Repérage d'en-têtes	95%	91%	93%

Tableau 7 : résultats pour le repérage des en-têtes dans notre corpus

La plupart des faux négatifs pour les en-têtes multiligne étaient dus à des déformations induites par le passage du message par plusieurs clients de courriel. Cela a parfois pour effet d'ajouter des passages à la ligne, et parfois, au contraire, d'en supprimer. Dans tous les cas, le formatage sur lequel se basent nos heuristiques de repérage de blocs avait été altéré.

Nous avons évalué la segmentation en phrases sur un corpus de 6994 segments issus de notre corpus, majoritairement en français (279 segments en anglais), et sur un corpus contenant 1000 segments issu du corpus EnronSent, tous en anglais. Pour cela nous avons aussi mesuré le nombre de frontières de phrases correctement identifiés.

Nous avons évalué notre outil en regard de NLTK (Bird et al., 2009) et de LingPipe (Alias-I, 2008). Le texte utilisé est celui des messages, sans les caractères de citation, et sans autre normalisation (chaque outil est supposé faire ensuite sa propre normalisation).

	Corpus-FR			EnronSent		
	Précision	**Rappel**	**F-mesure**	**Précision**	**Rappel**	**F-mesure**
SegDoc	85%	83%	84%	83%	81%	82%
NLTK	83%	81%	82%	87%	84%	85%
LingPipe	81%	77%	79%	88%	80%	84%

Tableau 8 : résultats pour le repérage de frontières entre phrases

Nous voyons dans le Tableau 8 que les trois systèmes produisent des performances comparables, avec un léger avantage pour SegNorm pour le français, et pour NLTK pour l'anglais.

Dans leur évaluation de neuf systèmes de segmentation de texte anglais non balisé (Read et al., 2012) ont montré que, d'une part, aucun outil de segmentation n'est fiable à 100 %, et que, d'autre part, les performances relatives des systèmes de segmentation sur un corpus donné n'étaient pas toujours constatées sur un corpus différent, y compris après normalisation des corpus et adaptation des segmenteurs.

Bien qu'aujourd'hui SegNorm utilise des SRX comme outil principal de segmentation en phrases, il est conçu pour pouvoir facilement intégrer d'autres outils de segmentation.

Synthèse et perspective

Nous avons présenté un outil de segmentation et de normalisation de fichiers en différents formats. Il a d'abord été conçu pour les textes bruts et balisés à la XML (et était alors appelé SegDoc), puis il a été étendu aux courriels pour segmenter des conversations imbriquées en messages individuels avec fragments chaînés. Cet outil, que nous appelons désormais SegNorm, intègre également une reconnaissance de signatures basée sur l'approche de (Cohen et Carvalho, 2004).

SegNorm est multilingue, extensible et paramétrable. Ses performances en segmentation en segments linguistiques (phrases ou titres) sont au niveau de l'état de l'art. Pour les autres tâches de segmentation, nous n'avons pas d'outil auquel le comparer.

Après la thèse, nous nous proposons d'étendre SegNorm en définissant un vrai langage formel permettant de décrire des segmentations multiniveau et récursives.

Chapitre III Repérage et traitement de courriels relatifs à des tâches

Dans ce chapitre, nous nous intéressons à l'extraction d'informations à partir de courriels, et plus précisément des informations relatives aux tâches. Nous passons en revue ce qui existe dans le domaine, et proposons une modélisation des tâches.

Cette modélisation nous a servi de guide pour annoter un corpus de courriels relatifs à des projets en informatique. Nous décrivons les expériences réalisées à l'aide de ce corpus annoté pour le repérage et typage de tâches et leurs attributs.

III.1 Position du problème et état de l'art

L'extraction de tâches à partir d'échanges de courriels professionnels se limite pour l'instant à repérer les énoncés qui concernent des tâches.

Pour arriver à de l'extraction puis de la gestion de tâches, nous proposons une caractérisation des énoncés parlant de tâches, et des tâches elles-mêmes, organisée comme une hiérarchie d'attributs. Nous l'avons utilisée pour annoter un corpus important de courriels liés à deux projets.

Nous travaillons d'abord sur l'identification des tâches dont il est question dans les courriels au moyen de techniques d'apprentissage automatique regroupant les énoncés parlant de la même tâche.

Au-delà de l'identification, nous cherchons alors, toujours en utilisant ce corpus annoté, à trouver une méthode efficace pour extraire l'information, c'est à dire pour trouver les valeurs des attributs des énoncés et des tâches, ainsi que les relations entre les tâches.

III.1.1 Notions et modèle de tâche dans notre contexte

Nous expliquons ici les raisons de notre intérêt pour les tâches dans les courriels.

III.1.1.1 Historique

L'idée directrice de ce travail était de doter l'outil SYNAPS d'un module sémantique capable de comprendre les intentions des utilisateurs et leur suggérer intelligemment des aides. Cette motivation est commune à de nombreux outils de gestion de projets, et formulée ainsi, est très vague.

Un plugin SYNAPS pour OUTLOOK avait été développé juste avant le début de cette thèse, et permettait d'importer des courriels dans la plate-forme. Nous avons constaté que la raison principale pour importer un courriel était d'en tirer des tâches à faire, et il a été décidé de développer un outil qui automatiserait le plus possible ces opérations.

III.1.1.2 Objectifs

Notre objectif dans ce chapitre est d'aider les participants à effectuer et suivre les tâches en extrayant un maximum d'informations relatives aux tâches dans les courriels, et en les mettant dans le contexte du projet en cours.

Précisons que, par *tâches*, nous n'entendons pas les tâches du point de vue des ergonomes, qui sont des actions articulatoires. Nous le précisons car, si on cherche « modèle de tâche » dans les moteurs de recherche, on trouve des travaux relatifs aux modèles de tâches articulatoires parmi les premiers résultats.

Il s'agit pour nous d'entités consistant en une ou plusieurs actions à réaliser par une ou plusieurs personnes, demandées par d'autres personnes, avec souvent des contraintes temporelles. D'autre part, les tâches peuvent avoir des propriétés de séquencement avec d'autres tâches.

III.1.2 Travaux antérieurs

III.1.2.1 Travaux consacrés aux différentes notions de tâche

En ce qui concerne les énoncés qui parlent de tâches, (Searle 1969) définit les actes de parole directifs comme ceux qui placent une obligation sur l'auditeur de faire quelque chose en réponse (par exemple les questions, les requêtes ou les invitations). Beaucoup d'actes directifs sont exprimés sous forme de questions (mais pas toutes, et toutes les questions ne sont pas des actes de parole directifs).

Dans la littérature en anglais, on retrouve différents termes pour désigner les énoncés des tâches. Paradoxalement, le terme *task* n'est pas aussi employé que, par exemple, *request* ou *action item*. On rencontre également le terme *assignment*.

Un autre terme est *commitment*, l'engagement. Dans (Kalia et al. 2013), ce terme peut à la fois désigner l'engagement exprimé par la personne qui s'engage (« d'accord, je le ferai »), mais aussi les demandes ou requêtes, catégorie de notre intérêt.

DAMSL (Allen et al. 1997), un formalisme d'annotation de dialogues inspiré par Searle, définit une catégorie d'actes de dialogue nommée action-directive. Cela correspond aux requêtes. Là où action-directive se distingue des actes de parole directifs de Searle est le fait que cette catégorie n'inclut pas les demandes d'information, pour lesquelles existe la catégorie *info-request*. À cela s'ajoute la catégorie open-option, qui, contrairement à action-directive, ne place pas d'obligation sur l'auditeur. Nous appelons cette catégorie *suggestion*.

Il y a eu depuis diverses modélisations liées à l'informatisation, en vue de la création de systèmes de dialogue intelligents pour les services vocaux, en particulier la SDRT[83] et la théorie des agents rationnels (cf. Le système NESTOR du CNET). Il y en a eu une utilisation directe (simplifiée) dans l'IF (Interface Format) du projet C-STAR de traduction de dialogues oraux finalisés.

Voici un exemple repris du mémoire de HDR d'Hervé Blanchon. L'agent est indiqué par `A:`, le client par `C:`, et le tour de parole contient un acte de parole (plus précisément, un acte de dialogue (comme `request-information`), un prédicat (comme `availability`), un argument, et des valeurs de diverses variables (comme le début et la durée d'une réservation, le nombre et le type des lits d'une chambre).

[83] *Segmented Discourse Representation Theory*, de Nicholas Asher et Alex Lascaridès. (http://homepages.inf.ed.ac.uk/alex/papers/iwcs4.pdf)

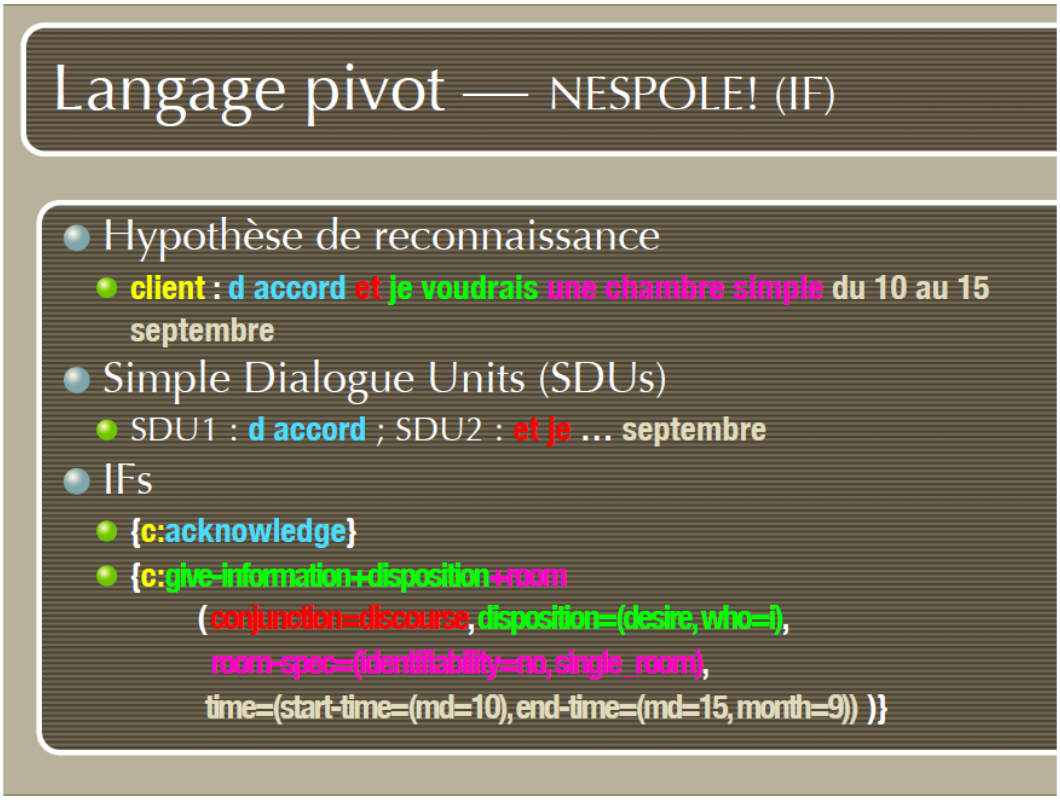


Figure 25 : exemple de description en IF d'un tour de parole dans le projet C-STAR de TA de parole

Du point de vue théorique, nous nous inscrivons donc dans la continuité de travaux relativement anciens, et d'autres plus récents, de nature souvent pluridisciplinaire, concernant le discours, le dialogue, l'ergonomie cognitive, et le traitement des langues.

Dans sa thèse, (Tomokiyo, 2000) a étudié de façon contrastive la relation entre les articulateurs du discours et les articulateurs du dialogue, sur des transcriptions de dialogues monolingues en plusieurs langues (japonais, français et anglais).

Du point de vue pratique, les recherches sur la pragmatique en TALN ont d'abord été motivées par le développement d'applications fondées sur du dialogue Homme-Machine, par exemple pour des interfaces avec des bases de données (système LUNAR de William Woods dès 1970), puis par la construction de systèmes plus actifs contenant explicitement la notion de tâche (suivi de projets, tableaux de bord, commande de robots).

C'est seulement ensuite qu'est apparue la nécessité de modéliser et de supporter des applications liées à des dialogues Homme-Homme, comme la traduction de dialogues oraux finalisés.

III.1.2.2 **Des travaux principalement pour l'anglais**

III.1.2.2.1 Courriels

Le repérage de tâches consiste à repérer des éléments textuels qui indiquent aux personnes auxquelles ils sont destinés d'effectuer une action. Ce repérage peut donc être vu comme une tâche de classification d'actes de parole. Des travaux ont déjà été réalisés dans ce domaine, mais hormis (Salim, Hernandez et Morin, 2016), jusqu'ici tous sur des courriels en anglais. En voici quelques-uns.

En 2004, le corpus ENRON (décrit ci-dessous dans le III.1.3) a été publié, ce qui a suscité un grand nombre de travaux.

(Cohen, Carvalho et al. 2004) ont construit un programme apprenant à classifier des courriels entiers en actes de parole.

Paul Bennett et Jaime Carbonnell (de CMU) ont travaillé durant plusieurs années sur le repérage d'*action items*[84] dans les courriels.

- Dans (Bennett et Carbonnell, 2005a), ils décrivent leurs expériences de classification automatique de courriels et de phrases isolées, à l'aide d'algorithmes d'apprentissage automatique (`k-NN`, `SVM`, *Naive Bayes*). Ainsi, dans un corpus de 744 courriels, dont 328 contiennent un ou plusieurs *action items*, ils identifient des courriels contenant un *action item* avec une F-mesure de 78% et des phrases avec une F-mesure de 67%, en utilisant des n-grammes comme traits.
- Dans (Bennett et Carbonnell, 2005b) (*Feature representation for effective action-item detection*), ils explorent en détail les traits utiles pour la classification de phrases et de courriels entiers, ainsi que les performances des différents classifieurs.
- Dans (Bennett et Carbonnell, 2007), ils décrivent une approche qui combine les sorties de plusieurs classifieurs avec un métaclassifieur `STRIVE` (*Stacked Reliability Indicator Variable Ensemble*) pour attribuer un rang aux courriels selon qu'ils contiennent ou non un *action item*. `STRIVE` produit un meilleur classement en termes de la mesure `AUC` (91%), mais produit une F-mesure légèrement en-dessous de la meilleure F-mesure pour la classification de phrases (78% vs 80% pour un classifieur *Naive Bayes*).

Andrew Lampert et ses collègues ont également travaillé durant plusieurs années sur le traitement de courriels pour l'extraction d'action et d'engagements.

- Dans (Lampert et Paris, 2007), ils s'intéressent à l'annotation de phrases dans les courriels (corpus ENRON), et à l'accord entre trois annotateurs pour les catégories *Request-for-action* (« demande ») et *Commitment-to-act* (« engagement »). Ils observent un kappa de 78% pour le premier et de 54% pour le second, et explorent les raisons des désaccords entre les annotateurs.
- Dans (Lampert et Paris, 2008), ils posent les bases théoriques des notions de *request* et *commitment*. Ils définissent formellement les *requests* comme un tuple `Request = ⟨Action, Requestor, Requestee, [Condition]⟩`, et les *commitments* comme `Committment = ⟨Action, Committor, Committee, [Condition]⟩`. Ils commencent également à décrire les variétés des réalisations de surface de ces entités, et de leur complexité. Ils explorent ces difficultés en détail dans (Lampert et al., 2008), que nous étudierons en détail plus loin dans ce chapitre, car nous avons aussi été confronté à certaines de ces difficultés.
- Dans (Lampert et al., 2009), ils décrivent la réalisation d'un plugin pour OUTLOOK (voir Figure 26) qui permet d'annoter manuellement des courriels comme contenant un *request* ou un *commitment*. L'annotation est possible au niveau du document entier ou au niveau du fragment de texte selectionné.
- Dans (Lampert et al., 2010), ils segmentent les messages en zones fonctionnelles (salutations, signature, etc.) et entraînent un classificateur SVM sur les n-grammes, ainsi que sur des traits de surface comme la présence de verbes modaux, la longueur des

[84] Souvent traduit incorrectement comme « élément d'action », ce terme, issu du domaine de la gestion, désigne un « évenement, tâche, activité ou action à effectuer ; une unité discrète d'action pouvant être faite par une seule personne » (https://en.wikipedia.org/wiki/Action_item). Cela correspond souvent à un élément d'une *todo list*.

messages, la présence de préfixes `FW:` ou `RE:` dans le sujet, et obtiennent un indice de Rand[85] de 84%.

- Ces travaux sont résumés dans la thèse d'Andrew Lampert (Lampert, 2014), intitulée *Making email actionable: the identification and use of obligation acts in workplace email*.

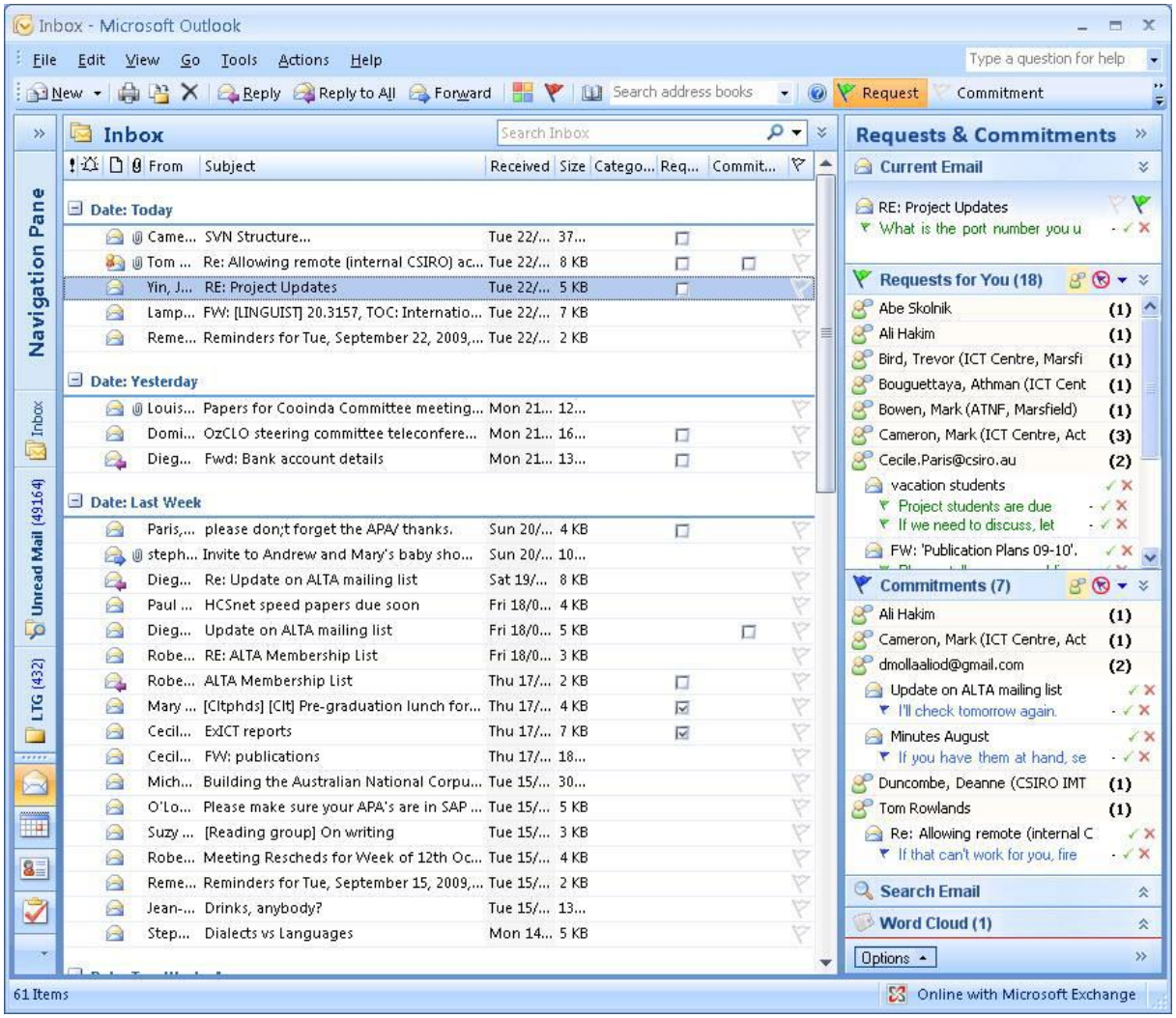


Figure 26 : image d'écran du plugin OUTLOOK développé par Lampert et al.

En utilisant des traits de surface et des traits linguistiques, (Corston-Oliver et al., 2008) identifient les messages qui contiennent des énoncés de tâches avec une précision et un rappel de 80%. Il s'agit donc là non pas de repérage de phrases ou de fragments textuels, mais de messages entiers.

(Jeong et al., 2009) étudient la classification semi-supervisée d'actes de dialogue dans les courriels et les forums. Ils utilisent 12 catégories, dont *action-motivator*, et obtiennent une micro F-mesure[86] de 79% pour l'ensemble des catégories.

[85] L'indice de Rand, aussi appelé *accuracy*, est une mesure du pourcentage de décisions correctes par un outil de classification. C'est le quotient de la somme des décisions correctes (vrais positifs + vrais négatifs) et de toutes les décisions (vrais positifs + vrais négatifs + faux positifs + faux négatifs). https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

[86] Une micro F-mesure est la moyenne harmonique de la micro-précision et du micro-rappel, et tient donc compte de la taille des classes (cf. http://damien.nouvels.net/cours/statscorp/06_Evaluation.pdf)

(Scerri et al. 2010) identifient quatre types d'action (*request, assign, suggest, deliver*[87]), chacun pouvant avoir pour catégorie d'objet des données ou des activités, et pour sujet l'émetteur, le destinataire ou les deux.

Ils définissent un ensemble de règles expertes qui portent sur la structure syntaxique de la phrase et la classification en actes de parole de Searle (1969), et obtiennent des résultats suffisants pour leurs besoins. Le Tableau 9 ci-dessous les résume.

Excellents	30%
Passables	11%
Pas tout à fait corrects	12%
Incorrects	19%
Non détectés	28%

Tableau 9 : résultats de Scerri et al. 2010

(Kalia et al. 2013) entraînent plusieurs classificateurs sur des traits issus d'une approche linguistique poussée, et proposent de détecter la création d'un engagement (*commitment*), le détachement, la décharge, l'annulation et la délégation. Ils obtiennent des F-mesures comprises entre 89% et 95% pour la détection de création d'engagements. Dans leur cas, les engagements incluent aussi bien des phrases impératives que des phrases de promesse (« nous le ferons »), les deux signalant l'apparition d'un engagement.

(Tavafi et al. 2013) décrivent leurs expériences de reconnaissance d'actes de dialogue dans des conversations synchrones (réunions, appels téléphoniques) et asynchrones (courriels, forums), en anglais. Ils utilisent des traits lexicaux (unigrammes, bigrammes) et des traits extralinguistiques (identifiant du locuteur, position de l'énoncé dans un message, longueur de l'énoncé. Dans leurs expériences, SVM-HMM obtient les meilleures performances

Le défunt PARAKWEET LABS[88] fournit un ensemble de 5000 segments décontextualisés[89] issus du corpus ENRON et annotés avec la présence d'*intent*, une catégorie qui combine les catégories *request*, *propose* et *commit* de (Cohen et Carvalho, 2004). Leurs expériences avec un classifieur SVM produisent une F-mesure de 78%.

Le seul travail que nous avons trouvé sur la classification automatique d'énoncés dans les courriels en français est celui de (Salim, Hernandez et Morin, 2016), qui s'intéressent à l'identification automatique des dimensions sémantiques et des fonctions communicatives dans des textes dans différentes modalités (courriel, forum, tchat).

En utilisant des traits lexicaux de surface tels que des n-grammes et des lemmes, et des courriels annotés par des catégories issues de DIT++ (Bunt, 2009) avec un SVM, ils obtiennent les performances suivantes sur la tâche d'identification d'actes de dialogue dans des courriels issus du corpus UBUNTU-FR.

Acte de dialogue	**Précision**	**Rappel**
Request for information	74%	62%
Request for action	41%	27%
Request for directive	50%	60%

Tableau 10 : résultats obtenus par (Salim et al., 2016) pour trois catégories d'actes de parole

[87] En français : demander, affecter, suggérer, fournir.
[88] http://www.parakweet.com/: cette entreprise de R&D visant le traitement intelligent des courriels a fermé durant notre thèse, en février 2016.
[89] https://github.com/ParakweetLabs/EmailIntentDataSet/wiki

Seuls (Kalia et al. 2013) cherchent à constituer une base de tâches qui est utilisée pour suivre les évolutions de leur état. Les règles expertes qu'ils ont développées ne se transposent pas facilement au français.
Aucun des travaux étudiés ne visait à déterminer si deux énoncés repérés concernent la même tâche sous-jacente. De plus, il n'y a pas de repérage d'attributs de la tâche qui irait au-delà du repérage des émetteurs et assignataires. On pense à des attributs tels que les contraintes temporelles, les justifications, les conditions, etc. À notre connaissance, il n'existe pas de corpus annoté avec les attributs des tâches. C'est pourquoi nous en avons créé un.

III.1.2.2.2 Réunions

Les réunions sont les principaux moments où les équipes discutent des tâches. Travailler sur les transcriptions de réunions est très difficile, car il faut non seulement disposer d'une excellente reconnaissance vocale, mais de plus les énoncés prononcés par les participants sont souvent fragmentés, hâchés[90] et bruités. Le tableau suivant illustre ces problèmes.

ID de l'énoncé	ID locuteur	*Timestamp*	Enoncé
#619ce63[...]	me003	180.372	[basically it just means build the tree .]
#13b584e[...]	me003	182.241	[and then it passes the tree onto uh - the ge- - the generation module .]
#da3e3f7[...]	me045	182.304	[o_k .]
#00e42a1[...]	mn015	188.100	[but i think that the point is that out of the twelve possible utterances that the german system can do we've already written the - the syntax trees for three or four .]
#2fb3285[...]	me003	198.650	[we- ==]
#9b19ba6[...]	me003	198.710	[yeah .]
#2209f3e[...]	me003	198.860	[so the syntax trees are very simple .]
#a2ce2d4[...]	me003	200.740	[it's like most of the sentences in one tree .]
#5d41186[...]	me045	203.325	[mm-hmm .]
#be9c837[...]	me003	203.683	[and instead of you know breaking down to like small units and building back up they basically took the sentences and basically cut them in half or you know into thirds or something like that and made trees out of those .]

Tableau 11 : exemple de transcription produite dans le projet CALO

Le projet CALO (Tur et al., 2008) se distingue par son ambition d'analyser des transcriptions de réunions et d'y identifier des *action items* et des prises de décision.

Ce travail d'envergure a montré la difficulté de la tâche (Purver et al., 2006). Il faut repérer non seulement les actes de parole correspondant à l'émission d'une tâche, mais aussi ceux qui les accompagnent, notamment les accords, les demandes de contraintes temporelles, etc. Il n'y a pas de prototype public disponible, mais il reste des sites archivés avec les modélisations développées et les outils d'annotation de dialogues[91].

Des travaux récents par MICROSOFT RESEARCH (Chen et al., 2015) ont montré l'utilité de réseaux de neurones à convolution[92] pour la reconnaissance d'un ensemble prédéfini d'actions dans les transcriptions de réunions.

[90] Pour l'équivalent du terme anglais *disfluent*, nous préférons ne pas utiliser « disfluent », qui n'est pas attesté, et « disfluide », qui ne semble être attesté qu'en physique.
[91] https://web.archive.org/web/20100316165645/http://caloproject.sri.com:80/
[92] https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

III.1.3 Corpus de courriels existants

III.1.3.1 Anglais

Le plus grand et le plus célèbre corpus de courriels est sans doute le corpus ENRON. Il est issu d'une enquête judiciaire, et a été publié en 2004. Dans son état original, il était très gros, sans pièces jointes, et publié sous forme brute, non immédiatement exploitable. Will Styler (2011) en a ensuite publié un sous-ensemble, sous forme de fragments normalisés et nettoyés.

Le corpus ENRON est constitué de courriels internes d'une grosse entreprise, d'où des particularités thématiques et stylistiques. Beaucoup de messages sont professionnels, mais il y a aussi des échanges personnels, des messages automatiques, de la publicité.

Ce corpus a été la base de nombreuses recherches en TAL des courriels et sur les réseaux sociaux formés par les participants des conversations. Il marque l'essor du TAL des correspondances électroniques.

Une autre source de données en anglais, bien plus petite (40 fils de discussion, 3222 phrases), est le corpus BC3[93].

Ce corpus de conversations, dont un exemple est donné ci-dessous, a l'avantage d'avoir été prétraité, segmenté en phrases, et normalisé. Il est doté d'annotations en thèmes, polarités et actes de parole (*Propose*, *Request*, *Commit*, *Meeting*).

```
<DOC>
      <Received>Wed Dec 09 20:21:11 -0800 1998</Received>
      <From>Terry Allen &lt;tallen@sonic.net&gt;</From>
      <To>discuss@apps.ietf.org,jpalme@dsv.su.se</To>
      <Subject>Re: Extending IETF meetings to two weeks?</Subject>
      <Text>
        <Sent id="2.1">&gt; The IETF meetings tend to become too large,
creating logistics and planning problems. ... </Sent>
        <Sent id="2.2">My problem over the past year or so is that there
are only a few session I wish to attend, but I cannot know for sure when
they will be scheduled, so I cannot make reasonable travel arrangements
(a week in Orlando for 6 hours of meetings is hard to sell to
management). </Sent>
        <Sent id="2.3">Now I know there is a rationale here, and that one
is encouraged to participate broadly. </Sent>
        <Sent id="2.4">And I am hopeful that new activities (my own and
in the IETF) will give me many more reasons to attend. </Sent>
        <Sent id="2.5">But firmer scheduling would be a big win. </Sent>
        <Sent id="2.6">regards, Terry </Sent>
        <Sent id="2.7">Terry Allen Electronic Commerce and Publishing
Consultant </Sent>
        <Sent id="2.8">tallen[at]sonic.net </Sent>
        <Sent id="2.9">http://www.sonic.net/~tallen/ </Sent>
        <Sent id="2.10">DocBook: http://www.ora.com/davenport/index.html
</Sent>
        <Sent id="2.11">Common Business Library:
http://www.veosystems.com/</Sent>
      </Text>
    </DOC>
```

Figure 27 : extrait du corpus BC3 illustrant la structuration d'un message

[93] https://www.cs.ubc.ca/cs-research/lci/research-groups/natural-language-processing/bc3.html

(Bennett et Carbonnell, 2005) ont publié un corpus de 744 messages[94] annotés au niveau de la phrase par la présence ou l'absence d'*action items*. Ce corpus est entièrement anonymisé, c'est-à-dire que chaque mot unique a été substitué par une chaîne aléatoire.

Étant donnée la faible variabilité morphologique des mots anglais, ce corpus se prête à la reproduction de leurs résultats de classification automatique à base de n-grammes.

```
From:
To:
Subject: qqnl kidng jziby
Date:
A:

nh  qqnl   bln
rerlfndo qho sad andcm  jziby mbk malf rhhr arvuztk.   qqnl ktbxudlcgo
osto   rar tlervnfbjzp ds exi jdxcs  dxlqbw exi yaqonejg. qqnl goiu
osto
  rar zwi cojfj mpzzfbzg dxlqbw exi yaqonejg.
   qqnl blerttygr osto img      qqnl goiu osto img
xibbwe sxdkrjt
```

Figure 28 : message en anglais anonymisé du corpus de Bennett et Carbonnell

Une telle approche à l'anonymisation ne conviendrait pas à des langues à morphologie flexionnelle plus riche, telles que le français, l'allemand ou le russe, car elle empêcherait une analyse morphologique.

III.1.3.2 **Français**

Nous n'avons pu trouver que deux corpus de courriels en français. Ce sont le corpus UBUNTU-FR[95] et le corpus SIMULIGNE. Le corpus UBUNTU-FR contient des échanges d'entraide autour de la distribution UBUNTU[96] de LINUX. Depuis le début de la liste de diffusion en 2004, il a accumulé plus de 40000 messages. Il peut être consulté sur Internet ou téléchargé, avec une licence CC BY-SA v3.0[97]. Ces courriels sont dans leur état brut, non prétraités, non segmentés et non normalisés. Il est intéressant de noter que ce sont non seulement les courriels qui sont disponibles, mais aussi le forum et le tchat.

Les inconvénients de ce corpus pour nous sont que (1) ce ne sont pas des courriels professionnels, et que (2) le sujet de discussion est trop restreint et technique.

Le corpus en français SIMULIGNE[98] (Chanier et al., 2008) est constitué d'échanges centrés sur l'apprentissage du français. Il est multimodal, car il inclut non seulement des courriels, mais aussi des tchats et des forums.

Il est intéressant par sa structuration mais, comme son domaine est assez éloigné de ce que nous voulons traiter, il présente peu d'intérêt pour nous.

[94] http://www.cs.cmu.edu/~pbennett/action-item-dataset.tgz
[95] https://lists.ubuntu.com/archives/ubuntu-fr/
[96] https://www.ubuntu.com/
[97] https://creativecommons.org/licenses/by-sa/3.0/deed.en
[98] https://repository.ortolang.fr/api/content/comere/v3.3/cmr-simuligne.html

Voici un extrait du corpus SIMULIGNE, qui montre une structuration d'un courriel en TEI[99].

```
<post xml:id="cmr-Simu-Mo-Nl1-Inbox-0005" when="2001-07-02T14:36:00"
   who="#cmr-Simu-Tc9" type="email-message">
   <head>
      <title>Question dans CUMULI</title>
      <listPerson>
         <person corresp="#cmr-Simu-Nl1">
            <event type="SendTo">
               <label>SendTo</label>
            </event>
         </person>
         <person corresp="#cmr-Simu-Tc9">
            <event type="Read" when="2001-07-02T14:36:00">
               <label>Read</label>
            </event>
         </person>
      </listPerson>
   </head>
   <p>Bonjour<name ref="#cmr-Simu-Nl1" type="person"
            ><forename>Hilary</forename></name> ! En tant que concepteur
      de CUMULI, je me permets de vous demander de (re-)poser votre
      question. La raison pour laquelle votre question n'apparaît pas dans
      le forum est que, dans l'étape 2 (après comparaison des autres
      entrées de la FAQ) vous N'AVEZ PAS EXPLICITEMENT pressé sur le
      bouton : "C'est une nouvelle question, vous voulez qu'elle soit
      posée aux autres apprenants". Il est indispensable de cliquer sur ce
      bouton pour que votre question soit réellement prise en compte. Je
      serais heureux que vous réessayiez rapidement. Si le problème vient
      d'ailleurs, merci de m'en faire part... Cordialement, <name
         ref="#cmr-Simu-Tc9" type="person"
            ><forename>Christophe</forename></name>. </p>
</post>
```

Figure 29 : extrait du corpus SIMULIGNE

III.1.3.3 Récapitulatif des corpus disponibles

Voici un tableau qui récapitule les principaux corpus d'échanges textuels en anglais et en français.

Nom et langue	Taille	Type	Prétraitements	Licence
ENRON-EN	~500000 messages	Courriels d'entreprise	Non (courriels dans des fichiers de texte bruts)	Domaine public
ENRON-SENT-EN	96107 messages	Courriels d'entreprise	Courriels nettoyés de leur métadonnées	Domaine public
PARAKWEET-EN	~5000 segments	Phrases issues principalement d'ENRON	Feuille XLS, segments annotés de manière binaire	Apache Software Licence[100]
BC3-EN	40 fils de discussion 3222 phrases	Courriels académiques	Messages structurés dans des fichiers XML en fils de discussion, segmentés, annotés	MIT Licence[101]
CMU-EN	744 messages	Courriels académiques	Messages nettoyés de texte cité et complètement anonymisés	Domaine public ?
UBUNTU-FR	40000+ messages	Courriels, forum, tchat informatiques	Non, fichiers EML	CC BY-SA v3.0
SIMULIGNE-FR	6790 messages tchat 2030 courriels 2686 messages forum	Courriels, forum, tchat d'étudiants de langue	Oui, mise au format TEI	CC0 1.0[102]

Tableau 12 : corpus d'échanges textuels disponibles pour le français et l'anglais fin 2015

[99] http://www.tei-c.org/index.xml
[100] https://www.apache.org/licenses/LICENSE-2.0.html
[101] https://opensource.org/licenses/mit-license.php
[102] https://creativecommons.org/publicdomain/zero/1.0/

III.2 Création de corpus de mels principalement en français

Nous avons constitué notre corpus de courriels issus d'activités collaboratives en faisant l'union de quatre plus petites collections de courriels. La première rassemble les échanges autour d'un projet ANR d'un laboratoire de recherche mené durant la période 2011-2013 (TRAOUIERO). La seconde (VISEO) provient d'échanges autour de plusieurs projets scientifiques collaboratifs d'une entreprise. Les deux autres sont relatives au projet MACAU (voir IV.1) et à des échanges concernant la gestion de l'emploi du temps de l'école PAGORA de G-INP (ADE).

Le corpus a été lu et un ensemble de catégories, présenté dans la section suivante, a été défini. Nous présentons plus bas les caractéristiques quantitatives de la partie annotée du corpus.

III.2.1 Sources

Nos courriels proviennent de quatre sources : des courriels internes de VISEO, des courriels d'un projet du GETALP, des courriels du projet MACAU (cf. Chapitre IV) et des courriels relatifs à la gestion d'un logiciel de planification.

Afin d'aider à pallier le manque de ressources en français, nous avons voulu constituer un corpus de courriels en français, représentatif des échanges professionnels, contenant des tâches, et annoté d'une manière fine qui identifie les énoncés, les tâches, et leurs attributs. Nous avons introduit un certain nombre d'attributs qu'on sait *a priori* nécessaires, (par exemple, une tâche possède une échéance, un émetteur, un ou des attributaires) mais sans limiter la liste, car d'autres attributs utiles auraient pu émerger au cours de l'analyse du corpus.

La première phase de ce travail a donc été la collecte des données et leur analyse par un lecteur. Cette analyse préalable à l'annotation a permis la formulation d'une première modélisation, qui a été enrichie au cours de l'étape suivante.

Ensuite est venue la phase d'annotation, qui vise à identifier les fragments textuels qui correspondent aux tâches et attributs définis dans l'étape précédente ou introduits au cours du processus d'annotation. Cette première expérience d'annotation pourra dans le futur servir de base pour formuler des instructions aux annotateurs pour une future annotation à plusieurs. En effet, (Lampert et al., 2008), qui ont effectué un travail similaire sur le corpus ENRON, ont observé un ensemble de cas d'ambiguïté qui surviennent dans les messages et pour lesquels les annotateurs n'ont pas d'idée claire sur la façon de les annoter. Afin de minimiser le désaccord entre annotateurs, des instructions pour traiter ce genre de cas doivent être prévues en amont.

Une fois l'annotation achevée, les données ont été exploitées pour entraîner des algorithmes d'apprentissage automatique et procéder à des expériences.

III.2.2 Annotation

Nous expliquons ici l'expérience d'annotation et justifions le choix des catégories retenues. Notre modèle d'annotation contient des relations telles que :

- « argument » pour rattacher des attributs aux segments qui réfèrent à des tâches.
- « alternative », pour indiquer qu'une tâche est alternative à une autre.
- « sous_tâche », lorsqu'on estime qu'un segment décrit une sous-tâche à une tâche décrite dans un autre segment.
- « *same_as* », pour indiquer que deux segments réfèrent à une même tâche, ou que deux fragments réfèrent à un même attribut (les noms de personnes et les dates).

III.2.2.1 Catégories d'annotation

Demande d'information. C'est une catégorie issue de DAMSL (Allen et al. 1997) qui place l'obligation de répondre sur la personne à qui elle s'adresse. Bien que l'action de répondre

puisse être considérée comme une tâche, puisqu'une action est attendue, nous avons préféré séparer ce type d'énoncé de la catégorie générale d'action, car ce type d'action ne requiert en général pas de la planifier, alors que la construction automatique d'un plan est une des applications envisagées pour ce travail.

Suggestion. C'est un énoncé qui propose d'effectuer une action, mais sans placer d'obligation sur la personne à laquelle la proposition est adressée.

Interdiction. C'est une demande de ne pas effectuer une action. Ce type d'énoncé est mentionné par (Lampert et al., 2008), et nous l'avons créé *a priori*, avant d'annoter le corpus. Dans le corpus annoté, nous n'avons repéré que 6 instances de cette catégorie.

Action. C'est la catégorie la plus peuplée ; elle regroupe les énoncés qui placent une obligation, sur les personnes auxquelles ils s'adressent, d'effectuer une action, cette action étant le plus souvent décrite dans l'énoncé même ou déduite du contexte environnant. Ici, nous retenons la définition de (Lampert et al., 2008) pour leur catégorie *request*. Nous nous sommes aperçu qu'environ un tiers des demandes d'action concerne des documents : il s'agit de travailler dessus ou de les envoyer. Nous avons donc créé deux sous-catégories qui le reflètent.

Invitation. C'est une catégorie que nous avons créée pour annoter les énoncés qui imposent au lecteur l'obligation de répondre ou de prévoir un créneau dans son calendrier (par exemple pour des réunions). En annotant le corpus, nous avons remarqué que de nombreuses invitations ne sont que des propositions sans aucune contrainte pour les personnes qui les reçoivent.

III.2.2.2 **Types d'attributs**

Émetteur. Il s'agit des personnes qui demandent la réalisation d'une action.

Attributaire. Il s'agit de la ou les personnes à qui la réalisation d'une action est demandée.

Bénéficiaire. Il s'agit de la personne qui bénéficie de la réalisation d'une action, lorsqu'elle est explicitement mentionnée et n'est pas l'émetteur.

Contrainte temporelle.

- **Début**. Cet attribut sert à indiquer quand la réalisation d'une action doit commencer (p.ex. : « après le rendu des livrables »). Nous n'avons observé que quatre instances de cette catégorie.
- **Échéance**. Cet attribut marque le moment pour lequel la réalisation de l'action est attendue

Justification. Cette annotation sert à marquer les énoncés qui expliquent pourquoi l'action doit être réalisée. Nous avons estimé utile d'annoter ces énoncés car, si on extrait les tâches pour les présenter à l'utilisateur, on voudrait également pouvoir lui présenter ce type d'explication.

Instruction. Cette catégorie sert à annoter les instructions, consignes ou détails de réalisation associés à une tâche.

Condition. Il s'agit des conditions qui doivent être vérifiées pour que la tâche soit réalisée.

Rappel. Cette catégorie marque les énoncés qui rappellent le besoin de réaliser une tâche.

Nous avons de plus des annotations de structure (non représentées dans la figure ci-dessous), dont le but est d'identifier des portions des messages qui, pour une raison ou une autre, doivent être ignorées lors d'une analyse automatique, car elles ne contiennent pas d'énoncés relatifs à la conversation et risquent d'introduire du bruit.

Arnaud,

Peux-tu me dire si tu prévoies de présenter des diapos sur le projet ? Si tel est le cas, peux-tu me les envoyer avant pour que j'ajuste les miennes afin d'éviter les redondances.

Merci

Véronique

Figure 30 : exemple de message annoté

Dans cet exemple, on voit une demande d'information (la première phrase), une tâche (la deuxième phrase), une condition (« si tel est le cas »), une justification (« pour que j'ajuste les miennes… »), un émetteur (« Véronique ») et un assignataire (« Arnaud »).

III.2.2.3 **Description et évaluation du processus d'annotation**

III.2.2.3.1 Outils

L'annotation a d'abord été réalisée à l'aide de l'outil WIRED MARKER[103], un plugin FIREFOX qui permet de définir une hiérarchie de catégories et de marquer les plages dans le texte par ces catégories. Un avantage de cet outil est de ne pas requérir un format spécifique ou une mise en forme particulière pour les données, et de fonctionner sur toute page que FIREFOX peut afficher, en stockant les annotations dans un fichier compagnon.

Un autre outil d'annotation, BRAT[104] (Stenetorp et al. 2012) permet des annotations plus riches, avec possibilité de définir des relations. Une transformation des données annotées qui s'y prêtent vers le format BRAT a été faite, et a permis d'effectuer la deuxième partie de l'annotation, qui consiste à lier les entités entre elles.

III.2.2.3.2 Prétraitement des données

Les fichiers EML ont été traités par SEGNORM de façon à les renommer, à les segmenter en messages individuels, à en extraire le texte, à le normaliser, et à grouper les fils de discussion dans des répertoires dédiés.

La première étape consiste à réunir les messages à analyser en un seul dossier, et à les charger en mémoire.

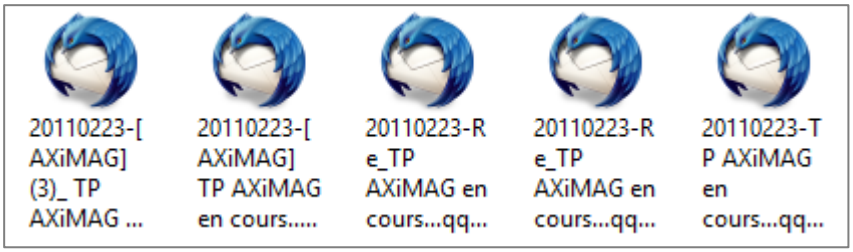


Figure 31 : étape 1 : état de départ

Ensuite, SEGNORM reconstruit les fils de discussion à l'aide des informations contenues dans les métadonnées des fichiers EML.

[103] http://www.wired-marker.org/en/
[104] http://brat.nlplab.org/

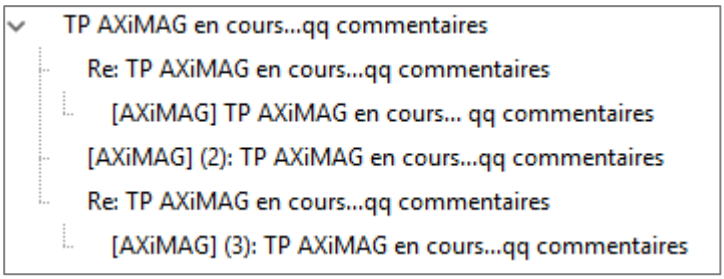


Figure 32 : étape 2 : reconstruction de l'arbre des conversations

La troisième étape consiste à extraire le contenu textuel des messages et à le normaliser. A cette étape SEGNORM reconstruit aussi les messages manquants à partir du texte des messages qui les contiennent.

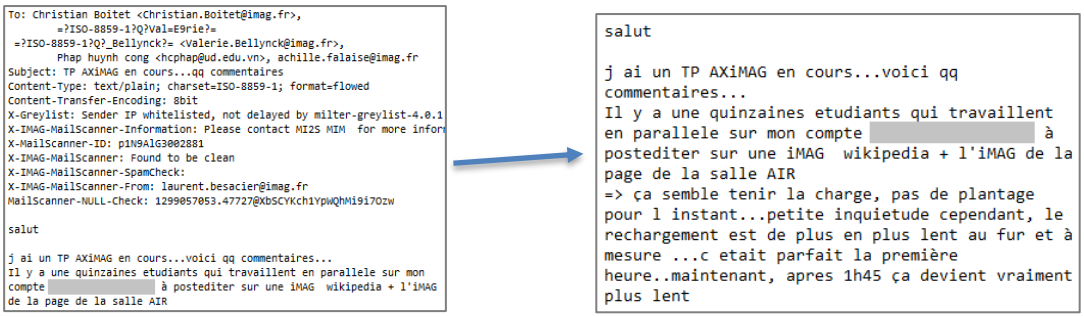


Figure 33 : étape 3 : extraction du texte

La quatrième étape crée des fichiers texte contenant les textes des messages pour chaque conversation.

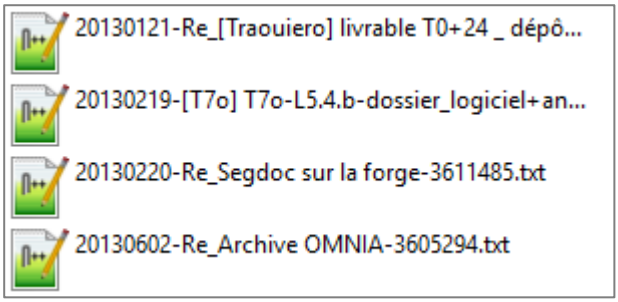


Figure 34 : étape 4 : fabrication des fichiers texte représentant la conversation

III.2.2.3.3 Bilan

L'annotation d'est déroulée sur un mois, à raison de 4 heures par jour, soit 88 heures au total.

Voici la répartition des segments par type d'annotation.

	Catégorie	**Nombre**	**Exemples**
Types d'énoncé	Action	279	Vérifiez avec Pierre la faisabilité de l'accueil de ce prototype, et j'appliquerai les décisions.
	Travailler sur document	97	Ci-joint pour validation avant envoi à ACME au plus tard demain midi.
	Fournir document	142	Merci de faire parvenir vos fiches remplies à S. Cavern, E. Teraume et moi-même. Peux-tu me renvoyer les fichiers d'organisation dans leur nouvel état ?
	Demande d'informations	212	Pierre, qu'en dis-tu? Je laisse JC te répondre pour le voltage (220/380). Si vous avez des exigences alimentaires spécifiques, merci de me mettre au courant.
	Suggestion	39	Je suis d'accord, si tu pouvais les fusionner ce serait bien. Vos commentaires, compliments, suggestions sont bienvenus jusqu'à Jeudi soir. N'hésitez pas à rajouter celles qui manquent le cas échéant.
	Invitation	39	Par contre, on va faire une réunion de travail avec JC et toi pour te montrer tous ces points. En tant que coordinateur de projet, nous vous demandons de bien vouloir y assister avec votre partenaire structure de valorisation.
	Interdiction	6	Ces originaux ne doivent être ni datés, ni imprimés en mode recto-verso.
Attributs	Émetteur	555	Pr. Jean-Marie Martin / Amélie / MJo / JC / F
	Attributaire	400	Didier / tous / partenaires industriels
	Bénéficiaire	9	Elizabeth BRUN / François / nous
	Justification	83	Là, ça m'empêche de mettre ton document dans le format des livrables, c'est donc très ennuyeux ! C'est nécessaire pour qu'on puisse cliquer sur "valider" et que l'APT puisse valider puis notifier.
	Instruction	126	Il faut arriver à peu près au même niveau de détail que ce qu'a écrit Alexandre. […] avec copie à la direction de la recherche
	Condition	39	S'il y a lieu […] / si c'est possible si vous n'arrivez pas à accéder à ce fichier […] ; si vous avez des dépenses à partir du 1er janvier ou du 15 janvier
	Échéance	88	demain / lundi dernier délai / 10/09/14
	Début	4	après le rendu des livrables
	Marqueur d'urgence	43	la date limite arrive... très rapidement SUPER-URGENT dqp !

Tableau 13 : entités et exemples

Le corpus se présente comme un ensemble de fichiers texte, chacun contenant une conversation. 320 conversations contenant des tâches ont été identifiées, chaque conversation contenant un ou plusieurs messages, formant un échange, pour un total de 1040 messages.
Ces messages contiennent plus de 12520 segments textuels, en moyenne 12 segments par message. Après annotation, nous avons identifié 730 segments exprimant une tâche (toutes sous-catégories confondues). Nous avons compté 601 tâches différentes.
Nous avons observé que, parmi ces tâches, 129 sont référées par plus d'un énoncé. Les énoncés correspondant à une même tâche peuvent se trouver à différents endroits d'un message ou de la conversation (par exemple, les rappels).

Le Tableau 13 ci-dessus donne le reste des nombres d'occurrences par catégorie, ainsi que des exemples d'éléments dans chaque catégorie.

Il est intéressant de remarquer les variations de graphie dans les noms, les dates, et certains autres attributs. Un système d'extraction automatique doit savoir ou apprendre à interpréter ces graphies.

L'annotation a été faite par un seul annotateur, l'auteur de ces lignes, mais c'est un travail préliminaire nécessaire pour pouvoir formuler des instructions à un futur groupe d'annotateurs. D'autre part, la nature confidentielle des données initiales limitait la possibilité de diffuser les données.

Des propriétés utiles peuvent être déduites du texte sans qu'on puisse identifier précisément le ou les énoncés qui les expriment. Un exemple est le niveau d'urgence, déductible à partir de la date d'envoi du message et d'une date limite indiquée dans le corps du message.

Notons que certaines catégories sont trop peu remplies pour l'instant pour être utiles pour de l'apprentissage automatique.

La difficulté de décider ce qui constitue ou non une tâche a été explorée dans la littérature. Dans l'article *Requests and Commitments are More Complex than You Think : Eight Reasons to be Cautious*, (Lampert et al. 2008) ont identifié huit phénomènes apparaissant dans les courriels, et rendant difficile l'identification de requêtes et engagements.

a. *Ambiguïté de position (locus ambiguity)*

Lorsque deux phrases indiquent la présence d'une requête, et que l'une l'indique implicitement et l'autre explicitement, les annotateurs trouvent que la présence d'une requête dans la première n'est pas sûre et sont en désaccord sur cette phrase. Si on ôte la deuxième phrase, l'ambiguïté sur la première disparaît.

Dans ce cas, (Lampert et al. 2008) préconisent d'annoter chaque phrase pouvant exprimer une requête, comme une requête. Nous avons fait de même dans notre corpus, mais avons ajouté une relation *same_as* entre les multiples phrases exprimant la même requête.

Une autre variante de cette ambiguïté se présente lorsqu'un courriel dans son ensemble indique une requête, mais qu'aucune phrase individuelle ne décrit la requête dans son ensemble.

b. *Les réunions*

Les annonces de réunions sont en fait des requêtes, dans le sens où elles sont des requêtes implicites de présence. Il n'est pas toujours clair de déterminer si l'expéditeur d'une annonce de réunion s'engage à y participer lui-même. Nous avons annoté les annonces de réunions, et autres appels à participer à une rencontre par la catégorie *Invitation*.

c. *Les formules de fin de phrase*

Des phrases comme « *let me know if you have any questions* » sont très courantes dans les courriels en anglais, et se rencontrent également en français. Parfois, elles portent la force illocutoire de requête, mais bien souvent leur présence est simplement le reflet de normes sociales. Il est difficile de départager ces deux cas, et il faut donc observer ces formules dans le contexte du courriel entier.

d. *Requêtes d'inaction*

Ces requêtes interdisent une action (« *This is an internal document, don't distribute it* ») ou appellent à ne pas agir. Il est clair qu'elles ne conviennent pas pour une liste de choses à faire, mais elles placent tout de même une obligation sur le récipiendaire. Nous avons annoté ces énoncés avec la catégorie *Interdiction*, mais n'en avons observé que 6.

Quant aux phrases comme « N'oubliez pas de nous renvoyer le document », ce sont bien des appels à agir, et non des requêtes d'inaction.

e. *Instructions de processus*

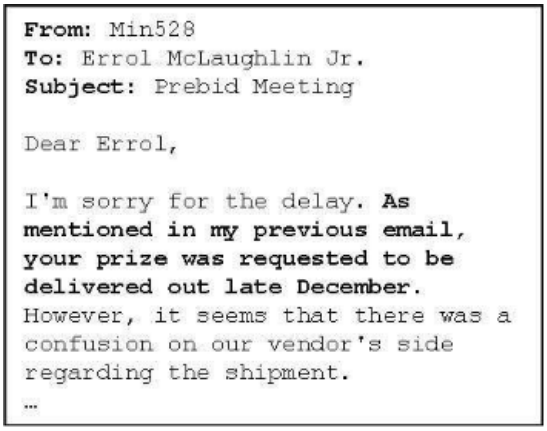
Il s'agit de phrases qui décrivent la conduite à tenir si une condition est vraie, par exemple « En cas d'incendie, dirigez-vous calmement vers la sortie de secours ». Pour déterminer si ce sont effectivement des requêtes, les annotateurs estiment la probabilité que la condition soit vraie. Pour la plupart, l'exemple ci-dessus n'est pas une requête.

f. *Requêtes demandant de revoir un fichier attaché*

Il s'agit de phrases comme « Voir fichier ci-joint ». Notre catégorie *Travailler_sur_document* inclut des énoncés laconiques de cette sorte.

g. *Requêtes et engagements reportés*

Il s'agit de segments qui, s'ils étaient isolés, seraient bien des expressions de tâches, mais qui sont suivis par des segments qui les annulent ou qui indiquent que ces tâches ne sont plus d'actualité. L'exemple ci-dessous de la Figure 35, issu toujours de (Lampert et al. 2008), illustre cela.


From: Min528
To: Errol McLaughlin Jr.
Subject: Prebid Meeting

Dear Errol,

I'm sorry for the delay. **As mentioned in my previous email, your prize was requested to be delivered out late December.** However, it seems that there was a confusion on our vendor's side regarding the shipment.
…

Figure 35 : illustration d'un segment suivi d'un indicateur de report

h. *Engagements de partie tierce*

Les segments du type « ma secrétaire Kim va le faire », sont à interpréter comme des requêtes si la personne mentionnée figure dans la liste des destinataires.

III.2.2.4 **Exemples**

L'exemple ci-dessous illustre une conversation annotée. Nous représentons ici les annotations avec des balises pseudo-XML.

Sujet : RE: projet PROJET
De : "RIVIERE, Nicolas" <Nicolas.Riviere@adresse1.fr>
Date : JJ/MM/AAAA HH:MM
Pour : "Marchand, Fabrice" <Fabrice.Marchand@adresse1.fr>, "BRIAN, Julie" <Julie.Brian@adresse1.fr>
Copie à : Françoise Généreux <françoise.genereux@adresse2.com>

《entity type="Send_document" id="1"》 Ok alors envoyer le fichier à 《entity type="Beneficiary" id="2" relation="Argument(1)"》 Françoise 《/entity id="2"》 《/entity id="1"》 cdt

---

Prof. 《entity type="Requester" id="3" relation="Argument(1)"》 Nicolas Rivière 《/entity id="3"》 , PhD

[SIGNATURE Nicolas Rivière]

De : MARCHAND, Fabrice
Envoyé : jour JJ mois AAAA HH:MM
À : BRIAN, Julie
Cc : RIVIERE, Nicolas; 'fgenereux@adresse2.com'
Objet : RE: projet PROJET

《entity type="Requestee" id="4" relation="Argument(5)"》 Julie 《/entity id="4"》 , je viens d'essayer d'appliquer la procédure de récupération de mot de passe, mais le système répond toujours "Courriel inconnu", j'ai essyé avec et sans majuscules, avec l'ancien Login de nicolas, rien à faire. Il doit y avoir un souci au niveau de l'enregistrement chez Site.

《entity type="Action_request" id="5"》 Peux-tu transférer les info à 《entity type="Beneficiary" id="6" relation="Argument(5)"》 Françoise 《/entity id="6"》 , STP. 《/entity id="5"》

Merci
《entity type="Requester" id="7" relation="Argument(5)"》 Fabrice 《/entity id="7"》

De : BRIAN, Julie
Envoyé : jour JJ mois AAAA HH:MM
À : MARCHAND, Fabrice
Objet : TR: projet PROJET

De : Françoise Généreux [mailto:fgenereux@adresse2.com]
Envoyé : jour JJ mois AAAA HH:MM
À : BRIAN, Julie
Cc : RIVIERE, Nicolas
Objet : RE: projet PROJET

Bonjour,
C'est Nicolas Rivière qui a du le ecevoir au debut du mois. Moi je ne l'ai pas.
《entity type="Send_document" id="8" relation="AlternativeTo(9)"》 Sinon envoyez moi vite les donnees et je les rentrerai 《/entity id="8"》
《entity type="Action_request" id="9"》 Autre solution ecrivez au support site 《/entity id="9"》 . Ils repondent vite
Merci

```
《entity type="Requester" id="10" relation="Argument(9)"
relation="Argument(8)"》 Françoise Généreux 《/entity id="10"》

De : BRIAN, Julie [mailto:Julie.Brian@adresse1.fr]
Envoyé : jour JJ mois AAAA HH:MM
À : Françoise Généreux
Objet : projet PROJET

Je suis la secrétaire de l'équipe EQUIPE,
Je travail sur le dossier en collaboration.
《entity type="Information_request" id="11"》 Nous aurions besoin d'un
login / pwd pour rentrez le tout dans SITE 《/entity id="11"》 . nous le
mettrons en copie à M.RIVIERE

Je reste à votre disposition pour tous rendeignements complémentaires
Cordialement

《entity type="Requester" id="12" relation="Argument(11)"》 Julie BRIAN 《
/entity id="12"》

[SIGNATURE Julie Brian]
```

Figure 36 : exemple de conversation pseudonymisée et annotée, sinon verbatim

III.2.2.5 Vers l'accès intégral à ces corpus

Étant donnée l'indisponibilité de courriels professionnels en français, annotés et téléchargeables, publier tout ou une partie de ces données serait certainement utile. Cela impliquerait d'anonymiser les données.

III.2.2.5.1 Nécessité d'anonymisation

La loi interdit en effet la diffusion publique de correspondances personnelles sans l'accord de tous les participants, à moins qu'elles n'aient été anonymisées ou « pseudonymisées ». Le terme « pseudonymisation » a été introduit par (De Mazancourt et al., 2014), et implique non seulement de supprimer les données sensibles (noms propres, contacts, etc.) mais aussi de les remplacer par des données qui leur ressemblent, de sorte qu'il soit possible d'effectuer des traitements automatiques et des expériences scientifiques.

III.2.2.5.2 Degrés d'anonymisation

Citons (De Mazancourt et al., 2014), qui distinguent trois niveaux d'anonymisation :

- *anonymisation de premier niveau : obfusquer*[105] *les métadonnées émetteurs et récepteurs (i.e. les champs from, to et cc des courriels),*
- *anonymisation de deuxième niveau : en plus de l'anonymisation de niveau 1, empêcher toute collecte des informations de contacts (par exemple à des fins publicitaires) contenues dans le corps des courriels,*
- *anonymisation de troisième niveau (ou anonymisation vraie) : garantir qu'il n'est pas possible, directement ou indirectement, d'identifier les émetteurs, les récepteurs ni les tiers mentionnés dans le corps des courriels.*

Nous avons vu dans la Figure 28 un exemple de courriel en anglais issu du corpus CMU et entièrement anonymisé, aucune métadonnée n'étant présente, et chaque chaîne unique étant remplacée par une chaîne aléatoire. Il n'est pas envisageable d'employer le même degré

[105] Cet anglicisme réfère en informatique à l'offuscation, « une stratégie de gestion de l'information qui vise à obscurcir le sens qui peut être tiré d'un message » (cf. https://fr.wikipedia.org/wiki/Offuscation). Wikipédia propose également les synonymes *masquage* et *opacification*, peut-être moins dysphoniques pour l'oreille française.

d'anonymisation lorsqu'on veut analyser la morphologie ou la syntaxe du texte (à moins d'adapter les outils).

Lors du développement, nous n'avons pas eu recours à une anonymisation automatique car nous ne voulions pas perdre des informations intéressantes sur les variations graphiques des noms dans les écrits. Le Tableau 13 (page 71) contient des exemples de noms de personnes, parmi lesquels on trouve des initiales, des abréviations, et autres cas de formes de noms différentes de leurs forme canonique (qu'on trouve dans les en-têtes ou les métadonnées des messages). Il nous a semblé important de préserver ces formes.

III.2.2.5.3 Méthode et type de dépôt

Actuellement, le corpus et ses annotations ne sont pas distribuables, mais on envisage d'en sélectionner et publier un fragment anonymisé ou pseudonymisé semi-automatiquement. Il sera publié sur un GITHUB, avec les courriels en plusieurs formes (fichiers EML + fichiers segnormalisés) et les annotations dans un fichier séparé.

III.3 Expériences de repérage

Notre objectif final consiste à identifier le plus possible de tâches à partir des énoncés qui en parlent (dits « repérés »), en partant du principe que plusieurs énoncés qui apparaissent au sein d'un message, voire d'une conversation, font souvent référence à la même tâche sous-jacente, et que cela peut être détecté par des mesures de sémantique lexicale.

À terme, nous envisageons aussi de construire un graphe des tâches faisant apparaître leurs relations (`sous-tâche_de`, `dépend_de`, etc.), puis de relier ce graphe à la description (ontologie ou plus simplement carte sémantique comme dans SYNAPS) du projet, si elle existe.

Pour cela, il faudra préalablement repérer les segments relatifs à des tâches. Ce problème a déjà été abordé dans le cadre d'études sur des courriels en anglais. Les résultats de méthodes expertes sont assez bons, et ceux de méthodes empiriques moins bons. Nous avons cherché à combiner les deux approches, en les appliquant au français, et nous obtenons des résultats d'assez bonne qualité pour, dans une partie des cas, arriver à identifier des tâches.

III.3.1 Objectifs des expériences

III.3.1.1 Repérage d'énoncés relatifs à des tâches

III.3.1.1.1 But

Le but ici est d'étiqueter chaque segment de façon binaire : est-il ou non relatif à une ou des tâches ?

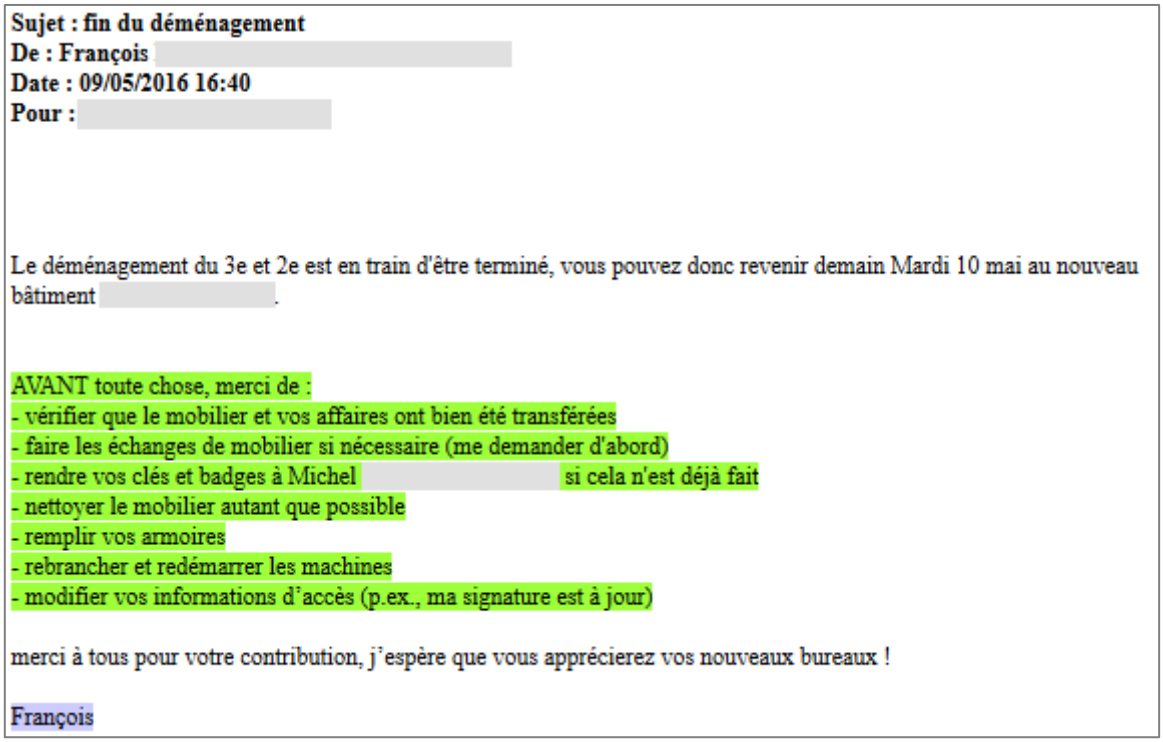
On peut utiliser une méthode experte, ou une méthode empirique (classifieur « appris »), ou bien les combiner en une méthode hybride.

De façon générale, dans une méthode experte, on calcule des scores partiels qu'on agrège en un score final, et dans une méthode empirique on obtient directement un score final. Ensuite, on fixe un seuil pour répondre oui ou non, avec la possibilité d'utiliser plus tard les valeurs effectives des scores des segments « repérés », par exemple comme mesures de confiance. À cause de ce parallélisme, il est en principe assez facile de combiner les deux types de méthode.

Nous nous sommes pour l'instant limité à comparer ces deux types de méthode, avant de chercher à les combiner.

Il est d'ailleurs nécessaire de résoudre ce problème, ne serait-ce que pour pouvoir évaluer l'utilité d'un outil de repérage : s'il rate un énoncé qui parle d'une tâche T, mais s'il en détecte un autre parlant aussi de T, alors on pourrait considérer que la tâche Ta été repérée.

Voici un exemple de sortie souhaitée.



Sujet : fin du déménagement
De : François
Date : 09/05/2016 16:40
Pour :

Le déménagement du 3e et 2e est en train d'être terminé, vous pouvez donc revenir demain Mardi 10 mai au nouveau bâtiment .

AVANT toute chose, merci de :
- vérifier que le mobilier et vos affaires ont bien été transférées
- faire les échanges de mobilier si nécessaire (me demander d'abord)
- rendre vos clés et badges à Michel si cela n'est déjà fait
- nettoyer le mobilier autant que possible
- remplir vos armoires
- rebrancher et redémarrer les machines
- modifier vos informations d'accès (p.ex., ma signature est à jour)

merci à tous pour votre contribution, j'espère que vous apprécierez vos nouveaux bureaux !

François

Figure 37 : exemple de repérage de tâches énumérées dans une liste

III.3.1.1.2 Difficultés a priori

Il est difficile de déterminer si plusieurs énoncés font référence à la même tâche, car cela impose d'être capable de repérer des similarités entre énoncés.

Il faut également repérer des références entre énoncés. Avant qu'un algorithme ne puisse le faire automatiquement, il faut avoir constitué un certain ensemble d'énoncés qui font référence à la même tâche.

III.3.1.1.3 Méthodes d'évaluation possibles

Dans l'idéal, il faudrait prendre en compte les étapes ultérieures pour évaluer le repérage, c'est à dire l'évaluer dans un contexte « end-to-end ».

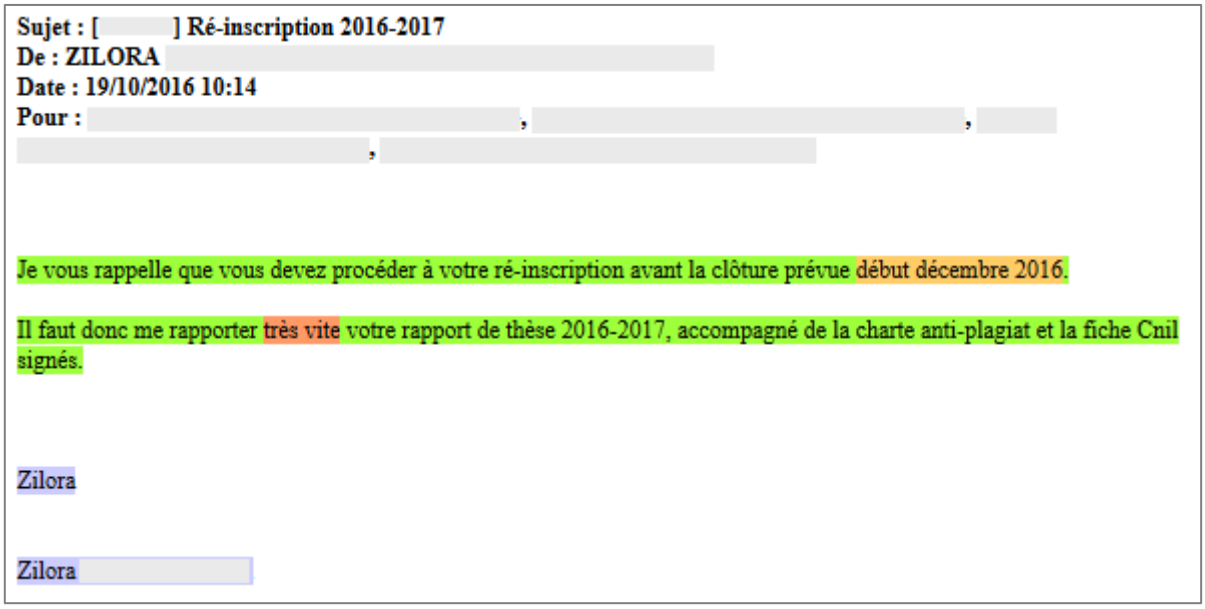
Pour l'instant, nous sommes obligé d'utiliser les mesures classiques, ce qui fait que nos résultats sont trop pessimistes.

III.3.1.2 **Repérage d'attributs**

Il s'agit en fait de trouver les valeurs d'un certain nombre de « traits » qu'on peut considérer comme des métadonnées liées à chaque segment. Par exemple, on veut déterminer l'auteur du segment, son destinataire, et son courriel d'origine s'il est cité.

On cherche également à repérer les informations relatives aux tâches, à savoir qui est à l'origine de la tâche, à qui elle s'adresse, pour quand elle doit être réalisée, et quel est son niveau d'urgence.

Voici un exemple de sortie souhaitée (et obtenue).


Sujet : [ ] Ré-inscription 2016-2017
De : ZILORA
Date : 19/10/2016 10:14
Pour :

Je vous rappelle que vous devez procéder à votre ré-inscription avant la clôture prévue début décembre 2016.

Il faut donc me rapporter très vite votre rapport de thèse 2016-2017, accompagné de la charte anti-plagiat et la fiche Cnil signés.

Zilora

Zilora

Figure 38 : exemple de sortie du repérage, avec contrainte temporelle et marqueur d'urgence

Nous voyons dans cet exemple deux segments relatifs aux tâches, un marqueur d'urgence (« très vite »), une échéance (« début décembre 2016 »), et un émetteur (« Zilora Zouaoui »).

III.3.2 Repérage d'énoncés

Nous décrivons ici les développements effectués et les résultats obtenus pour le repérage des énoncés exprimant une tâche.

III.3.2.1 Description de l'expérience

Notre approche a consisté à combiner des informations extraites de manière experte (à l'aide de règles préprogrammées), et empirique, c'est-à-dire basée sur un apprentissage automatique.

Nous avons utilisé l'outil P-LING de Cédric Lopez (VISEO), conçu pour l'identification des entités nommées. Il nous permet d'identifier des dates et des noms de personne, qui peuvent être des indicateurs importants de mention de tâche. Par exemple, il arrive souvent qu'un segment exprimant une tâche qui s'adresse à une personne, commence par le prénom de celle-ci.

Les règles linguistiques conçues pour repérer les tâches portent sur de nombreux traits caractéristiques, et ont été construites en analysant une partie du corpus. Par exemple, un indicateur fort de la présence de tâches est le fait qu'un courriel se termine par « Merci d'avance ». Nous avons identifié un ensemble de patrons caractéristiques :

- des phrases contenant « pourriez-vous » (et ses variantes « peux-tu », « pourrais-tu », « est-ce que tu pourrais », etc.)
- « merci de » (et ses variantes, comme « nous vous remercions par avance de ») suivi d'un verbe ou de pronoms et articles, puis d'un verbe (merci de nous les envoyer),
- des expressions de prière (« svp », « s'il te plait », « prière de »),
- des expressions comme « serait-(ce|il) possible de » suivies de pronoms et d'articles optionnels puis d'un verbe.
- etc.

L'Annexe 7 donne des règles linguistiques sous leur forme algorithmique.

Nous repérons également le fait qu'un segment mentionne des documents (les documents sont mentionnés dans au moins un tiers des segments exprimant des tâches) à l'aide d'un lexique de termes relatifs aux documents. Ce lexique inclut des termes assez génériques, comme « fiche », « fichier », « document », « formulaire », « justificatif », etc., ainsi que des extensions usuelles de documents (PDF, DOC(X), CSV, etc). Nous utilisons également un lexique de verbes relatifs au travail sur des documents (« compléter », « vérifier », « corriger », « relire », « consulter », « remplir », « écrire », « rédiger », « imprimer », « signer », « déposer », « faire parvenir », « envoyer », « transmettre »).

III.3.2.2 **Déroulement**

Nous nous intéressons au repérage de segments exprimant une tâche, et nous voulons comparer plusieurs approches : des approches d'apprentissage automatique (SVM et MAXENT, qui sont souvent à l'état de l'art pour ce type de tâche), une approche par règles linguistiques qui, comme on le verra, a une bonne précision mais un mauvais rappel, et enfin une combinaison des deux approches dans des classifieurs entraînés sur des traits de surface et des traits linguistiques extraits par l'approche experte.

Les traits de surface utilisés sont :

- des traits lexicaux (1-2-3-grammes, étiquettes morphosyntaxiques produites par TALISMANE,
- des traits structuraux ou liés aux métadonnées : le nombre de messages dans la conversation, le nombre de destinataires, le fait d'être premier destinataire, la présence ou non de pièces attachées, la position du segment dans le message, la valeur du champ `Priorité`, le fait d'être immédiatement précédé ou suivi par un fragment cité.

Nous avons constitué un ensemble de test à partir de conversations choisies au hasard dans les quatre sous-corpus. Le reste du corpus a servi à l'élaboration des règles linguistiques.

III.3.2.3 **Résultats et évaluation**

Le tableau ci-dessous présente les résultats de l'évaluation quantitative du repérage de segments. Nous donnons à la fois les performances sur un corpus test choisi au hasard parmi les courriels (4639 segments) et en validation croisée (1/5ème du corpus, 5 fois) sur l'ensemble complet de segments (12520 segments).

Nous donnons entre parenthèses les valeurs obtenues lorsqu'on tient compte du fait que plusieurs segments peuvent faire référence à la même tâche. Nous voyons que cela augmente légèrement le rappel, et donc la F-mesure, mais, du moins dans notre corpus, cela ne conduit pas à un grand changement qualitatif.

Nous ne donnons pas de score sur la validation croisée pour les approches combinant traits linguistiques et apprentissage automatique car elle inclut les données qui ont servi à l'élaboration des règles linguistiques, et donc les scores seraient artificiellement surestimés.

	Corpus de test			**Validation croisée (1/5)**		
	P	**R**	**F1**	**P**	**R**	**F1**
SVM seul	0,74	0,52 (0,56)	0,61 (0,64)	0,82	**0,54 (0,57)**	**0,65 (0,67)**
SVM + traits linguistiques	**0,84**	**0,59 (0,64)**	**0,69 (0,73)**	--	--	--
MaxEnt seul	0,77	0,47 (0.5)	0,59 (0,61)	**0,87**	0,45 (0.47)	0,59 (0.61)
MaxEnt + traits linguistiques	0,82	0,52 (0.55)	0,64 (0,66)	--	--	--
Règles linguistiques	0,79	0,41 (0,44)	0,54 (0.57)	--	--	--

Tableau 14 : résultats de repérage de tâches pour différentes approches

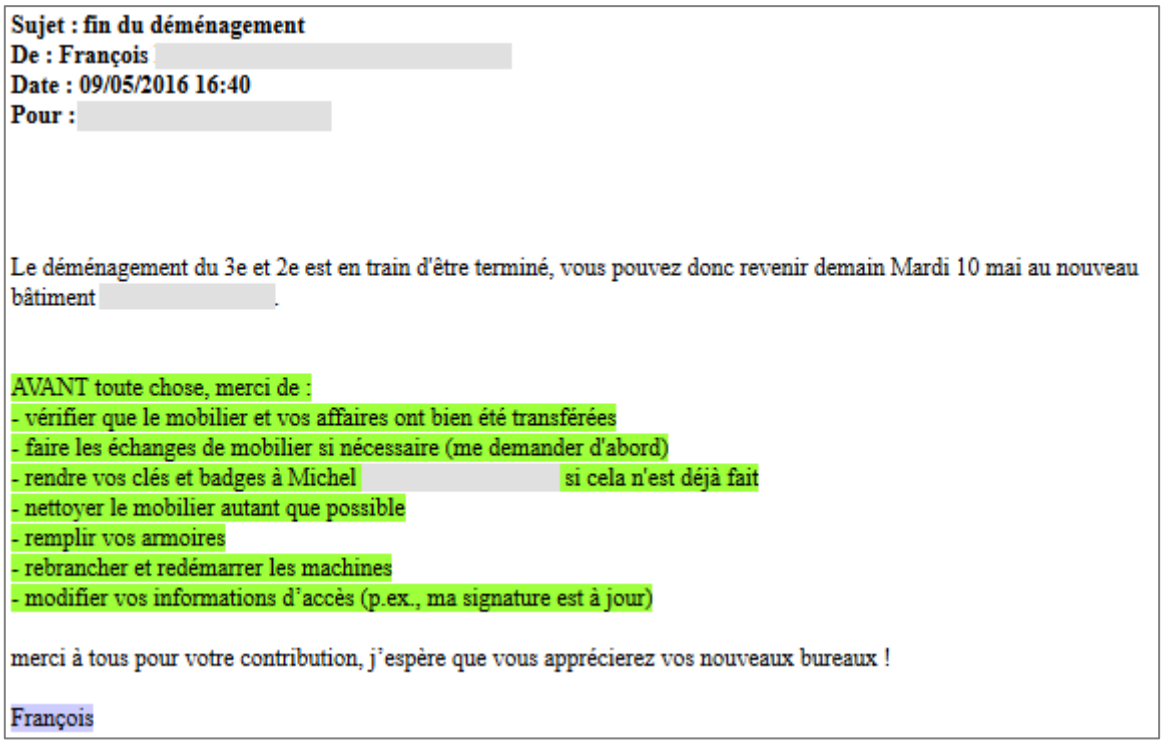
Les faux négatifs sont attribuables à 5 types de causes, à savoir :

- des erreurs d'orthographe (ou de ponctuation) :
  - « *Si elle ne l'a pas encore reçue, dite-moi, s'il vous plaît ce que je dois faire.* »
  - « *voila j ai fais la moitier des fichiers tu me dis si sa marche* »
- des expressions implicites de demande :
  - « *Par consequent je n'ai toujours aucun accès à cette archive* »
  - « *C'est bon, il ne me manque plus que l'accès à l'intranet.* »
- des expressions nominalisées :
  - « *Je vous remercie de votre retour pour le vendredi 27 mars.* »
- des segments courts ou qui sortent des schémas fréquents :
  - « *A vous de faire le tri maintenant…* »
  - « *on attend tes fichiers* »
  - « *Pas de PJ.* »
- l'utilisation de terminologie peu fréquente dans les courriels en français :
  - « *TO DO - je ne me sens pas vraiement qualifié à répondre à ce point* »
  - « *A terminer ASAP* »

Les faux positifs ont 3 causes principales.

- Certains segments expriment effectivement un besoin, mais ce besoin n'est pas adressé à qui que ce soit en particulier :
  - « *il faudrait arriver à factoriser ces codes, qui sinon devront être dupliqués dans tous les programmes implémentant des phases.* »
  - « *Pour la nouvelle version du papier il faudra prendre en compte les derniers commentaires qui sont de loin les plus articulé* »
- D'autres sont des formules de politesse ou n'expriment pas une demande qu'on pourrait considérer comme une tâche : « *La mise à jour de la liste de nos adresses étant en cours, il se peut que vous receviez ce courrier par erreur : veuillez nous en excuser* »
- D'autres avaient été jugés comme n'exprimant pas une tâche, jugement qui est subjectif et qui souligne l'importance d'avoir plusieurs annotateurs.

La Figure 39 ci-dessous montre un exemple de restitution du résultat du repérage. Ici, nous avons une liste à puces avec une phrase d'amorce. (L'annexe 5 donne quelques détails techniques sur le repérage de structures énumératives.) Les segments identifiés comme des tâches sont en vert. Ici, l'identification est binaire : soit le segment contient une tâche, soit il n'en contient pas.


Sujet : fin du déménagement
De : François
Date : 09/05/2016 16:40
Pour :

Le déménagement du 3e et 2e est en train d'être terminé, vous pouvez donc revenir demain Mardi 10 mai au nouveau bâtiment .

AVANT toute chose, merci de :
- vérifier que le mobilier et vos affaires ont bien été transférées
- faire les échanges de mobilier si nécessaire (me demander d'abord)
- rendre vos clés et badges à Michel si cela n'est déjà fait
- nettoyer le mobilier autant que possible
- remplir vos armoires
- rebrancher et redémarrer les machines
- modifier vos informations d'accès (p.ex., ma signature est à jour)

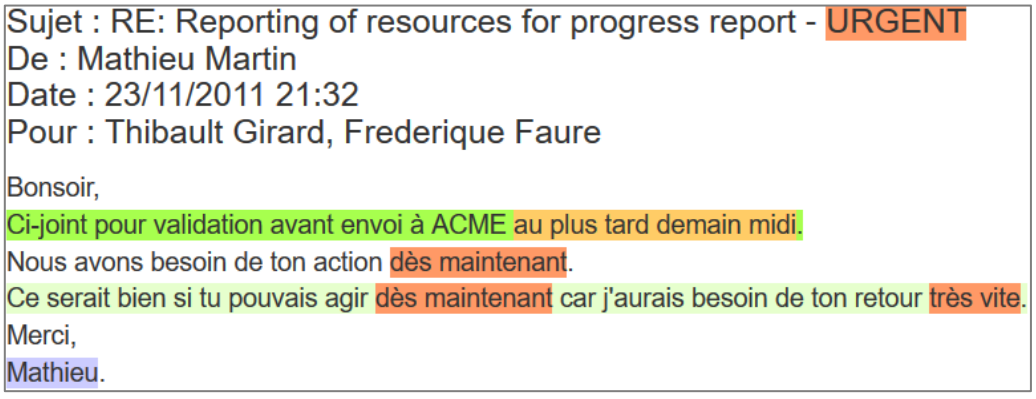
merci à tous pour votre contribution, j'espère que vous apprécierez vos nouveaux bureaux !

François

Figure 39 : exemple de repérage avec une structure énumérative

Dans la Figure 40, nous avons un autre exemple d'affichage : nous attribuons un score de confiance à chaque segment, et ce score est reflété par l'intensité du surlignage. Nous voyons que le premier segment vert est très visible, le deuxième (« nous avons besoin ») aurait dû l'être mais ne l'est pas, et le troisième est en vert pale.

Remarquons que ces trois segments font référence à la même tâche, et le fait qu'on n'ait pas repéré le deuxième n'est pas problématique, car la tâche a tout de même été mise en évidence avec deux autres segments.


Sujet : RE: Reporting of resources for progress report - URGENT
De : Mathieu Martin
Date : 23/11/2011 21:32
Pour : Thibault Girard, Frederique Faure

Bonsoir,
Ci-joint pour validation avant envoi à ACME au plus tard demain midi.
Nous avons besoin de ton action dès maintenant.
Ce serait bien si tu pouvais agir dès maintenant car j'aurais besoin de ton retour très vite.
Merci,
Mathieu.

Figure 40 : exemple de repérage avec des couleurs nuancées pour les segments

III.3.3 Repérage d'attributs

Nous nous intéressons maintenant au repérage des attributs de tâche dans le texte. Ceux auxquels on s'intéresse sont « *émetteur* », « *attributaire* », « *marqueur d'urgence* », « *contrainte temporelle* » (principalement des échéances, car nous n'avons identifié que 4 « *dates de début* ». Nous avons choisi ceux-là car ils sont les plus importants.

Il s'agit ici de retrouver dans le corps d'un message, et souvent dans les messages précédents, des informations nécessaires pour remplir les valeurs définies dans le modèle de la tâche.

Certaines informations utiles caractérisent la tâche, comme son niveau d'urgence ou d'importance, et ne sont pas toujours exprimées dans le message. Parfois, elles peuvent en être déduites, par exemple le niveau d'urgence, qu'on pourrait estimer à partir de la date d'envoi du message et d'une date limite qui apparaîtrait dans le texte.

Une difficulté est alors l'interprétation des fragments textuels qui ont été identifiés comme contenant les valeurs qu'on veut extraire. Les contraintes temporelles ne sont pas toujours exprimées sous forme de dates ; les personnes raccourcissent leurs noms, etc.

III.3.3.1 **Description de l'expérience**

III.3.3.1.1 Approches utilisées

Nous avons deux grandes catégories d'entités à identifier : les personnes et les expressions temporelles. La Figure 40 affiche les entités « échéance », « marqueur d'urgence » et « émetteur ».

Nous nous appuyons de nouveau sur l'outil P-LING de reconnaissance d'entités nommées de Cédric Lopez, développé à VISEO. Cet outil, écrit en JAVA, se base sur les sorties d'analyseurs morphosyntaxiques tels que TALISMANE[106], HOLMES[107] ou SYNAPSE[108] pour extraire un grand nombre d'entités nommées : noms de personnes, dates, noms de marque, et autres.

Pour l'identification des acteurs, nous considérons, par défaut, que l'émetteur d'un message est aussi l'émetteur des tâches qu'il contient, et que les destinataires sont les attributaires de ces tâches. Cependant, il est possible que le texte des messages permette d'attribuer les tâches plus précisément.

La Figure 41 montre que, bien que le gros du courriel s'adresse à une certaine personne, une des tâches peut être assignée à une autre personne, qui est explicitement interpellée par le segment qui contient cette tâche.

Date: Wed, 29 May 2013 15:06:52 +0200
To: Nicolas DUPONT
From: Christian GUERIN
Subject: [M1-CALC] [URGENT] quand se revoir pour votre examen?
Cc: Romain MOREAU, Cécile COEUR

Bonjour Nicolas, 29/5/13
ça fait presque une semaine que je vous ai écrit...
Je n'ai eu qu'une erreur en retour (@ UJF), celle des Centraliens marche donc.
Je vous signale que vous n'aviez pas corrigé ".com" en .org" sur les feuilles de présence (que je viens de vérifier), mais seulement oralement.
Il me faut au moins votre document, et avant vendredi midi.
Ensuite, je pars une semaine, et j'ai peur que le jury ne soit très bientôt.
==> Cécile, pourrais-tu l'appeler au cas où il n'ai pas pu lire cette boit mel?
Merci d'avance.
Cdlt,
CG

Christian Guérin (Pr. Université Joseph Fourier, coordinateur RI à l'UFR IM2AG)

Figure 41 : exemple de sortie de repérage de tâches et des attributs

Les annexes 1 et 2 montrent des exemples de contraintes temporelles et de marqueurs d'urgence extraits du corpus de développement.

[106] http://redac.univ-tlse2.fr/applications/talismane/talismane.html
[107] http://www.ho2s.com/
[108] http://www.synapse-developpement.fr/

III.3.3.1.2 Outils utilisés

Nous combinons la recherche de noms par l'outil P-LING avec les listes de noms extraites des en-têtes des messages. Ces noms sont normalisés (désaccentués, décapitalisés, sans traits d'union ni apostrophes).

De même, pour les contraintes temporelles, nous utilisons P-LING, qui implémente une classe TIME, qui sert au repérage des expressions temporelles. Ce repérage se fait itérativement. Lors d'une première passe, l'outil repère des expressions de base, dont :

- des mots désignant des heures : midi, minuit.
- des numéros de jours cardinaux et ordinaux : 1 – 31, 1er, etc.
- des noms de jours : lundi – dimanche.
- des noms de mois : janvier – décembre, jan. – dec.
- des numéros de mois : 1 – 12, 01 – 12.
- des unités de temps : seconde, minute, heure, jour, semaine, mois, etc.
- d'autres indicateurs :
    - avant-hier, hier, aujourd'hui, demain, etc.
    - début, fin, milieu, matinée, soirée, journée, etc.
    - suivant, après, dernier, avant, précédent, antérieur, ultérieur, prochain, prochaine.

Une fois ces éléments identifiés, les itérations suivantes servent à les combiner en expressions plus longues, ce qui, par exemple, fait passer de la séquence d'items

« `dimanche` », « `premier` », « `octobre` », « `2017` »

à

« `dimanche premier octobre 2017` ».

On la transforme ensuite en en une expression normalisée (ici « `01/10/2017` ») et on déterminer si cette date est une borne maximum ou minimum, en utilisant le contexte, par repérage d'expressions du type « `à partir d(e|u|es) DATE` », « `jusqu'(à|au) DATE` », « `du DATE au DATE` » ou « `DATE dernier délai` », « `DATE au plus tard` », etc. Les expressions ainsi typées sont rattachées à la tâche dont le segment les contient, sinon au segment-tâche qui les précède, sinon au segment-tâche qui les suit.

Pour le repérage des personnes, nous avons développé des règles simples basés sur la position des noms dans le texte :

Pour l'émetteur, c'est le nom qui suit une expression de fin de courriel comme « cordialement », ou le dernier nom avant la signature, ou le premier nom dans la signature.

Pour l'attributaire, c'est un ou plusieurs noms qui commencent une phrase exprimant une tâche, ou alors un ou plusieurs noms suivant les salutations du début de courriel (y compris les mots comme « tous »).

III.3.3.2 **Déroulement et mise en œuvre**

Nous avons procédé à 4 expériences, visant chacune à extraire un attribut différent. Pour chacune, nous avons développé un ensemble de 10 à 15 (pour les marqueurs d'urgence) règles, et nous avons utilisé le même corpus de test que pour le repérage de segments.

Pour chaque attribut, la précision et le rappel sont la moyenne de la précision et du rappel des 4 résultats obtenus. Ici aussi, nous donnons entre parenthèses les scores obtenus lorsqu'on tient compte du fait qu'une entité peut être référencée plusieurs fois dans le texte (comme sur la Figure 38).

III.3.3.3 **Résultats et évaluation sur 4 attributs**

Afin d'avoir une idée de la performance de l'approche dans les meilleures conditions, ici nous donnons les résultats du repérage des attributs en supposant que les segments référant à des tâches ont été correctement repérés. Cela nous permet d'étudier les raisons des erreurs inhérentes à l'approche.

4 attributs visés	**P**	**R**	**F1**
Émetteur	0,84	0,63 (0,66)	0,72 (0,74)
Attributaire	0,75	0,61 (0,62)	0,67 (0,68)
Contrainte temporelle	0,69	0,66	0,67
Marqueur d'urgence	0,9	0,85	0,87

Tableau 15 : résultats du repérage de 4 attributs de tâche

La performance est imparfaite à cause de nombreuses difficultés.

- Il est fréquent que les noms des personnes ne soient pas écrits de la même manière que dans les en-têtes, mais soient abrégés, parfois d'une façon non triviale :
  - Jean-Claude → *JC,*
  - Christian → *Xan*,
  - Frédérique → *F*,
  - Lingxiao → *LX.*

  En ce qui concerne l'émetteur, étant donné le rappel obtenu, il est meilleur de partir de l'hypothèse que l'émetteur des tâches d'un courriel est l'expéditeur du courriel, ce qui est vrai dans 92% des cas.
- Parfois, bien que les destinataires d'un message soient listés dans les en-têtes, le texte réfère à un sous-ensemble, que nous ne savons pas déterminer automatiquement :
  - *Quelqu'un.*
  - *partenaires industriels.*
  - *les personnes qui s'occupent des serveurs.*
  - *ceux qui disposent des originaux.*
  - *ceux qui à ce jour, n'ont pas remis leur rapport de suivi de thèse pour la ré-inscription 2015-2016.*
  - *Un d'entre vous.*
  - *GETALP.*
- Pour les contraintes temporelles, il arrive qu'une phrase en exprime plusieurs :
  - « *aujourd'hui, demain matin au plus tard* ».
  - « *d'ici lundi après-midi (ou, au pire, mardi matin)* ».
- Parfois, la contrainte temporelle n'est pas exprimée comme une date :
  - « *dès qu'elles en auront connaissance* ».
  - « *la veille du séminaire* ».
  - « le moment venu »
  - « quand la fiche sera imprimée »
- Certains marqueurs d'urgence, comme « *ASAP* », « *dqp !* », « *dernier carat* », ne sont présents que dans le corpus de test. D'autres contiennent des erreurs d'orthographe (« *c'esturgent* »). D'autres encore sont exprimées dans un courriel différent de celui qui contient l'énoncé auquel elles doivent être rattachées. D'autres encore sont rattachés à une seule tâche, au lieu de plusieurs.

III.3.4 Vers un vrai traitement des tâches

III.3.4.1 Positionnement du problème

Après le repérage, on imagine un processus en 5 passes pour arriver au but final décrit plus haut :

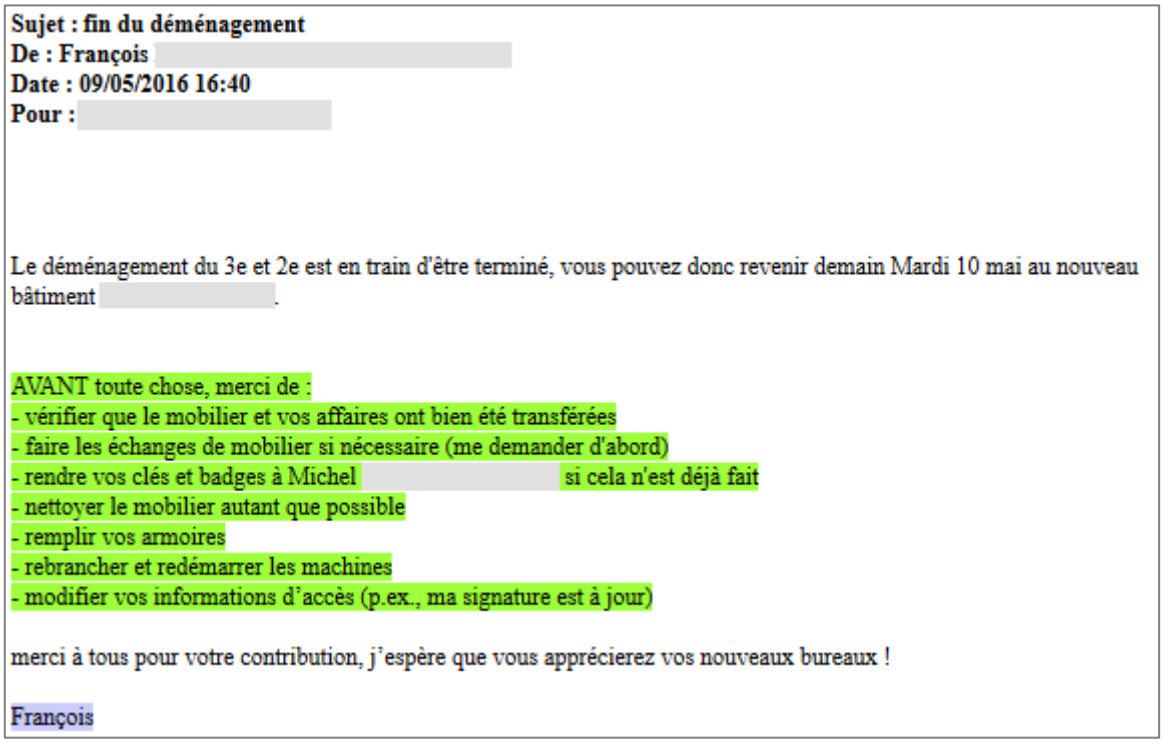
(1) typage des énoncés,

(2) nommage des tâches,

(3) relations entre tâches (sous-tâche, exclusivité, parallélisme, dépendance, séquencement),

(4) hypergraphe des tâches,

(5) interprétation dans la KB ou dans l'ontologie du projet.



Sujet : fin du déménagement
De : François
Date : 09/05/2016 16:40
Pour :

Le déménagement du 3e et 2e est en train d'être terminé, vous pouvez donc revenir demain Mardi 10 mai au nouveau bâtiment .

AVANT toute chose, merci de :
- vérifier que le mobilier et vos affaires ont bien été transférées
- faire les échanges de mobilier si nécessaire (me demander d'abord)
- rendre vos clés et badges à Michel si cela n'est déjà fait
- nettoyer le mobilier autant que possible
- remplir vos armoires
- rebrancher et redémarrer les machines
- modifier vos informations d'accès (p.ex., ma signature est à jour)

merci à tous pour votre contribution, j'espère que vous apprécierez vos nouveaux bureaux !

François

Figure 42 : point de départ pour le typage

Nous voyons dans la Figure 42 ci-dessous un ensemble de tâches, que nous pouvons nommer, résumer et typer ainsi :

Id	Description	Type
T1	Revenir à l'IMAG	Physique
T2	Vérifier que le mobilier et les affaires ont bien été transférés	Physique
T3	Faire les échanges de mobilier si nécessaire	Physique
T4	Rendre les clés et badges à Michel Vacher.	Physique
T5	Nettoyer le mobilier autant que possible	Physique
T6	Remplir les armoires	Physique
T7	Rebrancher et redémarrer les machines	Physique
T8	Modifier les informations d'accès	Informatique : modifier un document

Tableau 16 : possibilité de nommage et de typage des tâches de la Figure 42

On peut alors envisager le graphe de séquencement suivant pour ces tâches :

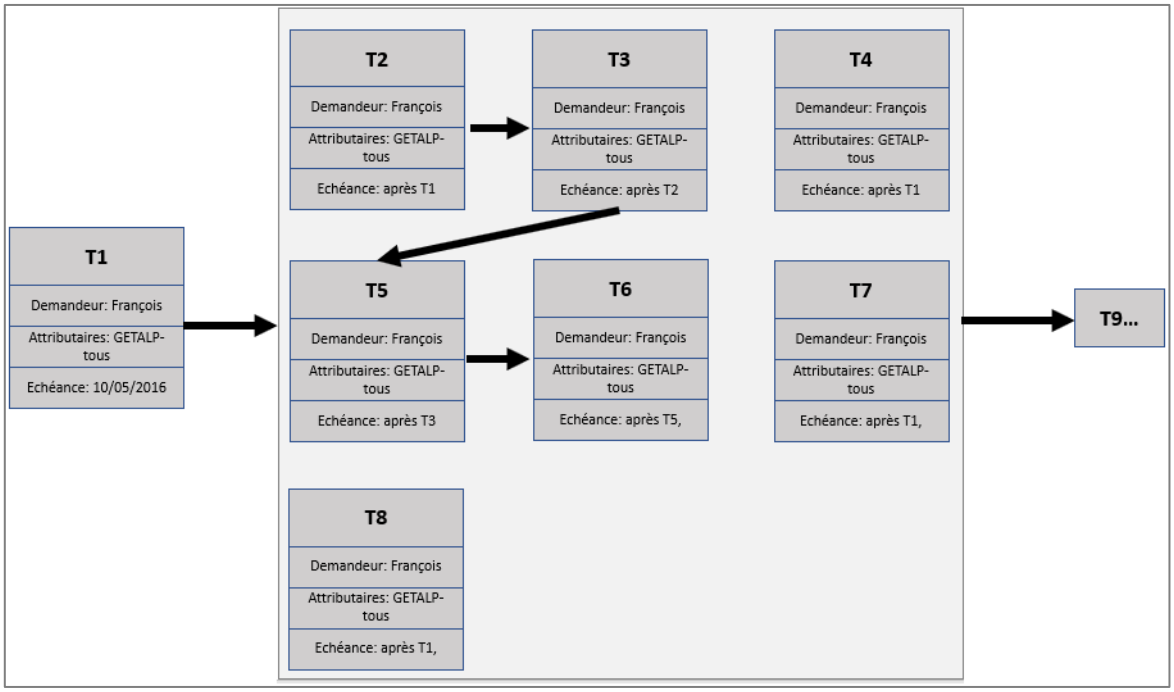


Figure 43 : exemple de séquencement possible pour les tâches de la Figure 42

III.3.4.2 **Analyse du problème et méthodes possibles**

III.3.4.2.1 Analyse du problème

Pour nommer une tâche, il faut savoir extraire la description de l'action ou des actions à réaliser à partir des énoncés qui lui sont associés. Pour l'instant, nous ne sommes pas en mesure d'extraire fiablement les informations de séquencement entre les tâches.

III.3.4.2.2 Méthodes possibles (en combinaison)

Nous avons prévu de combiner plusieurs règles expertes, en utilisant des règles P-LING et d'autres programmées directement en Java. En voici quelques-unes.

- Utilisation de la typographie : nous avons défini un ensemble d'introducteurs d'éléments d'énumération, tels que trouvés dans les courriels, comme `1.`, `1)` , `-1-`, `-`, etc. La liste complète est donnée en Annexe 5.
- Utilisation de lexiques constitués pour identifier le type d'action (`fournir_info`, `envoyer_document`, `travailler_sur_document`) : lexiques de verbes, lexique relatif aux documents.
- Utilisation de marqueurs de relations temporelles, comme `avant_de`, `avant_le`, `après_le`, grâce à des expressions écrites en P-LING.

III.3.4.2.3 Méthodes d'évaluation possibles

Pour l'identification (réduite ici au nommage) proprement dite des tâches, nous pourrions utiliser simplement le rappel et la précision (ainsi que la F-mesure qui en est déduite) sur les tâches telles qu'identifiées dans notre corpus annoté. Cela nous permettrait d'approcher la « qualité d'usage » mieux qu'en utilisant une mesure pénalisant le fait que certains énoncés ne seraient pas rattachés à une tâche par ailleurs identifiée.

Pour les relations, en particulier les relations de citation (sous-document) et les relations temporelles, nous pourrions considérer le graphe dont les nœuds sont les tâches et les arcs les relations, et calculer aussi la précision et le rappel. Ce n'est sans doute pas idéal, mais c'est ce

que ce qui est fait dans le domaine, par exemple pour calculer la qualité de structures de dépendances syntaxiques ou sémantiques.

Synthèse de ce chapitre

Dans ce chapitre, nous nous sommes intéressé au traitement sémantique de courriels autour de projets collaboratifs, et plus particulièrement au repérage de tâches dans ces textes.

Nous avons constitué un corpus de courriels professionnels, que nous avons annotés. Les expériences de repérage que nous avons mené produisent des résultats qui sont essentiellement comparables aux performances de l'état de l'art menées dans les études en anglais.

Nous n'avons pas encore décidé de la meilleure façon de restituer le résultat du repérage. Une façon qui paraît utile est la surbrillance telle qu'elle est présentée sur les illustrations. On peut également envisager de constituer une *todo list*.

Chapitre IV Traitement de documents complexes en contexte d'apprentissage actif

Introduction

Nous présentons nos travaux sur la plate-forme MACAU-CHAMILO, qui aide à l'apprentissage par (1) structuration de documents pédagogiques selon deux ontologies (forme et contenu), et (2) accès multilingue à du contenu initialement monolingue. Il s'agit donc de nouveau de structuration selon les deux axes, forme et sens.

(1) L'ontologie des formes permet d'annoter les fragments des documents par des concepts comme *théorème, preuve, exemple*, par des niveaux de difficulté et d'abstraction, et par des relations comme `élaboration_de`, `illustration_de`. L'ontologie de domaine modélise les objets formels de l'informatique, et plus précisément les notions de complexité calculatoire. Cela permet de suggérer aux utilisateurs des fragments utiles pour la compréhension de notions d'informatique perçues comme abstraites ou difficiles.

(2) L'aspect relatif à l'accès multilingue a été motivé par le constat que nos universités accueillent un grand nombre d'étudiants étrangers, qui ont souvent du mal à comprendre nos cours à cause de la barrière linguistique. Nous avons proposé une approche pour multilingualiser du contenu pédagogique avec l'aide d'étudiants étrangers, par *post-édition* en ligne de pré-traductions automatiques, puis, si besoin, amélioration incrémentale de ces post-éditions. (Nos expériences ont montré que des versions multilingues de documents peuvent être produites rapidement et sans coût.) Ce travail a abouti à un corpus de plus de 500 pages standard (250 mots/page) de contenu pédagogique post-édité vers le chinois, et c'est par cet aspect que nous ouvrons ce chapitre.

IV.1 Le projet MACAU

La partie de l'UGA correspondant à l'ex UJF[109] accueille chaque année environ 2300 étudiants étrangers. De même, environ 650 de nos étudiants font une partie de leurs études à l'étranger grâce à des programmes comme Erasmus.

Il est évident que leur réussite dépend fortement de leur maîtrise d'une langue étrangère (que ce soit le français ou l'anglais, sachant que 98% n'ont pas l'anglais comme langue maternelle, ni comme langue d'éducation), ce qui est une difficulté que n'ont pas les locuteurs natifs.

Certains cherchent des livres en anglais, mais ils savent souvent l'anglais encore moins bien que le français (nous l'avons observé avec des étudiants chinois), et ces livres utilisent des conventions différentes de celles utilisées dans leurs UE, et ne couvrent pas les mêmes choses au même niveau de détail.

Il leur faudrait disposer de supports dans leur langue, en phase avec les cours et TD de l'université d'accueil.

Nous avons vu que les étudiants et les enseignants produisent des documents de types, qualités, formats, complétudes et niveaux de formalisme différents. Nous souhaitons structurer ces productions et nous en servir pour générer à partir des "meilleurs morceaux" des documents plus complets, mais aussi des documents adaptés aux préférences des utilisateurs.

[109] L'UJF (Université Joseph Fourier) s'appelait avant 1987 USTMG (Université Scientifique, Technologique et Médicale de Grenoble). L'UGA est née début 2016 de la fusion entre l'UJF, l'U-Stendhal et l'UPMF.

IV.1.1 Motivations et historique

Motivé par ce constat, le projet MACAU (Kalitvianski et al., 2012), lancé en 2012 au sein de l'équipe GETALP du LIG, a pour but, entre autres, de donner un accès multilingue à du contenu pédagogique produit non seulement par nos enseignants, mais aussi par nos étudiants, qui produisent des documents qui peuvent être réutilisés (comptes rendus, notes de cours, diaporamas, sujets et corrigés d'examens, rapports de stage...).

Une approche naïve à l'accès multilingue consisterait à utiliser un service de traduction automatique (TA) gratuit et en ligne, tel que GOOGLE TRANSLATE (GT), sans autre traitement. GT offre un large choix de paires de langues, mais présente des problèmes importants.

1. La qualité des traductions, bien que tout à fait acceptable pour les courtes phrases conversationnelles, se détériore pour les domaines techniques spécialisés et avancés qui sont enseignés dans notre université, ainsi que pour de nombreuses paires de langues et pour des phrases plus longues.
2. Bien que GT permette d'apporter des corrections aux traductions, ces corrections ne sont pas affichées lors des visites subséquentes de la page. Elles sont stockées dans la mémoire de traductions de GOOGLE et utilisées pour réentraîner périodiquement leurs systèmes de TA empirique (statistique, et depuis peu neuronale).
3. GT nécessite une URL vers un dépôt de fichiers où les documents pédagogiques seraient stockés.

Bien que les traductions automatiques brutes aient une utilité limitée pour l'accès multilingue au contenu pédagogique, elles peuvent être très utiles pour accélérer la traduction humaine (Green et al., 2013).

L'approche adoptée dans le projet MACAU décrit dans chapitre est de donner immédiatement accès au matériel pédagogique (formaté en html) dans la langue d'accès souhaitée, en utilisant des « pré-traductions » de TA, puis d'améliorer la qualité des segments cibles de manière incrémentale et contributive.

En d'autres termes, si une personne n'est pas satisfaite de la traduction proposée, elle peut la corriger directement sur la page Web de manière transparente.

IV.1.1.1 État de l'art

Il semble y avoir peu de travaux préalables. Deux projets qui ressortent sont le projet européen BOLOGNA[110] et le projet contributif SLIDEWIKI[111] plus récent.

SLIDEWIKI (Tarasowa et al., 2013) est un projet visant à la construction collaborative en ligne de présentations éducatives. Ces présentations peuvent être construites sur le site Web ou être importées d'un format PPTX. Un aspect intéressant est la possibilité de produire des versions dans une langue différente en utilisant Google Translate. Cependant, limiter le type de contenu aux présentations semble trop restrictif et l'absence de mémoire de traductions limite l'efficacité du processus de traduction.

Le projet BOLOGNA était une initiative financée par l'UE visant à construire « *un service de traduction destiné à traduire des programmes de cours et d'études de 9 langues (néerlandais, anglais, finnois, français, allemand, portugais, espagnol, suédois et turc) vers l'anglais* », en utilisant des outils de traduction assistée par ordinateur. Le chinois a été ajouté plus tard, car les étudiants chinois sont souvent les plus nombreux parmi les étudiants étrangers.

[110] http://www.bologna-translation.eu/
[111] https://slidewiki.org/

Ce projet de trois ans s'est terminé en 2013 et offrait un démonstrateur de la plate-forme Web collaborative, mais il n'a pas débouché sur un service Web permanent. Les outils de traduction devaient être spécifiquement adaptés à la traduction des programmes de cours.

En 2013, nous avons évalué le démonstrateur en ligne de BOLOGNA et constaté que ses traductions étaient d'une qualité inférieure à celle de GOOGLE TRANSLATE. Ce service a été interrompu depuis.

Plusieurs idées sous-jacentes au projet BOLOGNA convergent avec celles de notre projet :

- une approche collaborative à la traduction et à son amélioration.
- une utilisation de mémoires de traductions spécialisées dans chaque contexte.
- une définition des rôles et des tâches, tels que traducteur, post-éditeur, modérateur, développeur de TA, etc.
- une gestion de différents formats (HTML, DOCX, XSLX, TXT, RTF, liens URL).

Cependant, BOLOGNA avait des défauts à la fois conceptuels et de mise en œuvre.

- Le projet manquait d'ambition, car il se limitait à traduire 9 des 22 [112] langues européennes en anglais et en chinois. Les étudiants internationaux arrivant dans un pays étranger sont pour la plupart non anglophones et ne maîtrisent pas suffisamment l'anglais pour comprendre réellement les documents traduits, et encore moins pour contribuer à l'amélioration des pré-traductions, puisqu'on devrait toujours post-éditer dans sa langue maternelle.
- Les systèmes de TA testés produisaient des résultats de qualité inférieure. Cela est dû à l'utilisation de la TA statistique, qui ne peut produire de résultats utiles que si elle est construite à partir d'un grand corpus de traductions parallèles de bonne qualité.
- L'accès à l'interface de post-édition était limité aux utilisateurs approuvés, et l'interface était elle-même lourde. Afin d'obtenir des contributions, l'interface aurait dû être directement disponible pour la post-édition sur le document affiché, et être librement accessible.

IV.1.1.2 **Genèse du projet MACAU**

Le projet MACAU a d'abord été formalisé dans notre sujet de stage de M2R. Ce projet (toujours en cours) aborde essentiellement deux problématiques, le multilinguisme et la structuration de documents. En 2012-2013, les bases de la plate-forme ont été posées.

- Une première version de SEGNORM, appelé alors SEGDOC, a été développée.
- L'accès multilingue via une IMAG a été intégré dans une plate-forme CHAMILO.
- Des documents de cours ont été collectés auprès d'enseignants et d'étudiants.

À la fin de notre M2R (été 2013), nous avons embauché deux stagiaires chinois afin de post-éditer vers le chinois des documents pédagogiques relatifs à l'informatique. Cela nous a permis de constituer un premier corpus de post-éditions, et de faire des mesures relatives aux temps de post-édition. Les résultats ont été publiés dans (Kalitvianski et al., 2015).

Nous avons poursuivi l'expérience en 2015, grâce à un soutien financier de l'action PEDAGOTICE[113] de l'UJF, ce qui nous a permis d'embaucher deux autres étudiants pour travailler sur le pré-traitement des fichiers et sur la post-édition du français vers le chinois.

Pour l'accès multilingue, nous utilisons une passerelle IMAG. Le concept de passerelle interactive d'accès multilingue (IMAG) a été proposé par Ch. Boitet et V. Bellynck en 2006 et est utilisé dans notre laboratoire depuis novembre 2008 (Boitet et Bellynck, 2008). Une IMAG

[112] Il y a maintenant 23 langues officielles dans l'UE, mais le croate n'a été ajouté qu'en 2013.
[113] http://sup.ujf-grenoble.fr/?q=pedagotice

est une passerelle informatique, à première vue très similaire à Google Translate : on spécifie l'URL d'une page Web et la langue d'accès, puis on navigue dans cette langue d'accès. L'IMAG affiche la page Web traduite, avec conservation de la mise en page. Lorsque le curseur survole un segment (généralement une phrase ou un titre), une palette affiche le segment source et propose de contribuer en corrigeant le segment cible, en post-éditant un résultat de TA.

Contrairement à GT, une IMAG est dédiée à un site Web élu, ou plutôt au sous-langage élu défini par une ou plusieurs URL et leur contenu textuel. Elle dispose d'une mémoire de traductions dédiée (MT). Les segments sont prétraduits non pas par un système de TA unique, mais par un ensemble (sélectionnable) de systèmes de TA. SYSTRAN et GOOGLE TRANSLATE sont principalement utilisés maintenant, mais des systèmes spécialisés développés à partir de la partie post-éditée de la MT ont également été utilisés, notamment pour le français → chinois.

Lors de la lecture d'une page traduite, il est possible non seulement de contribuer au segment sous le curseur, mais aussi de basculer dans un environnement de post-édition en ligne avancé, doté de fonctions d'aide proactive, de filtrage et de recherche et remplacement, puis de revenir au contexte de lecture.

Un intergiciel de TA, TRADOH, nous permet de sélectionner, paramétrer et appeler les systèmes de TA et de définir les chemins de traduction utilisées pour les différentes paires de langues. Un relais IMAG est prévu pour gérer les utilisateurs, les groupes, les projets (certaines contributions peuvent être organisées, d'autres opportunistes) et les droits d'accès. Pour le moment, ces fonctions sont gérées par SECTRA_W (Huynh et al., 2008), gestionnaire "dorsal" (*back-end*) de MT et de corpus. Des systèmes de TA adaptés aux sous-langages sélectionnés peuvent être construits et ont été construits (par des combinaisons de méthodes empiriques et expertes) à partir de la MT dédiée à un site Web élu donné. Cette approche augmente intrinsèquement la qualité linguistique et terminologique des résultats de TA, qui peuvent parfois être des traductions grossières plutôt que brutes[114] (Wang, 2015).

(Besacier, 2014) rapporte une expérience de traduction collaborative en français d'un petit roman en anglais[115] via une IMAG. Dans cette expérience, qui impliquait des traducteurs non professionnels, il a montré que des traductions gratuites de textes littéraires de qualité acceptable peuvent être produites relativement rapidement par des volontaires par post-édition, même si ces traductions présentent initialement un certain manque d'unité et des insuffisances stylistiques typiques du travail d'un traducteur débutant. Pour notre propos, cependant, les considérations stylistiques sont moins pertinentes.

IV.1.1.3 **Expérimentation sur plus de 500 pages**

IV.1.1.3.1 Spécifications

Notre objectif est de fournir une plate-forme permettant aux utilisateurs de téléverser leurs documents, et d'accéder à ces documents dans les langues de leur choix. Les versions traduites doivent préserver la mise en page du document original, ainsi que permettre aux utilisateurs d'éditer les traductions si nécessaire, de manière collaborative et incrémentale, grâce à une interaction directe avec les segments concernés.

Les apprenants en langues trouvent utile de voir simultanément le texte original et sa traduction. Cela leur permet d'apprendre les correspondances de phrase à phrase entre les langues. SECTRA permet d'afficher en parallèle la page Web source et sa traduction.

[114] Les termes *rough translation* et *raw translation* ont été introduits par Hans Karlgren lors de son séjour au GETA (Grenoble) en 1986.
[115] *The Book of Me*, de Richard Powers.

Bien que l'accès au contenu doive être ouvert à tous, une politique de gestion des droits pour la post-édition peut être mise en œuvre. Les IMAGs peuvent être configurées avec plusieurs modes de contrôle ou "modération", un peu comme WIKIPEDIA.

Pour contenir les documents pédagogiques, nous utilisons une plate-forme d'apprentissage en ligne open-source CHAMILO, dont des instances sont largement utilisées par nos universités[116]. Elle dispose d'une interface multilingue, et permet aux utilisateurs de créer des cours soit en téléversant des documents HTML existants, soit en les construisant en ligne via un éditeur HTML WYSIWYG. Elle permet également de communiquer via des forums ou des messages instantanés, ainsi que de définir des dictionnaires.

Nous avons ainsi mis en place une plate-forme CHAMILO et l'avons dotée d'outils de sélection de la langue d'accès d'un document. La liste des langues est définie par les systèmes de TA disponibles ou les mémoires de traductions[117].

La langue d'accès par défaut d'un document est sa langue d'origine. Pour y accéder dans une autre langue, on la sélectionne dans le menu "AXIMAG" et on clique sur "Traduire". La page de cours qui en résulte est reconstituée à partir des résultats de TA ou des post-éditions disponibles, qui sont toutes stockées dans une MT (mémoire de traduction) gérée par l'IMAG-MACAU.

Cela fonctionne pour tout contenu HTML disponible sur MACAU-CHAMILO, qu'il ait été créé à l'aide des outils de la plate-forme ou téléchargé par un utilisateur. Il convient de noter que nous ne sommes pas limités à CHAMILO : les IMAG peuvent être facilement intégrées à n'importe quelle autre plate-forme qui fournit une URL à son matériel de cours.

Figure 44 : illustration de l'intégration de l'accès multilingue à une plate-forme Chamilo

Pour un cours qui n'a pas encore été édité, la première traduction est obtenue par TA. L'utilisateur peut corriger la traduction via la palette qui apparaît lorsque le curseur survole les phrases, ou en mode avancé (cf. IV.1.1.2), et ces corrections sont sauvegardées dans la mémoire

[116] Par exemple, http://chamilo.univ-grenoble-alpes.fr/

[117] Il peut arriver que, comme pour le grand comorien (shingazidja), il y ait une MT vers une langue sans qu'il existe de système de TA vers cette langue (Abdourahamane et al., 2016)

de traductions gérée par le dorsal SECTRA_W de l'IMAG. Le niveau de fiabilité[118] (attribué par le système) et le score de qualité (attribué par l'utilisateur) sont utilisés pour classer les traductions et les post-éditions dans la mémoire ; la post-édition ayant le score le plus élevé est affichée lors de la visite de la page via l'IMAG.

Le processus de correction est appelé « post-édition », par opposition à « révision ». La différence est qu'il est absolument nécessaire de lire et de comprendre chaque phrase avant de corriger la « pré-traduction ». C'est pourquoi nous demandons régulièrement aux bons étudiants étrangers dans les filières où nous enseignons (L3 et Master d'informatique) de faire les premières post-éditions.

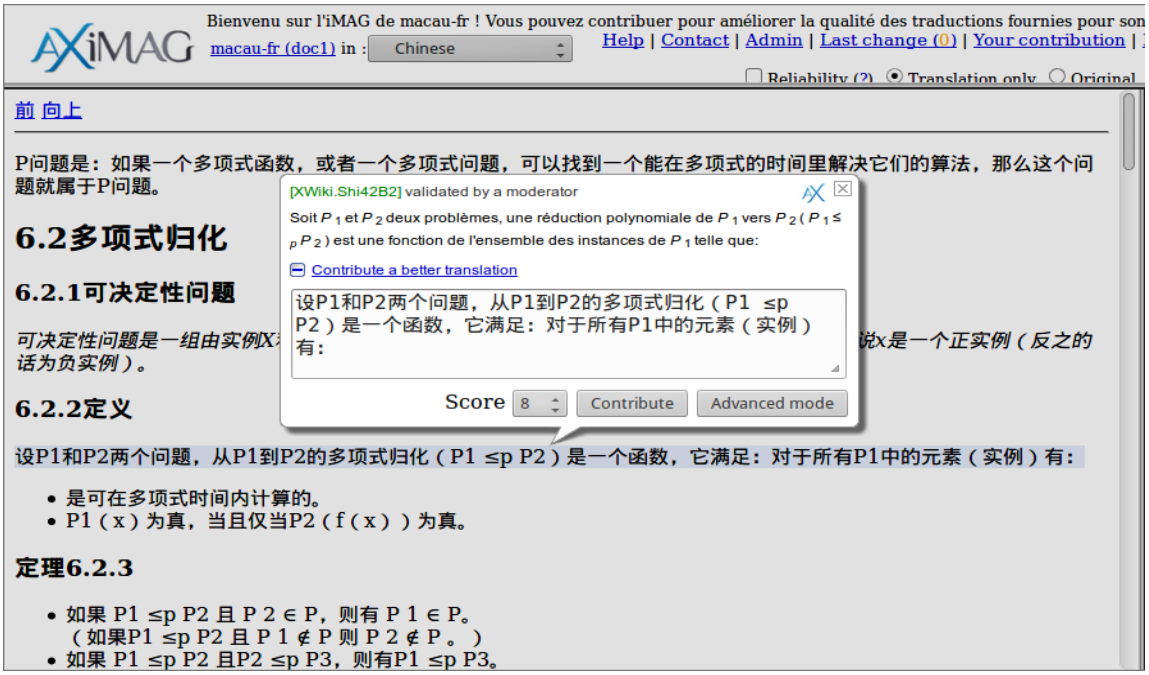


Figure 45 : illustration du processus de post-édition d'un document de cours sur la complexité calculatoire

L'utilisateur peut basculer entre la vue parallèle, qui affiche à la fois la traduction et l'original, et la vue de traduction, qui affiche uniquement la traduction. Les « crochets de fiabilité » optionnels autour des segments permettent de voir en un coup d'œil les post-éditions : les crochets verts indiquent que le segment a été validé par un modérateur, les crochets orange qu'il a été post-édité, mais attend la modération (pour les contributions des utilisateurs qui ne sont pas enregistrés) et les crochets rouges entourent les résultats de TA.

[118] Une note sur cinq étoiles est attribuée :
* : traduction dictionnairique, mot à mot ou « pidgin »
** : traduction automatique
*** : humain bilingue
**** : traducteur professionnel
***** : traducteur agréé par l'organisme créateur de l'information (ex. Unesco, IBM, …)

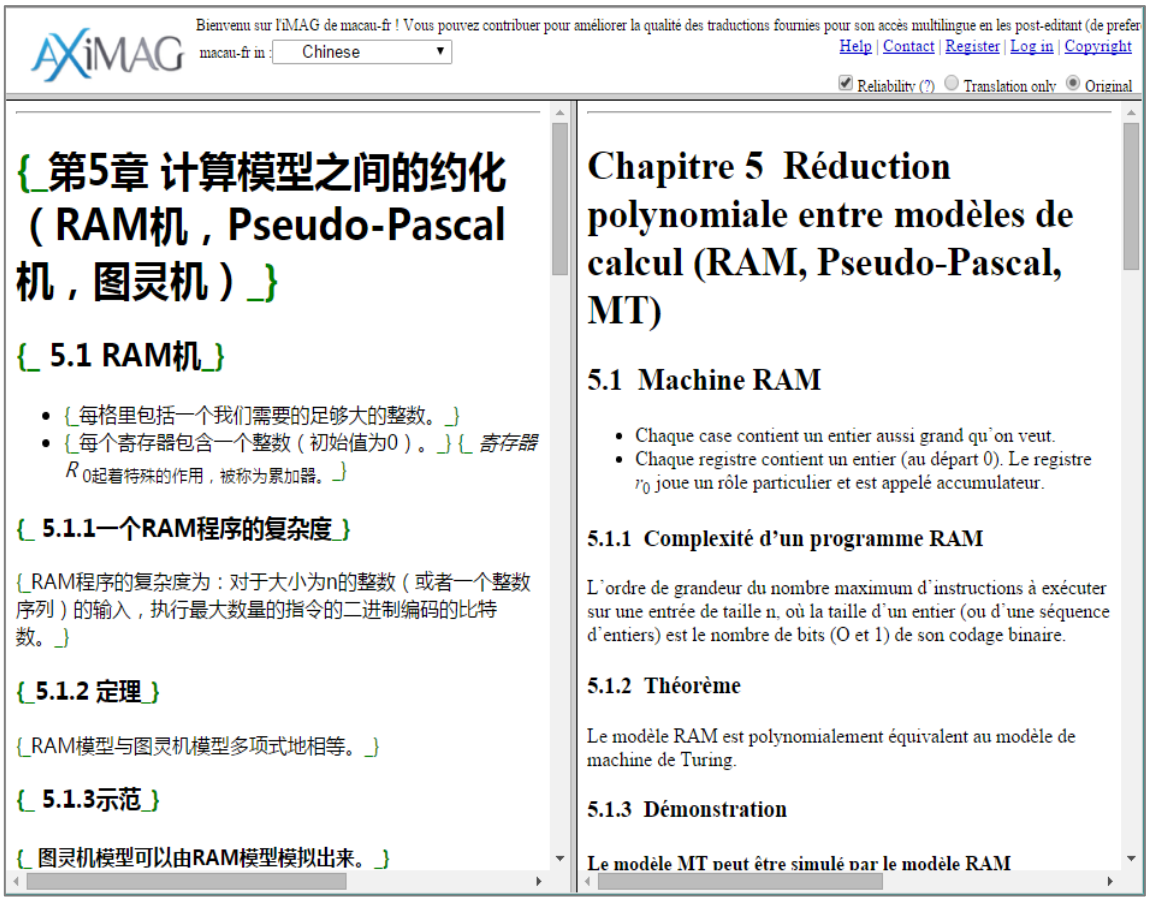


Figure 46 : illustration d'une vue parallèle d'un document pédagogique multilingualisé

Nos expériences confirment jusqu'à présent l'hypothèse que la post-édition de traductions automatiques de contenu pédagogique par des volontaires est une manière viable de produire des versions de qualité suffisante, même lorsque les systèmes de TA utilisés ne sont pas de bonne qualité. La qualité d'usage ne correspond pas nécessairement à la qualité linguistique.

IV.1.1.3.2 Déroulement

La première étape consiste à collecter du contenu pédagogique et à le convertir en HTML. Nous avons recueilli des documents éducatifs sur l'informatique produits par nos enseignants et nos étudiants. Ces documents comprennent un livre sur la logique (*Logique et démonstration automatique* de S. Devismes, P. Lafourcade et M. Lévy[119]), des notes de cours sur la complexité informatique, ainsi que divers polycopiés distribués.

Ces documents sont en différents formats. Le livre et les notes de cours étaient en LaTeX et ont dû être convertis en HTML à l'aide d'outils tels que HeVeA[120] (Maranget, 2003) et LaTeX2HTML[121]. D'autres étaient en format Microsoft DOCX, un format basé sur XML dont la conversion en HTML est simple et bien réalisée par des suites bureautiques telles que Microsoft Office, LibreOffice et AbiWord.

La situation était moins favorable pour les fichiers PDF. Lorsque cette expérience a été réalisée, il n'y avait pas d'outil de qualité acceptable pour convertir les PDF en HTML. Les outils disponibles soit ne faisaient qu'extraire le texte, sans tenir compte de la mise en page du document, soit produisaient des documents HTML en essayant de préserver la mise en page

[119] http://www-verimag.imag.fr/~devismes/INF242/cours_eleve.pdf
[120] http://hevea.inria.fr
[121] http://www.latex2html.org

typographique des pages, mais en produisant un code HTML assez difficile à analyser pour les soumettre aux systèmes de TA. Des progrès ont depuis été réalisés par MICROSOFT WORD, qui peut maintenant transformer certains fichiers PDF en documents WORD. C'est une chance, car certains enseignants ne peuvent nous fournir que des fichiers PDF, et non leurs sources.

Une deuxième étape est la segmentation en pages de taille pratique pour le système de TA que nous avons utilisé (GT dans cette expérience), généralement la taille d'un chapitre. Cette étape a été réalisée automatiquement en utilisant SEGNORM.

Une troisième étape cruciale est la normalisation, qui consiste à sélectionner des parties du flot HTML qui devraient être protégées de la traduction, typiquement des formules mathématiques dans leur transcription alphanumérique, ainsi que du code algorithmique, susceptibles d'être traités comme du texte par les systèmes de TA. Par exemple, il faut éviter qu'une variable nommée '`I`' puisse être interprétée comme un pronom à la première personne, ce qui serait problématique. D'autres littéraux risqueraient d'être supprimés ou inversés par le système de TA, déformant ainsi l'entité, comme l'illustre la Figure 47.

Exemple 1.1.7 *Soient la formule A = (a∨b) et B = (A∧¬A), nous avons l (A) = 5 et l (B) = 4 + 5 + 5 = 14.*

Пример 1.1.7 *Пусть формула = (∨ B) и B = (A ∧ ¬), имеем L (A) = 5 л (B) = 4 + 5 + 5 = 14.*

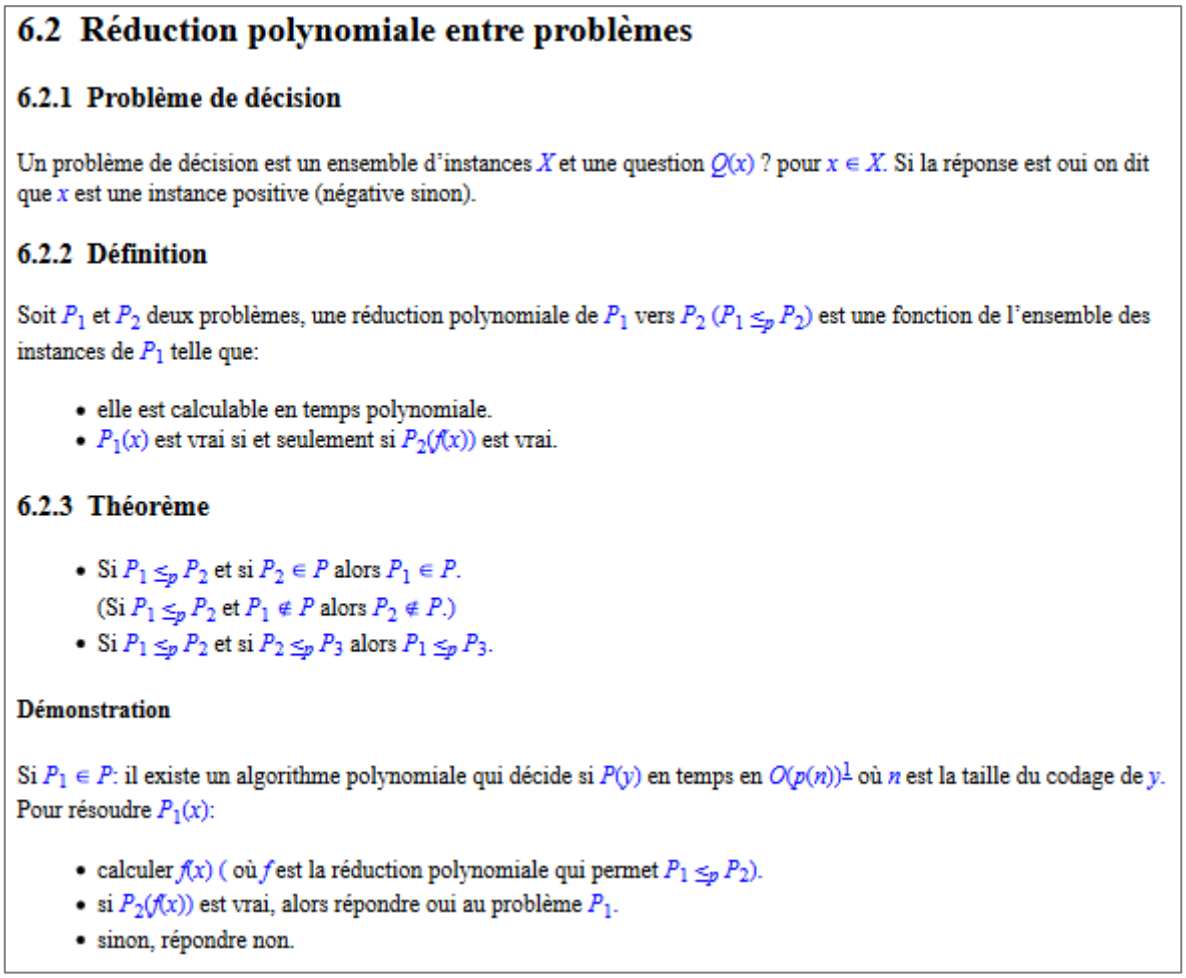
Figure 47 : exemple de formules logiques non protégées déformées par la traduction

La protection de ces sections consiste à insérer l'attribut "`translate = no`", qui fait partie de la norme HTML5[122], dans les balises HTML environnantes.

Au moment de la réalisation de l'expérience, nous n'avions aucun outil automatique pour détecter ces fragments non linguistiques et, dans la plupart des cas, cette étape a dû être faite manuellement.

Depuis, nous avons conçu et intégré dans le normaliseur une méthode qui détecte les formules mathématiques dans certains documents HTML produits par HEVEA. La figure suivante l'illustre.

[122] https://www.w3.org/TR/html5/dom.html#the-translate-attribute



6.2 Réduction polynomiale entre problèmes

6.2.1 Problème de décision

Un problème de décision est un ensemble d'instances X et une question $Q(x)$? pour $x \in X$. Si la réponse est oui on dit que x est une instance positive (négative sinon).

6.2.2 Définition

Soit P_1 et P_2 deux problèmes, une réduction polynomiale de P_1 vers P_2 ($P_1 \leq_p P_2$) est une fonction de l'ensemble des instances de P_1 telle que:

- elle est calculable en temps polynomiale.
- $P_1(x)$ est vrai si et seulement si $P_2(f(x))$ est vrai.

6.2.3 Théorème

- Si $P_1 \leq_p P_2$ et si $P_2 \in P$ alors $P_1 \in P$.
  (Si $P_1 \leq_p P_2$ et $P_1 \notin P$ alors $P_2 \notin P$.)
- Si $P_1 \leq_p P_2$ et si $P_2 \leq_p P_3$ alors $P_1 \leq_p P_3$.

Démonstration

Si $P_1 \in P$: il existe un algorithme polynomiale qui décide si $P(y)$ en temps en $O(p(n))$[1] où n est la taille du codage de y.
Pour résoudre $P_1(x)$:

- calculer $f(x)$ (où f est la réduction polynomiale qui permet $P_1 \leq_p P_2$).
- si $P_2(f(x))$ est vrai, alors répondre oui au problème P_1.
- sinon, répondre non.

Figure 48 : illustration de formules mathématiques repérées automatiquement dans un cours

Une fois que les documents ont été préparés et téléchargés, nous avons incité des étudiants étrangers (principalement des Chinois) à post-éditer.

En conséquence, 70 documents HTML, totalisant 16069 segments (phrases ou titres) d'une longueur moyenne de 8 mots par segment, ont été post-édités en chinois. Cela représente environ 514 pages standard. Le tableau ci-dessous montre l'état actuel de la plate-forme.

Thème du cours	**Type de contenu**	**Pages (html)**	**Traductions**
Introduction à la logique propositionnelle et du premier ordre	Livre complet	45	Chinois (complet), anglais (partiel), russe (partiel)
Programmation C	Documents de cours	14	Chinois (complet), Anglais (partiel)
Programmation orientée objet	Documents de cours	13	Chinois (complet), Anglais (partiel)
Complexité calculatoire	Notes de cours	13	Chinois (complet)
Interaction Homme-machine	Documents de cours	7	Chinois (complet)
Analyse Syntaxique	Documents de cours, polycopiés	5	Russe (partiel)
Modélisation de systèmes numériques	Sujet d'examen	2	Chinois (complet)
IA et planification automatique	Sujet d'examen	2	Chinois (complet)
Introduction à l'ergonomie	Compte rendu d'étudiant	1	Chinois (complet)

Figure 49 : état courant de la plate-forme Chamilo-MACAU

Tous ces documents sont en accès libre au public[123]. Nous sommes également ouverts à la création de nouvelles iMAG pour ceux qui sont intéressés par notre approche.

Un détail à noter est que la post-édition de contenu pédagogique en chinois s'est avéré très utile pour certains étudiants, car cela les a aidés à préparer certains examens exigeant une bonne et rapide compréhension de documents d'examen assez longs et complexes.

Un exemple frappant est celui d'un étudiant qui a obtenu d'excellentes notes au cours du semestre (20/20 en projet, 15/20 en TD/TP) dans des situations où il était capable de lire les instructions à son propre rythme et n'était pas obligé de produire de longues explications en français, mais a pourtant obtenu 2,5/20 à l'examen final. Après avoir post-édité avec nous, sa note est passée à 11/20 et il a validé ses matières. Nous concluons que ce processus l'a aidé à progresser tant dans le domaine que dans l'expression en français.

Avec cette expérience, en cours depuis 4 ans, nous avons donc :

- montré que notre approche est un moyen viable de produire du contenu multilingue à partir de sources monolingues.
- produit une quantité importante de documents en chinois, avec un accès gratuit.
- observé que la post-édition aide à comprendre le sujet.

GOOGLE TRADUCTION ne traduit pas bien les termes spécifiques à ce domaine. Nos étudiants ont résolu ce problème en créant et en "alimentant" un lexique terminologique multilingue en ligne spécifique dans un fichier GOOGLE SPREADSHEETS. Cela leur a permis de maintenir une cohérence entre traducteurs dans leurs traductions. L'importance de la disponibilité d'un lexique ou de la proximité linguistique entre la langue source et la langue cible pour l'efficacité de la post-édition a été démontrée par (Shah et al., 2015).

Un sous-ensemble des segments produits a été publié dans (Kalitvianski et al., 2016). (Wang, 2014) a montré qu'un système MOSES spécialisé entraîné sur de telles données produit des prétraductions qui sont plus rapides à post-éditer que celles produites par un système générique tel que GOOGLE TRANSLATE.

IV.1.2 De l'aide linguistique vers l'aide sémantique

Le problème de l'accès à l'information du point de vue du multilinguisme ayant été traité, nous nous proposons de nous attaquer à l'autre cause de difficulté à l'accès à l'information : la compréhension du sens intrinsèque des documents.

IV.1.2.1 Position du problème

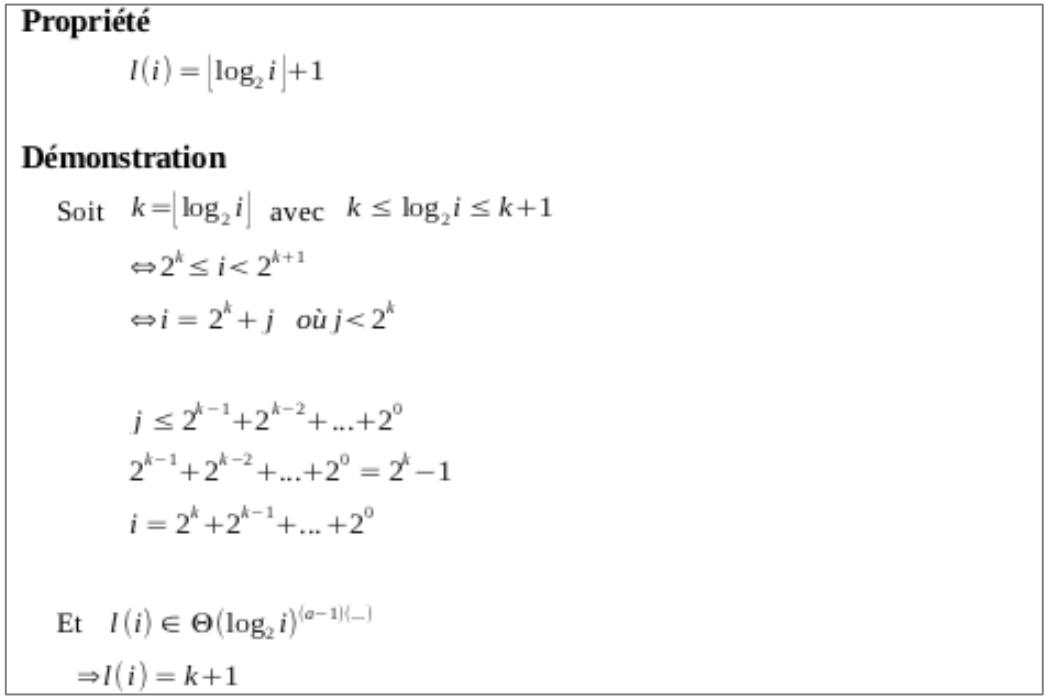
Il arrive souvent que le niveau de formalisme d'un support de cours, surtout en OFI, soit trop élevé pour des étudiants. Certains, indépendamment de leur niveau en mathématiques, éprouvent une anxiété devant la formalisation mathématique, les notations, etc.

Nous voudrions donc produire des documents dont le niveau de formalisme convienne à l'étudiant.

On dispose de notes de cours et éventuellement de comptes rendus de TD dont certains sont justement rédigés de façon moins formelle, et/ou moins laconique. Le problème est de trouver les meilleurs, de façon qu'ils soient enchaînables, cohérents, et au niveau le plus informel possible.

[123] http://tools.aximag.fr/macau/chamilo-macau/

Voici par exemple deux notes d'étudiants concernant l'estimation de la longueur minimale du codage binaire d'un entier (sans 0 inutile en tête). Dans la première, la propriété à démontrer n'est pas tout à fait exacte, et la démonstration a été prise en notes sans les commentaires oraux.



Propriété

$$l(i) = \lfloor \log_2 i \rfloor + 1$$

Démonstration

Soit $k = \lfloor \log_2 i \rfloor$ avec $k \leq \log_2 i \leq k+1$

$\Leftrightarrow 2^k \leq i < 2^{k+1}$

$\Leftrightarrow i = 2^k + j$ où $j < 2^k$

$j \leq 2^{k-1} + 2^{k-2} + \ldots + 2^0$

$2^{k-1} + 2^{k-2} + \ldots + 2^0 = 2^k - 1$

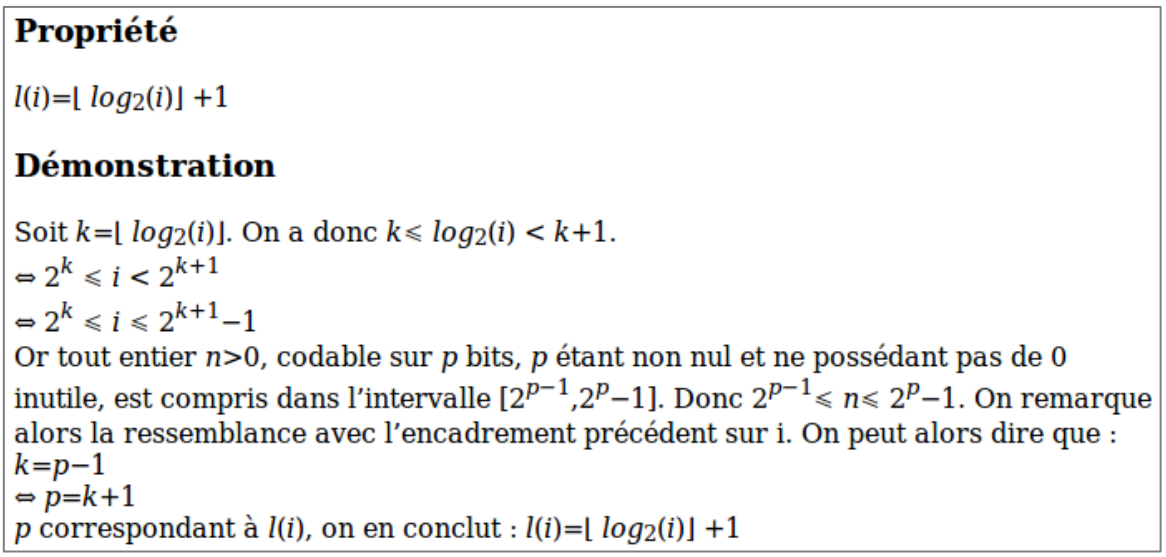
$i = 2^k + 2^{k-1} + \ldots + 2^0$

Et $l(i) \in \Theta(\log_2 i)^{(a-1)(\_)}$

$\Rightarrow l(i) = k+1$

Figure 50 : note 1 sur la longueur du codage binaire

Par contre, dans la seconde note, la propriété est toujours incomplète, mais la démonstration du cas usuel (n>0) est bien plus verbalisée et facile à suivre.



Propriété

$l(i)=\lfloor log_2(i) \rfloor + 1$

Démonstration

Soit $k=\lfloor log_2(i) \rfloor$. On a donc $k \leqslant log_2(i) < k+1$.
$\Leftrightarrow 2^k \leqslant i < 2^{k+1}$
$\Leftrightarrow 2^k \leqslant i \leqslant 2^{k+1}-1$
Or tout entier $n>0$, codable sur p bits, p étant non nul et ne possédant pas de 0 inutile, est compris dans l'intervalle $[2^{p-1}, 2^p-1]$. Donc $2^{p-1} \leqslant n \leqslant 2^p-1$. On remarque alors la ressemblance avec l'encadrement précédent sur i. On peut alors dire que :
$k=p-1$
$\Leftrightarrow p=k+1$
p correspondant à $l(i)$, on en conclut : $l(i)=\lfloor log_2(i) \rfloor + 1$

Figure 51 : note 2 sur la longueur du codage binaire

Remarquons que, dans les deux cas, les étudiants ont oublié de mentionner le cas (pourtant présenté en présentiel) où l'entier à coder (n) est nul. Il faudrait aussi annoter cela.

Nous remarquons une certaine structure dans ces documents : lorsqu'une *propriété* est énoncée, elle est suivie par une *démonstration* ; de même pour les *théorèmes*, qui peuvent de plus s'accompagner de *corollaires* et d'*exemples*. Nous observons cela aussi bien dans les deux figures qui précèdent, que dans la Figure 48. Cela nous pousse à penser qu'il serait intéressant de détecter cette structure dans les documents, d'identifier ces fragments, et de les proposer aux

apprenants, en tenant compte des propriétés pédagogiques de ces fragments, tels que leur niveau d'abstraction, leur enchaînement, le thème dont ils traitent, etc.

Pour arriver à ce qu'un programme puisse faire une proposition adaptée à chaque étudiant, il faut donc qu'il dispose d'informations sur lui (sans doute grâce à un « profil » réglable par lui et par l'enseignant) et sur les fragments identifiés dans les documents disponibles.

Il faudra donc réaliser de façon automatique ou semi-automatique une *fragmentation* des documents disponibles, et une *annotation* des fragments obtenus par des *métadonnées* adéquates. Nous donnons plus loin des détails sur la fragmentation et les métadonnées, dans le cas de MACAU-OFI[124].

IV.1.2.2 **Besoin d'une approche sémantique : conception de deux ontologies**

Il est facile de voir que ce domaine se formalise facilement : nous pouvons envisager une ontologie pour décrire la structure des documents pédagogiques, et une autre pour décrire le domaine pédagogique traité.

Nous suivons en cela le projet ACTIVEMATH, qui utilise trois ontologies[125], et le projet ANR OMNIA, qui en prévoyait aussi plusieurs (type et contenu, émotions exprimées, lieux et personnes) pour les images et textes de Belga-NEWS et Flickr.

Nous avons donc utilisé une ontologie de fragments pour décrire les documents et les fragments manipulés. En cela, nous rejoignons l'approche de TANGRAM[126] (Jovanovic et al., 2005), mais notre ontologie traite plus d'attributs (cours complet ou non, diaporama, production par un enseignant ou un étudiant, etc.).

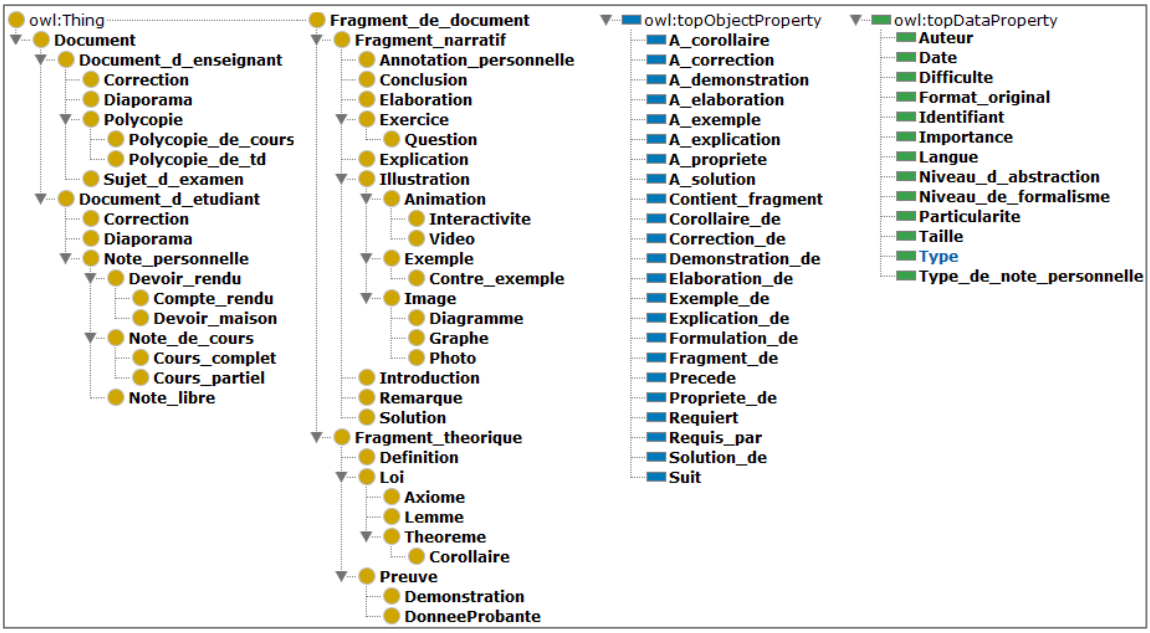


Figure 52 : diagramme de l'ontologie de documents pédagogiques

[124] OFI : outils formels de l'informatique.

[125] En particulier, une ontologie d'objets pédagogiques, qui sont classés en deux catégories : les objets fondamentaux (théorème, loi, fait, processus), et les objets auxiliaires qui s'y rapportent (preuve, démonstration, exemple), incluant aussi des éléments narratifs (introduction, remarque, conclusion) et interactifs (exercices, explorations, etc.) : https://www.researchgate.net/figure/The-ontology-of-instructional-objects_fig1_200166263

[126] TANGRAM propose une ontologie décrivant la structure formelle des documents (http://jelenajovanovic.net/ontologies/loco/alocom-content-structure/spec/) allant à des niveaux assez bas (liste à puces non numérotées, contenu d'une case de tableau, etc.) et une ontologie de rôles pédagogiques ou narratifs des objets (http://jelenajovanovic.net/ontologies/loco/alocom-content-type/spec/)

Dans cette ontologie, nous avons défini des concepts à l'échelle du document et à celle des sous-éléments, comme *Loi*, *Preuve*, *Élaboration*, etc.

La relation principale est l'inclusion de classes ; elle forme une hiérarchie. Il y a aussi des relations inverses et des relations transversales. Par exemple, un fragment théorique peut être élaboré ou expliqué par un fragment narratif, et il y a deux liens entre un théorème et une preuve.

L'ontologie comporte aussi des attributs contenant en particulier les métadonnées usuelles (*auteur*, *date*, *taille*, *type*, *langue*, *format initial*…).

La deuxième ontologie est celle d'un domaine pédagogique. Dans notre cas, nous avons modélisé les notions de complexité calculatoire enseignées en M1-info[127].

Nous voyons ci-dessous une version réduite de l'ontologie, datant de 2013. On trouvera en annexe la dernière version, beaucoup plus complète, datant de 2015.

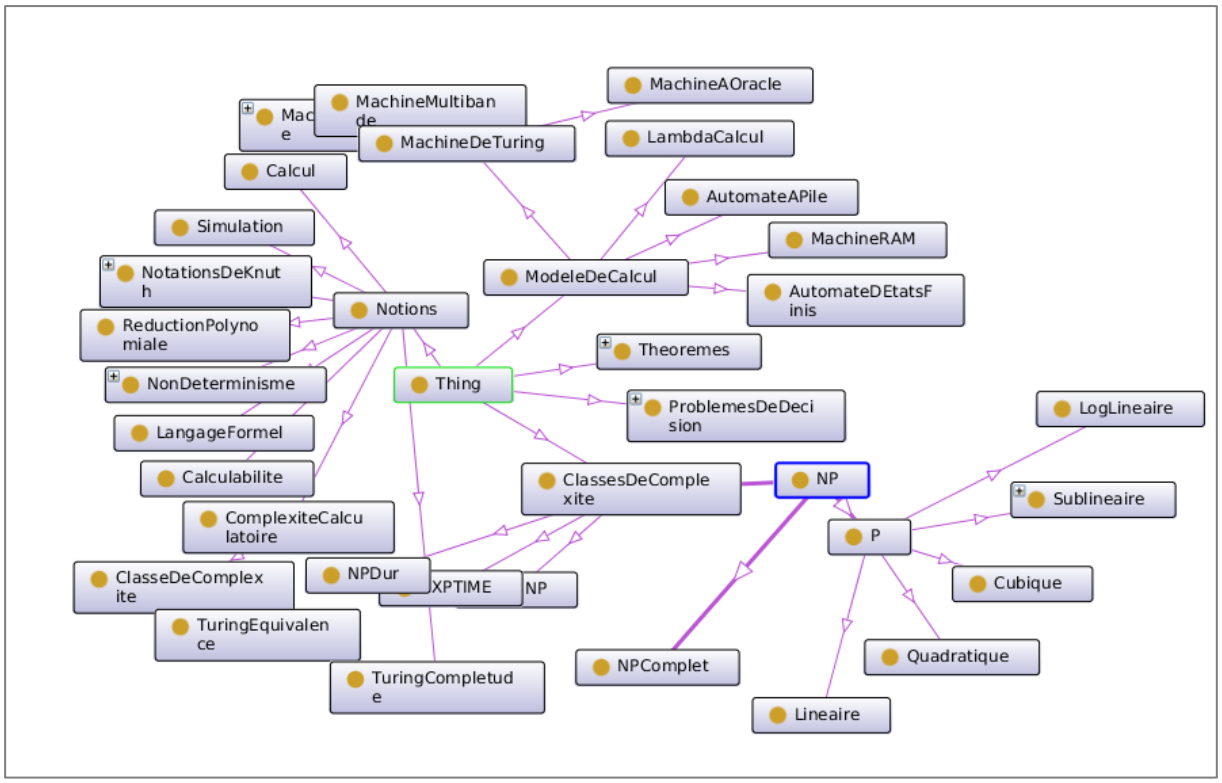


Figure 53 : ontologie du domaine de la complexité calculatoire dans MACAU

La construction de l'ontologie du domaine OFI a été partiellement guidée par les fragments obtenus en traitant les documents fournis par les enseignants, aussi bien que les notes de cours des étudiants.

Il y a deux approches à la construction d'une ontologie d'un domaine pour un usage en enseignement, l'une orientée vers la "taxonomie réelle" du domaine et l'autre orientée vers la structure du cours. La deuxième est induite par la structure des documents de cours et a donc l'avantage de pouvoir être extraite semi-automatiquement.

De façon générale, on sait que le processus de construction d'une ontologie de domaine peut être partiellement automatisé (Zouaq et al., 2006). Malgré tout, la construction d'une ontologie de domaine de bonne qualité fera toujours appel à des spécialistes de ce domaine, et à des cogniticiens connaissant bien les ontologies et le domaine plus récent des onto-terminologies.

[127] 1ère année de Master d'informatique de l'UFR IM²AG de l'UGA.

IV.1.2.3 **Nécessité d'extension de la modélisation des « corpus »**

Nous nous apercevons ici que la situation est inversée par rapport à celle du chapitre précédent : les documents que nous traitons ont une structure plus complexe, mais les liens entre eux sont plus simples que les graphes de citation des courriels.

Néanmoins, pour parvenir à un traitement de documents pédagogiques vraiment intéressant, les aspects « construction de parcours » et l'évolution de la forme et du contenu au fil des éditions (aspect temporel) devraient être considérés. Nous avons abordé ces aspects de façon très préliminaire.

IV.1.2.3.1 Aspect « construction de parcours »

Nous voudrions que les apprenants puissent se constituer des documents personnels d'apprentissage actif, qui tiennent compte de leurs connaissances déjà acquises, et de leurs préférences. C'est seulement une perspective à ce point, mais elle justifie la modélisation à une granularité faible.

Nous n'envisageons pas de nous substituer au professeur, mais plutôt de mettre à disposition des étudiants un outil de rédaction leur permettant de construire leurs cours personnalisés, et en 2 ou 3 langues. Pour cela, les segments devraient être multilingues, comme dans les brevets. Un étudiant chinois pourra ainsi non seulement traduire en chinois et voir « dans le texte » un exercice en français et en chinois, mais peut-être aussi écrire lui-même sa version de la solution, d'abord en chinois, puis en français (et pourquoi pas aussi en anglais).

Il faudrait aussi ajouter à MACAU un « tableau des post-éditions souhaitées ». Ainsi, les bons anglophones de la même promotion pourraient post-éditer les TA chinois-anglais, ou français-anglais.

IV.1.2.3.2 Aspect temporel

Les documents pédagogiques peuvent évoluer d'année en année, et il peut être intéressant de pouvoir suivre cette évolution. Si l'on devait le faire de manière automatique, cela nécessiterait sans doute d'utiliser des méthodes venant de la génétique textuelle. Voir par exemple la thèse de J. Bourdaillet[128] (2007).

Il faudrait en particulier arriver à distinguer les changements structurels, concernant par exemple l'ordre de la présentation, des changements textuels, concernant probablement le contenu, et des changements linguistiques, concernant la correction orthographie, grammaticale et terminologique.

IV.2 Traitement des documents textuels liés à MACAU-Chamilo

Comme dans le Chapitre 2 et le Chapitre 3, nous traitons de nouveau les aspects « forme » et « contenu » des documents. Nous présentons ici les adaptations apportées à SEGNORM pour fragmenter les documents pédagogiques, et les développements faits pour indexer et formuler des requêtes portant sur ces fragments.

IV.2.1 Traitement des documents pédagogiques au niveau de la forme

Les documents pédagogiques ont été fragmentés par un module de SEGNORM en fragments correspondants à des unités telles que Théorème, Preuve, Loi, etc.

[128] Bourdaillet, J. (2007). *Alignement textuel monolingue avec recherche de déplacements: algorithmique pour la critique génétique* (Thèse, Paris 6).

IV.2.1.1 Modélisation en sur-documents et sous-documents

Dans le cas des documents pédagogiques, on retrouve, comme pour les courriels, une structuration en sur-documents et sous-documents. Elle est assez différente, car (1) on rencontre beaucoup moins de citations de fragments arbitraires, (2) les liens sont plus riches, et (3) il y a une structuration hiérarchique forte.

(1) Par exemple, on cite souvent une définition, ou un théorème, ou un exemple (complet) d'automate, mais rarement une partie d'un théorème ou d'une démonstration.

(2) Les liens ne sont pas que des liens de citation. Il peut s'agir d'illustration, d'élaboration, de rappel, etc.

(3) Un cours est un sur-document composé de chapitres, et chaque chapitre est un sur-document composé de sections, elles-mêmes souvent divisées en sous-sections, puis en segments, puis en fragments textuels et formels. En effet, on trouve souvent des formules (expressions ou relations) dans une phrase.

IV.2.1.2 Implémentation : fragmentation récursive

SEGNORM a été doté d'un module de fragmentation spécialement conçu pour les types de documents que nous traitons.

Parmi les documents pédagogiques que nous avons collectés, 7 ont été identifiés comme étant de qualité et de variété suffisantes pour une première expérience. Une exigence majeure était la qualité du HTML : nous n'avons pris que des documents issus de conversions par WORD ou HEVEA, et pas de PDF.

Les 7 fichiers HTML correspondent à des cours complets ou partiels. Nous avons paramétré SEGNORM pour segmenter ces documents en chapitres (à l'aide de mots-clés et de la hiérarchie des balises HTML). Cela a produit 18 chapitres.

Ensuite, chaque chapitre a été fragmenté, là encore à l'aide de mots-clés et des balises, et un fichier-squelette a été construit, comme l'illustre la Figure 54. Sur cet exemple, cela a produit 158 fragments.

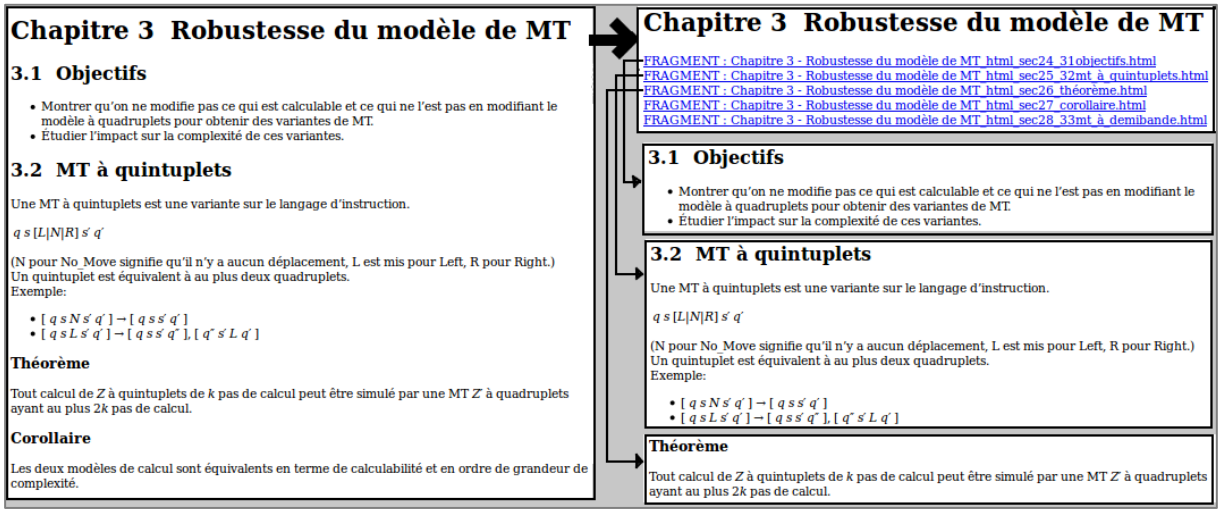


Figure 54 : illustration de la fragmentation automatique d'un document pédagogique

Voici le diagramme des traitements effectués par SegNorm pour les documents pédagogiques.

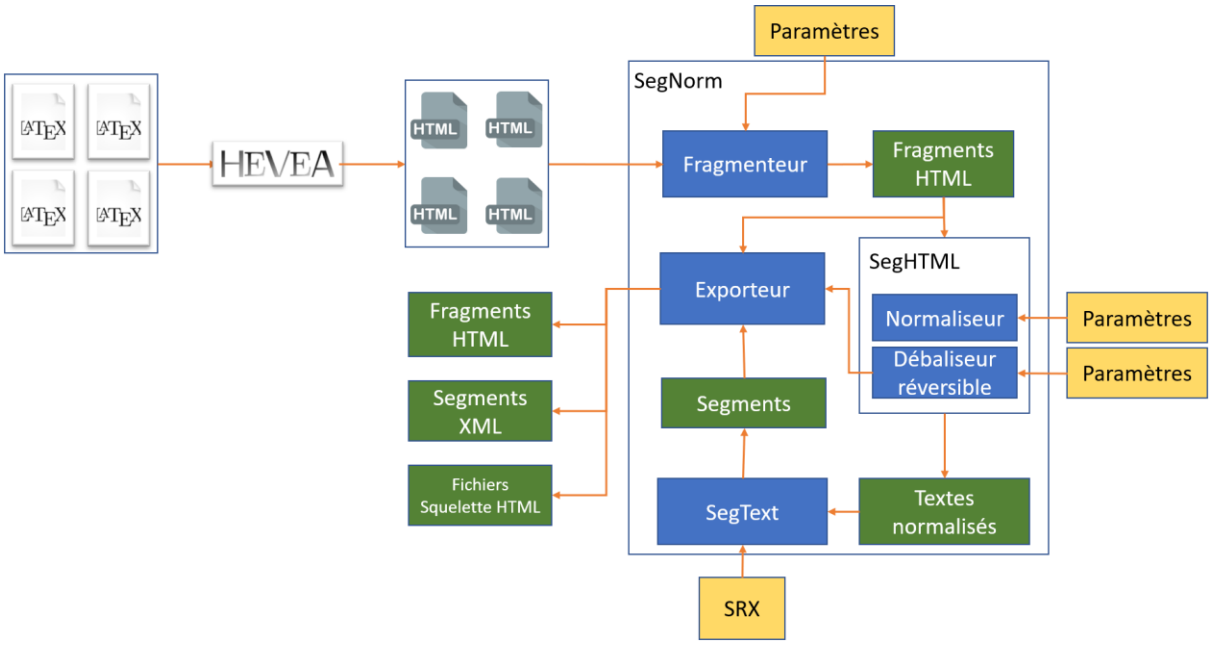


Figure 55 : chaîne de traitements SegNorm pour les documents pédagogiques

IV.2.1.3 Annotation par l'ontologie des formes

Après une révision manuelle, lors de laquelle nous avons scindé certains fragments et en avons fusionné d'autres, 147 fragments ont été importés dans l'ontologie des fragments (cf. Figure 52) et annotés à l'aide de PROTEGE.

Voici la répartition des fragments pour les catégories les plus dotées :

Catégorie	Nombre
Motivation	9
Définition	42
Théorème	21
Corollaire	10
Démonstration	19
Illustration	38
Remarque	7

Tableau 17 : répartition de fragments pédagogiques par catégories de l'ontologie des formes

Malheureusement, parmi les 18 documents retenus, les niveaux d'abstraction varient peu, ce qui ne nous a pas permis de faire d'expériences intéressantes sur cet aspect qui nous avait paru important *a priori*.

IV.2.2 Traitement des documents pédagogiques au niveau du contenu

IV.2.2.1 Spécification

Nous voudrions que les étudiants puissent formuler des requêtes, par exemple pour retrouver les passages concernant tel ou tel concept, ou bien les solutions utilisant le même type de méthode, comme la preuve par récurrence, ou la preuve par induction nœthérienne, ou la preuve d'équivalence de grammaires par passage aux automates, etc.

Nous n'avons pour l'instant abordé que le premier type de requêtes.

Pour cela, il nous a paru nécessaire d'annoter les fragments de certains types par l'ontologie du domaine, en nous limitant ici à la complexité calculatoire. Cette ontologie est plutôt une taxonomie, puisqu'elle n'a pas de relations autres que l'inclusion des classes.

Les 147 fragments ont été étiquetés par les concepts de l'ontologie.

IV.2.2.2 Premier algorithme d'exécution de requêtes et implémentation

Nous avons développé un script PHP qui permet de formuler des requêtes concernant les deux ontologies. Nous utilisons HERMIT[129] comme moteur de raisonnement ontologique.

La construction de l'ensemble de fragments du résultat se fait par intersection des résultats pour chaque ontologie. Ces résultats sont triés par précédence chronologique dans le cours, et par niveau d'élaboration, du plus succinct au plus élaboré.

Le résultat est restitué sous forme d'une page HTML éphémère, qui inclut tous les fragments dans des `iframe`. Chaque fragment affiche son auteur et permet d'aller au document source.

L'illustration ci-dessous montre le résultat d'une requête « Théorème » dans l'ontologie des fragments et « Équivalence polynomiale » dans l'ontologie du domaine.

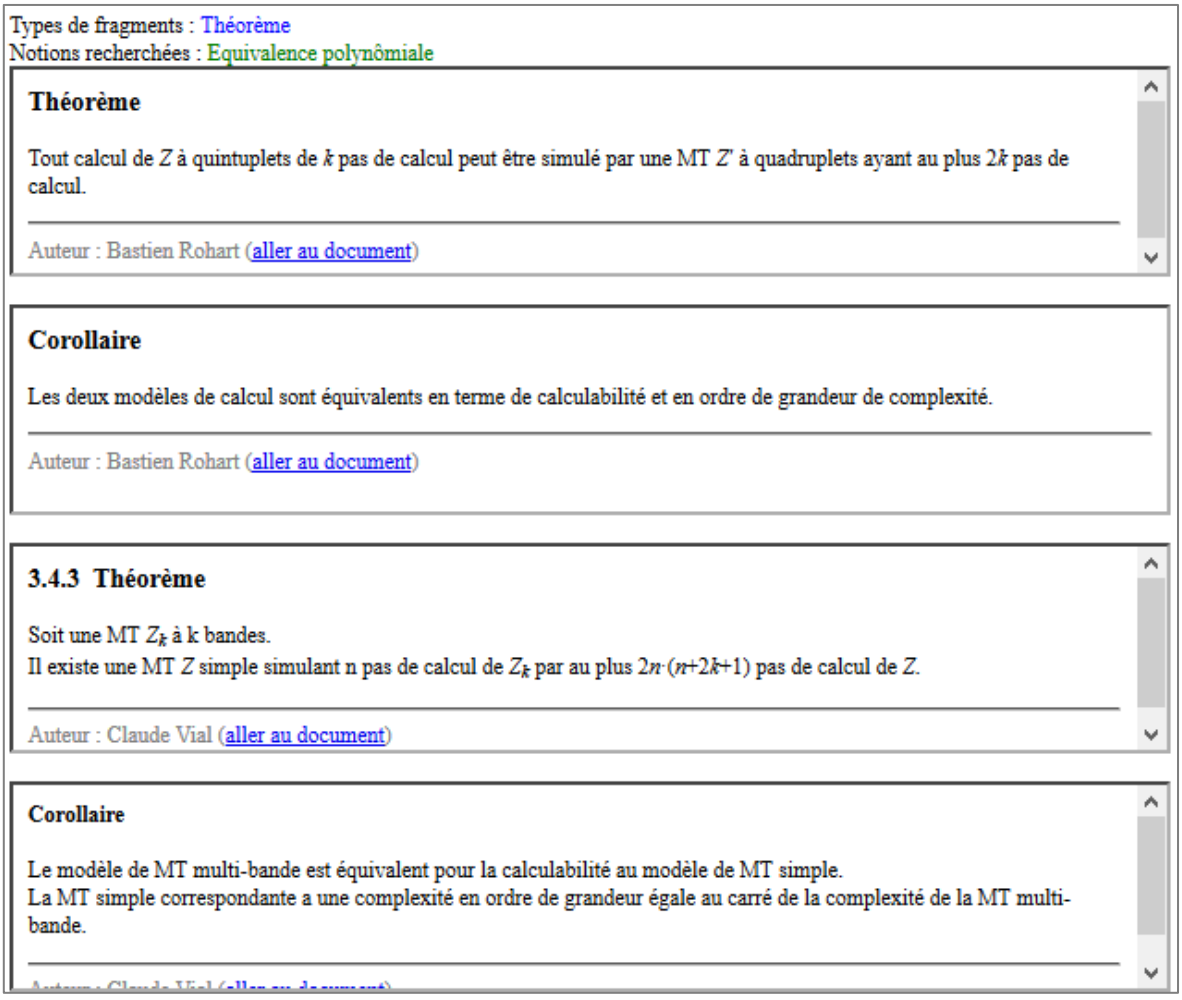


Figure 56 : résultat de la requête « théorème » + « équivalence polynomiale »

IV.2.3 Vers un étiquetage automatisé avec apprentissage

Pour réaliser l'expérience décrite ci-dessus, nous avons effectué l'étiquetage de façon manuelle. Lors de ce processus, nous nous sommes aperçu qu'il devrait être possible d'en automatiser certaines étapes, soit en construisant des patrons de façon experte, soit par apprentissage.

[129] http://www.hermit-reasoner.com/

IV.2.3.1 **Patrons d'étiquetage des types de fragment**

La Figure 56 et la Figure 48 rendent très manifeste le fait que le type d'un fragment peut souvent être « deviné » automatiquement en regardant le titre de la section ou de la sous-section qui le contient, donc en analysant le texte entre les balises `<hi>` et `</hi> (i≤9)`.

IV.2.3.2 **Détermination du séquencement des fragments et étiquetage des relations**

Une fois que cela est fait, déterminer le séquencement des fragments est tout aussi facile. Il suffit (la plupart du temps) de les prendre dans leur ordre d'apparition dans les documents. Par exemple, dans la Figure 51, il est clair que **Propriété** vient avant **Démonstration**.

En examinant automatiquement le séquencement des fragments, on pourra ainsi découvrir des motifs fréquents, tels que « propriété, démonstration », « définition, exemple », « théorème, preuve » ou « théorème, corollaire, preuve ». On pourra alors en déduire des relations plus sémantiques, comme `demonstration_de, exemple_de`, etc.

On voit ici que des termes différents, comme « démonstration » et « preuve », peuvent renvoyer à la même relation sémantique *dans le domaine considéré.* En effet, ces deux termes sont parfaitement synonymes dans le domaine des cours d'outils formels pour l'informatique, mais pas en général : une démonstration de force n'est pas nécessairement une preuve de force.

Le traitement automatique trouvera sans doute ses limites, et devra être complété par une intervention de l'enseignant ou des étudiants. Ainsi, si on a « théorème, lemme, démonstration, démonstration, corollaire, preuve », il est possible que la première démonstration soit celle du lemme, ou bien celle du théorème, le lemme étant démontré ensuite.

IV.2.3.3 **Méthode possible pour automatiser le typage sémantique**

Le problème ici est d'interpréter des fragments, le plus souvent des termes, dans l'ontologie du domaine. D'autre part, dans notre contexte, il faudrait pouvoir le faire à partir de plusieurs langues.

Pour cela, le plus simple semble être d'utiliser la technique du projet OMNIA, qui consiste à passer par une annotation interlingue. On peut assez facilement associer à chaque concept de l'ontologie un ensemble de mots-clés interlingues exprimés comme des UW (lexèmes interlingues) UNL.

Prenons par exemple le concept de preuve, disons, `` `proof` `` dans notre ontologie. Nous lui associerons les deux UW suivantes :

```
proof(icl>process, fld>math, syn>demonstration, agt>human, obj>proposition)
demonstration(icl>process, fld>math, syn>proof, agt>human, obj>theorem)
```

et d'autres éventuellement, construits à partir des termes anglais.

Pour chaque langue, nous aurons un dictionnaire reliant ses termes (comme « preuve ») à un ou plusieurs UW. Le traitement consistera à (1) annoter chaque terme reconnu par la ou les UW lui correspondant, puis (2) à calculer un score pour chacun, en fonction du contexte, par une des méthodes classiques (algorithmes de fourmis, recuit simulé, vecteurs sémantiques…), et (3) à désambiguïser en retenant l'UW de plus fort score sur la trajectoire elle-même de plus fort score[130].

[130] Il est possible d'avoir plusieurs trajectoires s'il y a des ambiguïtés, de segmentation ou d'attachement, comme dans « automate à pile non-déterministe » ou « machines à bandes multiples », ou « white paper wall » (dans un autre contexte !).

Une fois cela fait, nous pourrons relier chaque fragment aux concepts reliés à ses UW, et cela indépendamment pour chaque ontologie, et en retour annoter chaque fragment par ces concepts.

Bilan de ce chapitre et perspectives

Ce chapitre a été consacré à l'accès multilingue et sémantique à des documents pédagogiques, dont l'organisation et la structure sont différentes de celles des courriels. Notre expérience d'enseignement a montré que les principaux freins à l'accès à l'information sont linguistiques et sémantiques.

Pour résoudre le premier problème, nous avons proposé et implémenté un accès multilingue assisté par la traduction automatique post-éditable.

Pour le second, nous avons proposé et implémenté une approche de type web sémantique, en construisant et utilisant deux ontologies de domaine.

À cause de la nature des documents pédagogiques, nous avons été amené à étendre la modélisation et le schéma de traitement mis en œuvre pour les courriels, en introduisant une phase générique de segmentation/normalisation et une autre, spécifique à chaque type de traitement.

Il s'agit ici non seulement de segmentation/normalisation pour la TA, mais aussi de fragmentation en unités plus grandes et plus autonomes que les segments-phrases usuels ; elles peuvent être recombinées pour former des documents de cours « sur mesure » pour des étudiants. Nous avons ainsi annoté des fragments à l'aide de deux ontologies, et implémenté un premier algorithme pour interroger les ontologies.

Les expériences que nous avons menées nous conduisent à envisager de concevoir un véritable *langage de segmentation*, qui servirait à exprimer ce qui constitue un segment et comment procéder à la segmentation. Un tel langage, qu'on imagine permettre des segmentations multiples et récursives, remplacerait avantageusement la programmation directe qui a été faite pour le fragmenteur des documents de MACAU.

Conclusions et perspectives de la thèse

Conclusions

Notre recherche s'inscrit dans la problématique de l'extraction de sens à partir de textes et flux textuels, produits dans notre cas lors de processus collaboratifs. Plus précisément, nous nous sommes intéressé aux courriels de travail et aux documents textuels objets de collaboration, avec une première application aux documents éducatifs. Notre motivation principale, en quelque sorte notre fil conducteur, a été et est toujours d'aider les utilisateurs à accéder plus rapidement aux informations utiles. Par conséquent, notre but principal est de repérer ces informations dans les textes. Ainsi, nous nous intéressons aux tâches dans les courriels, et aux fragments de documents éducatifs qui concernent les thèmes de leurs intérêts. Nous avons constitué deux corpus, un de courriels et un de documents éducatifs, principalement en français. Cela était indispensable, car il n'y avait pratiquement pas de travaux antérieurs sur ce type de données en français.

Notre première contribution théorique est une modélisation générique de la structure de ces données. Nous l'avons utilisée pour spécifier le traitement formel des documents, prérequis au traitement sémantique. Nous avons démontré la difficulté du problème de segmentation, normalisation et structuration de documents en différents formats source, et présenté l'outil SEGNORM, première contribution logicielle de cette thèse. SEGNORM segmente et normalise les documents (en texte brut ou balisé), récursivement et en unités de taille paramétrable. Dans le cas des courriels, il segmente les messages contenant des messages cités en messages individuels, en conservant l'information du chaînage entre les fragments entremêlés. Il analyse également les métadonnées des messages pour reconstruire les fils de discussions, et retrouve dans les citations les messages dont on ne possède pas le fichier source.

Nous avons ensuite abordé le traitement sémantique de ces documents. Nous avons proposé une modélisation (ontologique) de la notion de tâche, puis décrit l'annotation d'un corpus de plusieurs centaines de messages issus du contexte professionnel de VISEO et du GETALP. Nous avons alors présenté la deuxième contribution logicielle de cette thèse : l'outil de repérage de tâches et d'extraction de leurs attributs (contraintes temporelles, assignataires, etc.). Cet outil, basé sur une combinaison d'une approche experte et d'apprentissage automatique, est évalué selon des critères classiques de précision, rappel et F-mesure, ainsi que selon la qualité d'usage.

Enfin, nous avons présenté nos travaux sur la plate-forme MACAU-CHAMILO, troisième contribution logicielle, qui aide à l'apprentissage par (1) accès multilingue à du contenu initialement monolingue, et (2) structuration de documents pédagogiques selon deux ontologies (forme et contenu). Il s'agit donc de nouveau de structuration selon les deux axes, forme et sens.

(1) L'aspect relatif à l'accès multilingue a été motivé par le constat que nos universités accueillent un grand nombre d'étudiants étrangers, qui ont souvent du mal à comprendre nos cours à cause de la barrière linguistique. Nous avons proposé une approche pour multilingualiser du contenu pédagogique avec l'aide d'étudiants étrangers, par *post-édition* en ligne de pré-traductions automatiques, puis, si besoin, amélioration incrémentale de ces post-éditions. Nos expériences ont montré que des versions multilingues de documents peuvent être produites rapidement et sans coût, et que les étudiants « post-éditeurs » avaient été aidés par cet outil, progressant à la fois en français et dans le domaine des documents post-édités.

Ce travail a abouti à un corpus de plus de 500 pages standard (250 mots/page) de contenu pédagogique post-édité vers le chinois. Ce corpus constitue une troisième contribution au niveau des ressources mises à disposition en utilisation libre[131].

(2) L'ontologie des formes permet d'annoter les fragments des documents par des concepts comme *théorème, preuve, exemple*, par des niveaux de difficulté et d'abstraction, et par des relations comme `élaboration_de`, `illustration_de`. L'ontologie de domaine modélise les objets formels de l'informatique, et plus précisément les notions de complexité calculatoire. Cela permet de suggérer aux utilisateurs des fragments utiles pour la compréhension de notions d'informatique perçues comme abstraites ou difficiles. Ces deux ontologies sont elles-aussi mises à disposition des chercheurs.

Perspectives

En ce qui concerne les courriels, de nombreuses perspectives se profilent. Dans cette thèse, nous avons principalement abordé le repérage de segments exprimant des tâches, ainsi que de leurs attributs. La section III.3.4 propose des directions vers le typage des tâches, notamment leur séquencement temporel. Il est intéressant d'étudier de façon détaillée la manière dont les informations de ce genre sont exprimées (ou non) dans les textes, et comment les extraire automatiquement.

Une technologie actuellement très prometteuse est l'apprentissage profond *(deep learning).* De nombreuses architectures sont utilisées, dont les réseaux à convolution, à mémoire, à repliement temporel *(time warping),* et autres. Ces réseaux opèrent sur des représentations des mots en vecteurs à grand nombre de dimensions (*word2vec,* (Mikolov et al., 2013)), appris sur de grandes quantités de texte. Aujourd'hui, il n'existe pas de vecteurs *word2vec* appris sur des courriels en français issus de collaborations professionnelles[132]. Une perspective serait d'en construire et d'expérimenter ces approches.

En ce qui concerne MACAU, plusieurs perspectives sont envisagées. La plus importante serait une vraie évaluation de l'utilité pédagogique de l'accès multilingue et de l'aide apportée par l'usage des ontologies. Plus particulièrement, nous voudrions évaluer l'apport à la compréhension, non seulement par le simple accès multilingue, dont nous savons déjà qu'il facilite la lecture et par là la compréhension du domaine dans la langue d'enseignement, mais aussi par le processus de post-édition, qui est de nature plus active. Nos post-éditeurs chinois ont rapporté avoir mieux compris les matières post-éditées, et nous voudrions étendre cette expérience à d'autres matières et d'autres publics. Cela entraînerait donc la construction d'ontologies de domaines autres que la complexité calculatoire, sur laquelle nous avons fait l'essentiel de notre expérience en français-chinois. L'accès multilingue étant possible grâce aux IMAG, nous voudrions aussi apporter des améliorations techniques au frontal et au dorsal. Enfin, nous voudrions rendre ce service plus répandu, au travers de collaborations internationales, comme celle que nous avons engagée en 2014 et 2015 avec les universités sibériennes, notamment Tomsk et Novosibirsk.

L'outil essentiel qui nous a permis de réaliser ce travail a été SEGNORM, outil de segmentation et de normalisation produit durant cette thèse par extension des fonctions de segmentation (et « fragmentation » de l'ancien SEGDOC, et par ajout du module de normalisation NORMDOC. Nous envisageons de concevoir un vrai langage de segmentation qui permettrait aux non-

[131] Il est actuellement disponible sur http://tools.aximag.fr/macau/chamilo-macau/ et sera mis très prochainement sur PerSCiDO à http://persyval-platform.univ-grenoble-alpes.fr.

[132] Les très gros corpus de textes en français sont Wikipédia-fr, qui n'est pas adapté car ce ne sont pas des dialogues, et Ubuntu-fr, qui contient bien des échanges électroniques, mais qui relève d'un domaine très particulier, donc pas tout à fait adéquat pour le type de texte qui nous intéresse.

programmeurs d'exprimer des graphes de segmentations possibles. SEGNORM pourrait également bénéficier grandement d'une interface graphique. Enfin, nous voudrions l'étendre à d'autres types de segmentation, plus sémantiques, comme la segmentation discursive de dialogues ou la segmentation thématique.

En ce qui me concerne, je voudrais continuer à travailler sur le traitement de textes spontanés non structurés. Le traitement adéquat du multilinguisme est plus que jamais un besoin. Comme nous travaillons depuis longtemps avec UNL, nous voudrions tout naturellement utiliser UNL pour un élaborer des règles de repérage de tâches dans un contexte véritablement multilingue, ou plutôt interlingue.

Bibliographie

1. Allen, James, and Mark Core. 1997. Draft of DAMSL: Dialog Act Markup in Several Layers. *URL http://www. cs. rochester. edu/research/cisd/resources/damsl/RevisedManual*.
2. Baldwin, Breck, and Bob Carpenter. 2003. LingPipe. Available from World Wide Web: http://alias-i. com/lingpipe.
3. Bennett, Paul N, and Jaime Carbonell. 2005. Detecting action-items in e-mail. Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval.
4. ———. 2005. Feature representation for effective action-item detection. *ACM SIGIR Special Interest Group on Information Retrieval*.
5. Bennett, Paul N, and Jaime G Carbonell. 2007. Combining Probability-Based Rankers for Action-Item Detection. Proc. of HLT-NAACL.
6. Berger, Adam L, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics* 22 (1):39-71.
7. Besacier, Laurent. 2014. Traduction automatisée d'une oeuvre littéraire: une étude pilote. Actes de Traitement Automatique du Langage Naturel (TALN).
8. Bird, Steven, Ewan Klein, and Edward Loper. 2009. Natural language processing with Python: analyzing text with the natural language toolkit: O'Reilly Media, Inc.
9. Blanchon, Hervé. 2004. Comment définir, mesurer et améliorer la qualité, l'utilisabilité et l'utilité des systèmes de TAO de l'écrit et de l'oral. Une bataille contre le bruit, l'ambiguïté, et le manque de contexte. *Grenoble, UJF. HDR,* 380 p.
10. Boitet, Christian, HUYNH Cong Phap, NGUYEN Hong Thai, and Valérie Bellynck. 2010. The iMAG concept: multilingual access gateway to an elected Web sites with incremental quality increase through collaborative post-edition of MT pretranslations. Proc. of *TALN-2010* 8 p.
11. Bunt, Harry. 2009. The DIT++ taxonomy for functional dialogue markup. Proc. of AAMAS 2009 Workshop, Towards a Standard Markup Language for Embodied Dialogue Acts.
12. Carvalho, Vitor R, and William W Cohen. 2004. Learning to extract signature and reply lines from email. Proceedings of the Conference on Email and Anti-Spam.
13. ———. 2005. On the collective classification of email speech acts. Proceedings of the 28th annual international ACM SIGIR conference on Research and Development in Information Retrieval.
14. Chang, Chih-Chung, and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2 (3):27.
15. Chanier, Thierry, Marie-Noelle Lamy, Christophe Reffay, Marie-Laure Betbeder, and Maud Ciekanski. 2009. LETEC (Learning and Teaching Corpus) Simuligne. *Chanier T. (ed.) Banque de corpus CoMeRe. Ortolang.fr : Nancy. [cmr-simuligne-tei-v1 ; https://hdl.handle.net/11403/comere/cmr-simuligne-tei-v1]*
16. Chen, Yahui. 2015. Convolutional neural network for sentence classification, University of Waterloo.
17. Chen, Yun-Nung, Dilek Hakkani-Tür, and Xiaodong He. 2015. Detecting actionable items in meetings by convolutional deep structured semantic models. Proc. of Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on.
18. Cohen, William W, Vitor R. Carvalho, and Tom M Mitchell. 2004. Learning to Classify Email into" Speech Acts". Proc. of EMNLP.

19. Corston-Oliver, Simon, Eric Ringger, Michael Gamon, and Richard Campbell. 2004. Task-focused summarization of email. Proc. of ACL-04 Workshop: Text Summarization Branches Out.

20. Cortes, Corinna, and Vladimir Vapnik. 1995. Support vector machine. *Machine learning* 20 (3):273-297.

21. Couillault, Alain, Axelle Vinckx, Hugues De Mazancourt, Fanny Grandry, and Gaëlle Recourcé. 2014. Rapport technique Projet Gramlab: livrable SP5. 1 Use Case Eptica/Lingway: identification d'amorces de reprises.

22. De Mazancourt, Hugues, Alain Couillault, and Gaëlle Recourcé. 2014. L'anonymisation, pierre d'achoppement pour le traitement automatique des courriels. Proc. of Journée d'Etude ATALA Ethique et TAL.

23. Dehghani, Mostafa, Azadeh Shakery, Masoud Asadpour, and Arash Koushkestani. 2013. A learning approach for email conversation thread reconstruction. *Journal of Information Science* 39 (6):846-863.

24. Fafiotte, Georges, and Christian Boitet. 2003. ERIMM: a platform for supporting and collecting multimodal spontaneous bilingual dialogues. Proc. of Natural Language Processing and Knowledge Engineering, 2003.

25. Falaise, Achille, David Rouquet, Didier Schwab, Hervé Blanchon, and Christian Boitet. 2010. Ontology-driven content extraction using interlingual annotation of texts in the OMNIA project. Proceedings of the 4th International Workshop on Cross-Lingual Information Access at COLING.

26. Farzindar, Atefeh, and Diana Inkpen. 2015. Natural language processing for social media. *Synthesis Lectures on Human Language Technologies* 8 (2):1-166.

27. Francis, W. Nelson, and Henry Kucera. 1979. Brown corpus manual. *Brown University* Volume 2.

28. Gillick, Dan. 2009. Sentence boundary detection and the problem with the U.S. Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers.

29. Green, Spence, Jeffrey Heer, and Christopher D. Manning. 2013. The efficacy of human post-editing for language translation. Proceedings of the SIGCHI conference on human factors in computing systems.

30. Guthrie, David, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. 2006. A closer look at skip-gram modelling. Paper read at Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006).

31. Handschuh, Siegfried, Knud Möller, and Tudor Groza. 2007. The NEPOMUK project-on the way to the social semantic desktop. *Proc. of I-Semantics' 07*.

32. Hernandez, Nicolas, and Soufian Salim. 2014. Exploiting the human computational effort dedicated to message reply formatting for training discursive email segmenters. Proceedings of LAW VIII-The 8th Linguistic Annotation Workshop.

33. Hernandez, Nicolas, Soufian Salim, and Elizaveta Loginova Clouet. 2016. Ubuntu-fr: A Large and Open Corpus for Multi-modal Analysis of Online Written Conversations. Proc. of LREC-2016.

34. Huynh, Cong-Phap. 2010. Des suites de test pour la TA à un système d'exploitation de corpus alignés de documents et métadocuments multilingues, multiannotés et multimédia, Thèse, Institut National Polytechnique de Grenoble (INPG).

35. Huynh, Cong-Phap, Christian Boitet, and Hervé Blanchon. 2008. SECTra_w. 1: an Online Collaborative System for Evaluating, Post-editing and Presenting MT Translation Corpora. Proc. of LREC-2008.

36. Jackson, Thomas, Ray Dawson, and Darren Wilson. 2001. The cost of email interruption. *Journal of Systems and Information Technology* 5 (1):81-92.

37. Jeong, Minwoo, Chin-Yew Lin, and Gary Geunbae Lee. 2009. Semi-supervised speech act recognition in emails and forums. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3.

38. Jovanovic, Jelena, Gasevic Dragan, and Vladan Devedzic. 2005. TANGRAM: An ontology-based learning environment for intelligent information systems. Proc. of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education.

39. Kalia, Anup K., Hamid R. Motahari-Nezhad, Claudio Bartolini, and Munindar P. Singh. 2013. Monitoring commitments in people-driven service engagements. Paper read at Services Computing (SCC), 2013 IEEE International Conference on.

40. Kalitvianski, Ruslan, Valérie Bellynck, and Christian Boitet. 2015. Multilingual Access to Educational Material Through Contributive Post-editing of MT Pre-translations by Foreign Students. Proc. of International Conference on Web-Based Learning, Guangzhou, 9 p.

41. ———. 2017. Un outil de segmentation de courriels imbriqués en courriels individuels et en phrases. Actes de Atelier Fouille des Données Complexes @ EGC-2017 (Extraction et Gestion des Connaissances).

42. Kalitvianski, Ruslan, Christian Boitet, and Valérie Bellynck. 2012. Collaborative Computer-Assisted Translation Applied to Pedagogical Documents and Literary Works. Paper read at COLING (Demos), Mumbai.

43. Kalitvianski, Ruslan, Lingxiao Wang, Valérie Bellynck, and Christian Boitet. 2016. An Aligned French-Chinese corpus of 10K segments from university educational material. Proc. of *NLPTEA 2016*, Osaka.

44. Klimt, Bryan, and Yiming Yang. 2004. The Enron corpus: A new dataset for email classification research. *Machine learning: ECML 2004*:217-226.

45. Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, and Richard Zens. 2007. Moses: Open source toolkit for statistical machine translation. Proceedings of the 45th annual meeting of the ACL, interactive poster and demonstration session.

46. Lampert, Andrew, Daniel Breese, Cécile Paris, and Robert Dale. 2009. A Tool for Capturing Context-Sensitive Judgements in Email Data. In *CSIRO ICT Centre Technical Report (EP092057)*.

47. Lampert, Andrew, Robert Dale, and Cécile Paris. 2008. The nature of requests and commitments in email messages. Proceedings of the AAAI Workshop on Enhanced Messaging.

48. ———. 2008. Requests and commitments in email are more complex than you think: Eight reasons to be cautious. Proc. of Australasian Language Technology Association Workshop 2008.

49. ———. 2009. Segmenting email message text into zones. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2.

50. Lampert, Andrew, Robert Dale, and Cecile Paris. 2010. Detecting emails containing requests for action. Proc. of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics.

51. Lampert, Andrew, Cécile Paris, and Robert Dale. 2007. Can requests-for-action and commitments-to-act be reliably identified in email messages. Proceedings of the 12th Australasian Document Computing Symposium.

52. Lampert, Andrew Thomas. 2014. Making email actionable: the identification and use of obligation acts in workplace email, PhD Thesis, Department of Computing, Macquarie University, Sydney, Australia.

53. Lascarides, Alex, and Nicholas Asher. 2008. Segmented discourse representation theory: Dynamic semantics with discourse structure. In *Computing meaning*: Springer.

54. Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Standford CoreNLP Natural Language Processing Toolkit. Proc. of ACL (System Demonstrations).

55. Maranget, Luc. 2003. On using HEVEA, a fast LATEX to html translator. *Eutypon, proceedings of the Greek TeX friends group*.

56. Marcus, Mitchell P., Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics* 19 (2):313-330.

57. Mattmann, Chris, and Jukka Zitting. 2011. *Tika in action*: Manning Publications Co.

58. McCallum, Andrew Kachites. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu. 2002.

59. Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. Proc. of Advances in neural information processing systems.

60. Miłkowski, Marcin, and Jarosław Lipski. 2009. Using SRX standard for sentence segmentation in LanguageTool. *Human Language Technologies as a Challenge for Computer Science and Linguistics*:556-560.

61. Musen, Mark, M. Crubézy, R. Fergerson, N.F. Noy, S. Tu, and J. Vendetti. 2009. The Protégé ontology editor and knowledge acquisition system. *Stanford Center for Biomedical Informatics Research*.

62. Partalas, Ioannis, Cédric Lopez, Nadia Derbas, and Ruslan Kalitvianski. 2016. Learning to search for recognizing named entities in Twitter. Proc. of *WNUT 2016*:171.

63. Planas, Emmanuel. 1998. TELA: Structure et algorithmes pour la traduction fondée sur la mémoire, Thèse, Université Joseph-Fourier-Grenoble I.

64. Planas, Emmanuel, and Osamu Furuse. 1999. Formalizing translation memories. Proc. of Machine Translation Summit VII, Singapore.

65. Purver, Matthew, Patrick Ehlen, and John Niekrasz. 2006. Detecting action items in multi-party meetings: Annotation and initial experiments. Proc. of MLMI, Bethesda, USA.

66. ———. 2006. Shallow discourse structure for action item detection. Proceedings of the HLT-NAACL 2006 Workshop on Analyzing Conversations in Text and Speech.

67. Read, Jonathon, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. 2012. Sentence boundary detection: A long solved problem? *COLING (Posters)* 12:985-994.

68. Riss, Uwe V, Marlen Jurisch, and Viktor Kaufman. 2009. E-mail in semantic task management. Proc. of IEEE Conference on Commerce and Enterprise Computing, 2009. CEC'09.

69. Rouquet, David. 2012. Multilinguisation d'ontologies dans le cadre de la recherche d'information translingue dans des collections d'images accompagnées de textes spontanés, Thèse, Université de Grenoble.

70. Sagot, Benoît. 2010. The Lefff, a freely available and large-coverage morphological and syntactic lexicon for French. Proc. of 7th international conference on Language Resources and Evaluation (LREC 2010).

71. Salim, Soufian, Nicolas Hernandez, and Emmanuel Morin. 2016. Comparaison d'approches de classification automatique des actes de dialogue dans un corpus de conversations écrites en ligne sur différentes modalités. Paper read at TALN 2016, at Paris, France.

72. Scerri, Simon, Gerhard Gossen, Brian Davis, and Siegfried Handschuh. 2010. Classifying Action Items for Semantic Email. Proc. of LREC-2010, Valletta, Malta.

73. Searle, John R. 1969. *Speech acts: An essay in the philosophy of language*. Vol. 626: Cambridge university press.

74. Sénellart, Jean, Christian Boitet, and Laurent Romary. 2003. XML Translation Workflow. Proc. of Machine Translation Summit IX.

75. Shah, Ritesh, Christian Boitet, Pushpak Bhattacharyya, Mithun Padmakumar, Leonardo Zilio, Ruslan Kalitvianski, Mohammad Nasiruddin, Mutsuko Tomokiyo, and Sandra Castellanos Páez. 2015. Post-editing a chapter of a specialized textbook into 7 languages: importance of terminological proximity with English for productivity. Proc. of 12th International Conference on Natural Language Processing.

76. Stenetorp, Pontus, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a web-based tool for NLP-assisted text annotation. Paper read at Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, Trivandrum, India.

77. Strötgen, Jannik, and Michael Gertz. 2010. Heideltime: High quality rule-based extraction and normalization of temporal expressions. Proc. of Proceedings of the 5th International Workshop on Semantic Evaluation.

78. Styler, Will. 2011. The EnronSent Corpus. Boulder: University of Colorado at Boulder Institute of Cognitive Science.

79. Tang, Jie, Hang Li, Yunbo Cao, and Zhaohui Tang. 2005. Email data cleaning. Proc. of Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.

80. Tarasowa, Darya, Ali Khalili, and Sören Auer. 2015. Crowdlearn: Crowd-sourcing the creation of highly-structured e-learning content. *International Journal of Engineering Pedagogy (iJEP)* 5 (4):47-54.

81. Tavafi, Maryam, Yashar Mehdad, Shafiq R Joty, Giuseppe Carenini, and Raymond T Ng. 2013. Dialogue Act Recognition in Synchronous and Asynchronous Conversations. Proc. of SIGDIAL Conference.

82. Tomokiyo, Mutsuko. 2000. Analyse discursive de dialogues oraux en français, japonais et anglais: élaboration et validation par anlayse comparative de corpus réels, Thèse, Paris 7.

83. Tur, Gokhan, Andreas Stolcke, Lynn Voss, John Dowding, Benoît Favre, Raquel Fernández, Matthew Frampton, Michael Frandsen, Clint Frederickson, and Martin Graciarena. 2008. The CALO meeting speech recognition and understanding system. Proc. of Spoken Language Technology Workshop, 2008. SLT 2008. IEEE.

84. Uchida, Hiroshi, Meiying Zhu, and Tarcisio G. Della Senta. 2006. *UNL: Universal Networking Language*: UNDL Foundation, International environment house.

85. Ulrich, Jan, Gabriel Murray, and Giuseppe Carenini. 2008. A publicly available annotated corpus for supervised email summarization. Proc. of AAAI Email-2008 Workshop, Chicago, USA.

86. Urieli, Assaf. 2013. Robust French syntax analysis: reconciling statistical methods and linguistic knowledge in the Talismane toolkit, PhD thesis, Université Toulouse le Mirail-Toulouse II.

87. Wang, Lingxiao. 2015. Outils et environnements pour l'amélioration incrémentale, la post-édition contributive et l'évaluation continue de systèmes de TA. Application à la TA français-chinois, Thèse, Université Grenoble Alpes.

88. Wang, Lingxiao, and Christian Boitet. 2013. Online production of HQ parallel corpora and permanent task-based evaluation of multiple MT systems: both can be obtained through iMAGs with no added cost. Proceedings of MT Summit XIV, 2nd Workshop on Post-editing Technologies and Practice.

89. Winograd, Terry. 1986. A language/action perspective on the design of cooperative work. Proceedings of the 1986 ACM conference on Computer-supported cooperative work.

90. Yeh, Jen-Yuan, and Aaron Harnly. 2006. Email Thread Reassembly Using Similarity Matching. Proc. of CEAS, Mountain View, USA.

91. Zhou, Yingjie, Mark Goldberg, Malik Magdon-Ismail, and Al Wallace. 2007. Strategies for cleaning organizational emails with an application to Enron email dataset. Proc. of 5th Conf. of North American Association for Computational Social and Organizational Science.

92. Zouaq, Amal, Roger Nkambou, and Claude Frasson. 2006. The knowledge puzzle: An integrated approach of intelligent tutoring systems and knowledge management. Paper read at Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on.

Table des définitions

Annexes

Cette section présente huit annexes :

1. Extrait d'un fichier de paramétrage de SEGNORM pour la segmentation
2. Exemples de sorties de segmentation et de normalisation
3. Variété linguistique des en-têtes trouvées dans les messages
4. Exemple de nos annotations produites à l'aide de l'outil BRAT
5. Exemples de marqueurs de listes à puces dans les courriels
6. Exemples d'expressions de contraintes temporelles dans les corpus de courriels
7. Lexiques de termes caractéristiques de tâches, et exemples de règles linguistiques
8. Ontologies de MACAU

Annexe 1 Extrait d'un fichier de paramétrage de SEGNORM pour la segmentation

Le tableau ci-dessous est un des quatre fichiers de paramétrage de SEGNORM.

```
################################################
############### Config des regex ###############
################################################

# Caractères évalués comme des blancs
htmlSpaceChars=\\s|\\u00A0| 

# Tags dont il faut ignorer le contenu
htmlIgnoreTags=style|script|code

# Tags forts
htmlStrongTags=html|head|body|title|p|div|table|tr|dd|dt|dl|td|th|ul|li|h[0-
9]|label|option|object|pre


############################################################
############### Config de la segmentation fine ###############
############################################################

# Considerer <br> comme tag fort?
brStrong=true

# Considerer un tab (\t) comme séparateur fort (séparateur de segments) ?
tabAsSeparator=true

# Inclure le texte non balisé (pureTextForm) dans l'export en XML
inclPTFXML=false

# Ne pas inclure les balises encadrantes les segments textuels
excludeWrappingTags=true

# Squelette sans les segments, uniquement avec des identifiants de segments
bareSkeleton=false

# Si "vrai", alors le fichier-squelette contient des spans autofermants
# identifiant les frontières des segments.
# Exemple : <span class=SEG256-open" />corps du segment 256.<span class=SEG256-
close" />
# cela permet d'insérer les frontières comme des éléments HTML mais sans conflits
"inline/block"
spansInSkeleton=false

# Enlever les blancs encadrant une chaîne ?
trim=true

# Préfixe des identifiants de segments dans le squelette (par défaut SEG).
# Si "rien", alors le segment est préfixé uniquement de son numéro.
segmentPrefix=S

# Corriger les erreurs de segmentation causées par les références dans wikipedia
tryCorrectWikiRefs=true

################################################
########### Configs de la récursivité ###########
################################################

# Segmentation recursive (desactivée pour l'instant)
recursive=false

# Attributs de récursivité à segmenter
recursiveAtts=title|alt

# Attributs optionnels de récursivité à segmenter
```

```
optRecursiveAtts=

################################################
############## Configs de l'export ##############
################################################

# Nombre maximal de caractères que Google Translate peut traiter dans un fichier
texte (et, supposément, html)
# Si un fichier à exporter est trop long pour pouvoir être traité par GGL, on le
divise en plusieurs fichiers.

useGoogleLimit=false
googleUpperLimit=25000


# Pour des segments qui contiennent des balises dont les balises complémentaires
(fermante pour ouvrante et réciproquement) ne sont pas dans le segment
# ex 1 : " toto1 </span> toto2."
# ex 2 : " toto1 <span class="une_classe"> toto2."
# on peut les compléter automatiquement :
# ex 1 : "<span class="geta-auto-opened-tag">toto1 </span> toto2."
# ex 2 : " toto1 <span class="geta-auto-closed-tag_une_classe"> toto2.</span>"
# Cela est nécessaire pour pouvoir exporter chaque segment en html, mais il faut
par la suite enlever ces rajouts.

autocloseSegmentInternalTags=false

################################################
############## Configs diverses #################
################################################

# Utiliser la base de données ? (0|1 - faux par défaut)
useDB=0

# Type de SECTra (none|text|HTML - none par défaut)
type=HTML
```

Annexe 2 Exemples de sorties de segmentation et de normalisation

La première figure illustre le résultat de la segmentation d'un document pédagogique en phrases, avec une étape préalable de normalisation, qui consiste à détecter les formules mathématiques (mises en bleu dans l'exemple ci-dessous), puis à leur substituer des occurrences spéciales, avant segmentation.

〈S3: 6.2 Réduction polynomiale entre problèmes〉

〈S4: 6.2.1 Problème de décision〉

〈S5: Un problème de décision est un ensemble d'instances X et une question $Q(x)$? pour $x \in X$.〉 〈S6: Si la réponse est oui on dit que x est une instance positive (négative sinon).〉

〈S7: 6.2.2 Définition〉

〈S8: Soit P_1 et P_2 deux problèmes, une réduction polynomiale de P_1 vers P_2 ($P_1 \leq_p P_2$) est une fonction de l'ensemble des instances de P_1 telle que:〉

- 〈S9: elle est calculable en temps polynomiale.〉
- 〈S10: $P_1(x)$ est vrai si et seulement si $P_2(f(x))$ est vrai.〉

〈S11: 6.2.3 Théorème〉

- 〈S12: Si $P_1 \leq_p P_2$ et si $P_2 \in P$ alors $P_1 \in P$.〉
  〈S13: (Si $P_1 \leq_p P_2$ et $P_1 \notin P$ alors $P_2 \notin P$.)〉
- 〈S14: Si $P_1 \leq_p P_2$ et si $P_2 \leq_p P_3$ alors $P_1 \leq_p P_3$.〉

La deuxième figure montre un exemple de sortie associée, en XML, qui contient les segments sous leur forme HTML et texte « pur » (sans balises).

```
<?xml version="1.0" encoding="UTF-8"?>
<segments sourceURI="./6_hevea_detectedFormulas">
   <segment id="1" lang="fr">
      <originalForm format="html">Classification des problèmes</originalForm>
      <pureTextForm>Classification des problèmes</pureTextForm>
   </segment>
   <segment id="2" lang="fr">
      <originalForm format="html">La classe P des fonctions polynomiales ou des
         problèmes polynomiaux est l'ensemble des fonctions/problèmes qui peuvent
         être calculés/décidables par un algorithme de complexité polynomiale.
      </originalForm>
      <pureTextForm>La classe P des fonctions polynomiales ou des problèmes
         polynomiaux est l'ensemble des fonctions/problèmes qui peuvent être
         calculés/décidables par un algorithme de complexité polynomiale.
      </pureTextForm>
   </segment>
   <segment id="3" lang="fr">
      <originalForm format="html">6.2  Réduction polynomiale entre problèmes
      </originalForm>
      <pureTextForm>6.2 Réduction polynomiale entre problèmes</pureTextForm>
   </segment>
   <segment id="4" lang="fr">
      <originalForm format="html">6.2.1  Problème de décision</originalForm>
      <pureTextForm>6.2.1 Problème de décision</pureTextForm>
   </segment>
   <segment id="5" lang="fr">
      <originalForm format="html">Un problème de décision est un ensemble
         d'instances &lt;span class="mathFormula"&gt;&lt;span
         style="font-style:italic"&gt;X&lt;/span&gt;&lt;/span&gt; et une question
         &lt;span class="mathFormula"&gt;&lt;span
         style="font-style:italic"&gt;Q&lt;/span&gt;(&lt;span style="font-
         style:italic"&gt;x&lt;/span&gt;)&lt;/span&gt;
         ? pour &lt;span class="mathFormula"&gt;&lt;span style="font-
```

```
         style:italic"&gt;x&lt;/span&gt; ∈ &lt;span
         style="font-style:italic"&gt;X&lt;/span&gt;.&lt;/span&gt;</originalForm>
      <pureTextForm>Un problème de décision est un ensemble d'instances X et une
         question Q(x) ? pour x ∈ X.
      </pureTextForm>
   </segment>
   <segment id="6" lang="fr">
      <originalForm format="html">Si la réponse est oui on dit que &lt;span
         class="mathFormula"&gt;&lt;span
         style="font-style:italic"&gt;x&lt;/span&gt;&lt;/span&gt; est une
         instance positive (négative sinon).
      </originalForm>
      <pureTextForm>Si la réponse est oui on dit que x est une instance positive
         (négative sinon).
      </pureTextForm>
```

Annexe 3 Variété linguistique des en-têtes trouvées dans les messages

Voici un extrait du code de la classe chargée de traiter les en-têtes retrouvés dans le texte des courriels. Il en illustre le multilinguisme.

```
fieldsFrom = From | De | Expéditeur | От кого | 发件人 | Von | Sender |
Répondre à | Reply-To | Da | Van | De \(renvoi\) | 差出人

fieldsTo = Pour | Pour \(renvoi\) | Para | Destinataire | Destinataires |
收件人 | To | À | Ã€ | Кому | An | Aan | A | 宛先

fieldsCopy = Cc | Cci | Copie à | Copie | 抄送 | Copy to | Bcc | CC | CCI
| BCC | cc
fieldsDate = Enviado el | Enviadas | Date | Verzonden | Gesendet | Envoyé
| Message du | Date de renvoi | 发送时间 | Inviato | Отправленные | 送信日時

fieldsSubject = Asunto | Objet | Onderwerp | Sujet | 主題 | Oggetto |
Subject | Тема | Betreff | 件名

fieldsImportance = Importance | Importanza | Важность

fieldsOnBehalfOf = Im Auftrag von | On behalf of | de la part de

fieldsMessageSeparators = Original message | Исходное сообщение | Message
original | Message d'origine | Письмо | Messaggio originale | Message
transféré | Début du message réexpédié
```

Annexe 4 Exemple d'annotations BRAT

Les trois tableaux ci-dessous sont des fichiers produits par l'outil d'annotation BRAT, associés aux fichiers de conversations annotés. Ils contiennent les entités et les relations que nous avons trouvées dans trois conversations. Dans tous les exemples, les fragments de texte sont donnés *verbatim*, afin d'illustrer la présence de fautes et de disfluences.

L'exemple 1 ci-dessous contient plusieurs types de demande (envoi d'un document, information, action…).

```
T1	Send_document 2374 2429	tu m'avais parler de fichier latex tu peux les envoye ?
T2	Information_request 2064 2126	voila j ai fais la moitier des fichiers tu me dis si sa marche
T3	Action_request 1743 1796	Pourrais-tu voir ce qui pourrait causer le problème ?
T4	Requester 1981 1987	Ruslan
R1	Argument Arg1:T4 Arg2:T3
T5	Requestee 557 564	Mickaël
R2	Argument Arg1:T5 Arg2:T3
T6	Action_request 2593 2739	Pourrais-tu, très vite, dans tous les fichiers que tu as produit, remplacer les balises <code> par d'autres balises inline qui ont le même style ?
T7	Instruction 2759 2833	Tu peux, par exemple, les remplacer par <tt> ou revenir aux balises <pre>.
R3	Argument Arg1:T7 Arg2:T6
T8	Requester 3168 3174	Ruslan
T9	Requestee 2562 2569	Mickael
R4	Argument Arg1:T9 Arg2:T6
R5	Argument Arg1:T8 Arg2:T6
T10	Justification 2856 3017	Il semblerait que les balises <code>, ou peut-être de mauvaises imbrications de span et de code, bloquent Google dans sa traduction et il ne traduit pas au-delà.
R6	Argument Arg1:T10 Arg2:T6
```

Exemple 1 : courriel pris dans le corpus MACAU

L'exemple 2 montre l'expression de deux tâches, dont une est sous-jacente à l'autre (elles sont reliées ici par la relation SubTask).

```
T1	Action_request 315 373	Gilles, tu pourras mettre à jour les participants à l'UE ?
T2	Requestee 315 321	Gilles
R1	Argument Arg1:T2 Arg2:T1
T3	Action_request 382 475	Ruslan, il faut que tu fasses les procédures pour les vacations, si ce n'est pas encore fait.
T4	Requestee 382 388	Ruslan
R2	Argument Arg1:T4 Arg2:T3
T5	Action_request 476 650	Pour cela, tu dois joindre le service RH (rh.im2ag@ujf-grenoble.fr) en indiquant le code de l'UE et le nombre d'heures que tu vas faire, en précisant si c'est du TD ou du TP.
R3	SubTask Arg1:T5 Arg2:T3
T6	Requester 677 682	lydie
R4	Argument Arg1:T6 Arg2:T3
T7	Information_request 1017 1049	Comment s'appelle-t-il (+mail) ?
T8	Requester 1104 1109	lydie
R5	Argument Arg1:T8 Arg2:T7
```

Exemple 2 : courriel du corpus MACAU reliant une tâche et une sous-tâche

L'exemple 3 illustre d'autres entités intéressantes, telles qu'invitation, condition, suggestion.

```
T1	Suggestion 299 469	Je vais essayer de relayer certaines de tes problématiques, mais si tu veux qu'on aborde certains points précis tu es la bienvenue pour faire un mail que je transmettrai.
T2	Requester 596 601	Laure
R1	Argument Arg1:T2 Arg2:T1
T3	Action_request 1219 1292	Isabelle pourrais tu nous confirmer une salle (8/10 personnes avec vidéo)
T4	Action_request 1294 1403	Raphaël si tu ne peux pas décaler avec Ergi peux tu me transmettre tes remarques ou solutions d'amélioration.
T5	Condition 1302 1337	si tu ne peux pas décaler avec Ergi
R2	Argument Arg1:T5 Arg2:T4
T6	Requestee 1294 1301	Raphaël
R3	Argument Arg1:T6 Arg2:T4
T7	Requestee 1219 1227	Isabelle
R4	Argument Arg1:T7 Arg2:T3
T8	Requester 1443 1451	Isabelle
R5	Argument Arg1:T8 Arg2:T4
R6	Argument Arg1:T8 Arg2:T3
T9	Invitation 1149 1217	C'est donc ce *vendredi 19 à 10h* qui convient au plus grand nombre.
```

Exemple 3 : courriel du corpus ADE illustrant des actes de parole (invitation, suggestion

Annexe 5 Exemples de marqueurs de listes à puces

Voici un exemple de fichier paramètres de SEGNORM donnant les marqueurs de début d'énumération potentiels. Lorsque ces symboles apparaissent dans au moins deux lignes consécutives, potentiellement suivis par « . » ou «) », nous considérons que ces lignes forment une liste. Cela nous permet de mieux traiter les listes de tâches présentées sous cette forme.

Merci de :

1. *...*
2. *...*

```
listBullets =
        {"*", "°", "◖", "◗", "▪ ", "–", "—", "—", "--", "▶", "·", "◦",
        "•", "o", "•", "◦", "◘", "◙", "■", "□", "□", "▣", "◆", "◇",
        "◈", "□", "▪", "▫", "●", "○", "◎", "□", "▪", "▫", "◊", "○",
        "◌", "●", "◦", "•"};
orderedNumbers =
        {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11",
        "12", "13", "14", "15"};
orderedRomanNumerals =
        {"I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX", "X",
        "XI", "XII", "XIII", "XIV", "XV"};
orderedLetters =
        {"a",  "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l",
        "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y",
        "z"};
```

Annexe 6 Exemples d'expressions de contraintes temporelles

Le tableau ci-dessous contient des contraintes temporelles extraites du corpus, pour en illustrer la variété. Il y a des échéances, des marqueurs d'urgence, et une ou deux dates de début. Les contraintes trouvées sont en rouge et sont entourées par des fragments de texte pour donner un peu de contexte. Aucune normalisation particulière, hormis une anonymisation minimale, n'a été effectuée.

…de nous communiquer **aujourd'hui, demain matin au plus tard**, les tableurs…
…avant envoi à ACME **au plus tard demain midi**…
…occuper a partir du **2 Novembre.** D ici la merci aux…
…lui envoyer cela **aujourd'hui** (directement à elle…
…Votre retour **pour demain soir** serait idéal…
…Merci de le remplir **vite** afin de fixer une…
…l'original papier), de préférence **courant de la semaine prochaine**…
…signatures **pour le 20/03 13h**…
…é des choses (**avant le 24 octobre** si possible pour…
…r complétée **pour le 24 octobre.** Nous pourrons…
…ntendu c'était **pour avant-hier**/)…
…ions sont bienvenus **jusqu'à Jeudi soir.** Un peu de contexte…
…s commentaires **pour vendredi soir** (**lundi fin de matinée dernier délai**) …pour que je puisse…
…e le retourner **pour le lundi 5 mai** afin que nous puis…
…aire un retour **pour vendredi 13 matin au plus tard** car je suis déjà…
…contributions **pour le 4 mars**…
…echerche **d'ici à la fin février 2014**…
…merci de m'en faire **avant demain (mercredi) midi**…
…ats-marchés **pour le 30 novembre 2010**…
…**c'esturgent** que tu me répondes…
…ir nous l'adresser **au plus vite** …
…Objet **: TRES URGENT :** ANR ACME…
…**Quand la fiche sera imprimée,** on aura besoin de…
…que j'envoie tout **lundi dernier délai**…
…**la date limite arrive...** Peux-tu ajouter la…
…pourrais-tu (**dqp!**) vérifier ce livrable…
…de vendredi ? C'est **urgent**…
…merci de me le dire **tout de suite.**
…les documents **pour le 4 avril au plus tard,** car j'ai une semai…
…nds donc ta réponse **aujourd'hui 30/6**!
…votre document, et **avant vendredi midi.** Ensuite, je pars u…
…xposé correspondant **assez vite,** car on me réclame…
…invitons à établir **dès à présent** le contrat de coll…
…peux-tu regarder ça **dès que possible**?
…ributeurs spontanés **DÈS QUE POSSIBLE** (bien sûr, au dépa…
…ag/ACME). Il est **SUPER-URGENT** qu'il puisse faire…
…la page d'accueil **tout de suite** stp?
…Peux-tu me répondre **d'urgence** stp: ma correspond…
…envoyer le poster **d'ici lundi après-midi** (ou, au pire, **mardi matin**)?
…du ACME est **ce vendredi, 19 février.** Suite aux question
…envoyer tes slides **d'ici mardi soir *dernier délai*** stp?
…faudrait les infos... **ce soir** ou **demain,** et le chèque **au plus tard lundi à midi,** car je dois aller…
…pour traitement **rapide** de notre demande.
…de votre retour **pour ce près-midi** si possible.
…ci dessous **avant mercredi prochain**…
…documents demandés **dans les plus brefs délais.** (la validation de…
…peux-tu soumettre **dès que possible** stp ?
…easychair.org) **pour le 4 janvier 2017**…
…ACME **avant le 31 janvier 2017.**
…mél votre diaporama **mercredi - la veille du séminaire** ou **le jeudi matin au plus tard**?
…de me la communiquer **rapidement** ainsi que la date…
…à compléter **d'urgence** parce que les étudiants…
Le moment venu, nous vous demandons…
…cette rubrique **pendant les 10 jours à venir**…
…vouloir répondre **au plus tôt** à ce mail pour m'i…

Annexe 7 Lexiques de termes caractéristiques de tâches, et exemples de règles linguistiques

Voici les lexiques de termes souvent rencontrés dans les phrases et courriels exprimant des tâches.

- Termes désignant des documents :
  - *document, word, excel, fiche, fichier, txt, pdf, doc, .doc, .docx, rtf, ppt, powerpoint, .tex, xls, csv, diapo, diaporama, slide, transparent, page, formulaire, tableau, justificatif, livrable, pièce justificative, facture, devis, dossier, copie, contrat, rapport, convention, attestation, compte-rendu, scan, photocopie, article, annexe, manuscrit, exemplaire, original*
- Verbes relatifs aux documents :
  - *compléter, vérifier, corriger, relire, consulter, remplir, écrire, rédiger, imprimer, signer, déposer, faire-parvenir, faire-suivre, retourner, fournir, rapporter, envoyer, transmettre, joindre (ci-joint)*
- Verbes relatifs à l'information :
  - *préciser, informer, indiquer, dire, tenir-au-courant, décrire, communiquer, rappeler, faire-savoir, répondre, confirmer, valider*
- Expressions d'urgence :
  - *urgent, d'urgence, ça presse, ça urge, urgemment, immédiatement, sans (délai|attendre), aussi (vite|rapidement) que possible, très rapidement, (assez|très) (vite|rapidement), au plus tard, dès que possible, dernier délai, le plus (vite|rapidement) possible, dans les plus brefs délais, dans les délais les plus brefs, dépêche-toi, dépêchez-vous*
- D'autres traits caractéristiques extraits :
  - *merci-de-VERBE : nous vous remercions de nous envoyer...*
  - *merci-d-avance : je vous remercie par avance…*
  - *S'il-vous-plait : s'il vous plait, s'il te plait, stp, svp, prière de...*
  - *devoir : tu dois, vous devez, vous devriez, tu devais, vous deviez...*
  - *Lexique d'importance : important, crucial, essentiel, vital, indispensable, nécessaire*
  - *Si-tu-peux/si-vous-pouvez : si tu pouvais, si vous pouviez, ...*
  - *Ne-pas-oublier : n'oubliez (surtout) pas, ...*
  - *Besoin : j'ai besoin, nous avons besoin, nous aurions besoin...*
  - *IlFaut : il((me|te|nous|leur|lui|la))?((le|les|la|leur))? (faut)|(faudra(it)?)*

Voici un exemple de règle implémentée pour repérer des expressions de la forme « REMERCIEMENT de VERBE », par exemple, « merci de remplir… » ou « nous vous remercions donc de nous envoyer … ». Elle illustre la manière dont les autres règles linguistiques sont implémentées.

```
 public static void extractMerciDeVerbe(AnnotatedSegment s) throws IOException,
ClassNotFoundException {
        ArrayList<Token> tokensByPling = s.tf.tokensByPling;

        String phrase = s.getNormalizedString()
                    .replaceAll("['’]", " ")
                    .replaceAll("[0-9=>().,:;]", " ")
                    .replaceAll("\\s+", " ")
                    .trim();

        if (!Objects.equals(phrase, phrase.replaceAll("((en (serais?)|(seri?ons)) (tr[èe]s)
                    reconnaissants?)  |((serais?)|(seri?ons)) (tr[èe]s) reconnaissants? de",
                    ""))){
            s.tf.merciDe_Verbe.value = true;
            return;
        }

        if (phrase.endsWith("merci de") && s.tf.followedByList.value) {
            s.tf.merciDe_Verbe.value = true;
            return;
        }

        for (int j = 0; j < tokensByPling.size(); j++) {
```

```
            Token pt = tokensByPling.get(j);
            if (Objects.equals(pt.getForm().toLowerCase(), "de") ||
                    (Objects.equals(pt.getForm().toLowerCase(), "d'"))) {
                if ((j > 0 && (Objects.equals(tokensByPling.get(j -1)
                                    .getForm()
                                    .toLowerCase(), "merci") ||
                            Objects.equals(tokensByPling.get(j - 1).getForm().toLowerCase(),
                                    "remercions") ||
                            Objects.equals(tokensByPling.get(j - 1).getForm().toLowerCase(),
                                    "remercie"))) ||
                            (j > 1 && Objects.equals(tokensByPling.get(j - 1)
                                    .getForm()
                                    .toLowerCase(), "donc") &&
                                (Objects.equals(tokensByPling.get(j-2)
                                    .getForm()
                                    .toLowerCase(), "merci") ||
                                      Objects.equals(tokensByPling.get(j - 2)
                                    .getForm()
                                    .toLowerCase(), "remercions") ||
                                      Objects.equals(tokensByPling.get(j - 2)
                                    .getForm()
                                    .toLowerCase(), "remercie")))) {
                    for (int k = j; k < tokensByPling.size() && k < j + 4; k++) {
                        if (Objects.equals(tokensByPling.get(k).getPos(), "VINF") ||
                                    Objects.equals(tokensByPling.get(k).getPos(), "V")) {
                            s.tf.merciDe_Verbe.value = true;
                            return;
                        }
                    }
                }
            }
        }
    }
```

Le tableau ci-dessous montre des règles ajoutées à P-LING pour extraire certaines expressions temporelles comme des dates limites. Cela revient à ajouter l'attribut *Max* à l'entité nommée *Time*.

```
// [[lundi dernier délai]].
else if (tokensMatchLookahead(sentence, idToken+1, dernierDelai) &&
sentence.getToken(idToken).getType().contains("Time")){
    Candidate candidate=new Candidate();
    candidate.setRule("Time>expression");
    candidate.setType(sentence.getToken(idToken).getType()+">Max");
    sentence.getToken(idToken).addCandidate(candidate);

    Phrase phrase=new Phrase();
    phrase.setId(idToken);
    token.addPhrase(phrase);
    sentence.getToken(idToken).addPhrase(phrase);
}

//Lundi prochain, la semaine prochaine
else if (idToken+1 < sentence.getTokens().size() &&
sentence.getToken(idToken+1).getLemma().matches("prochain(e?s?)") &&
sentence.getToken(idToken).getType().contains("Time")){
    Candidate candidate=new Candidate();
    candidate.setRule("Time>expression");
    candidate.setType(sentence.getToken(idToken).getType());
    sentence.getToken(idToken+1).addCandidate(candidate);

    Phrase phrase=new Phrase();
    phrase.setId(idToken);
    token.addPhrase(phrase);
    sentence.getToken(idToken).addPhrase(phrase);
}


// [[pour demain soir]]
// [[pour vendredi 13 matin au plus tard]]
// [[pour vendredi soir]]
```

```
else if(token.getLemma().equals("pour") &&
(sentence.getToken(idToken+1).getType().contains("Time"))){
    Candidate candidate=new Candidate();
    candidate.setRule("Time>expression");
    candidate.setType(sentence.getToken(idToken+1).getType()+">Max");
    sentence.getToken(idToken+1).addCandidate(candidate);

    Phrase phrase=new Phrase();
    phrase.setId(idToken);
    token.addPhrase(phrase);
    sentence.getToken(idToken+2).addPhrase(phrase);
}

// [[pour le 24 octobre]].
// [[pour le 30 novembre 2010]],
// [[pour le 4 janvier 2017]].
// [[pour le lundi 5 mai]]
// [[pour le 4 avril au plus tard]]
else if(token.getLemma().equals("pour") &&
(sentence.getToken(idToken+1).getForm().toLowerCase().matches("le|la") &&
sentence.getToken(idToken+2).getType().contains("Time"))){
    Candidate candidate=new Candidate();
    candidate.setRule("Time>expression");
    candidate.setType(sentence.getToken(idToken+2).getType()+">Max");
    sentence.getToken(idToken+2).addCandidate(candidate);

    Phrase phrase=new Phrase();
    phrase.setId(idToken);
    token.addPhrase(phrase);
    sentence.getToken(idToken+2).addPhrase(phrase);
}

// [[d'ici à la fin février 2014]].
// [[d'ici mardi soir *dernier délai*]] stp?
// [[d'ici lundi après-midi]]
else if((tokensMatchLookbehind(sentence, idToken, dIci) ||
(tokensMatchLookbehind(sentence, idToken, dIciLe) ||
tokensMatchLookbehind(sentence, idToken, dIciALa)) &&
(sentence.getToken(idToken+1).getType().contains("Time")))){
    Candidate candidate=new Candidate();
    candidate.setRule("Time>expression");
    candidate.setType(sentence.getToken(idToken+1).getType()+">Max");
    sentence.getToken(idToken+1).addCandidate(candidate);

    Phrase phrase=new Phrase();
    phrase.setId(idToken);
    token.addPhrase(phrase);
    sentence.getToken(idToken+1).addPhrase(phrase);
}

// [[avant le 24 octobre]]
// [[avant le 31 janvier 2017]]
else if(token.getLemma().equals("avant") &&
(sentence.getToken(idToken+1).getForm().toLowerCase().matches("le|la") &&
sentence.getToken(idToken+2).getType().contains("Time"))){
    Candidate candidate=new Candidate();
    candidate.setRule("Time>expression");
    candidate.setType(sentence.getToken(idToken+2).getType()+">Max");
    sentence.getToken(idToken+2).addCandidate(candidate);

    Phrase phrase=new Phrase();
    phrase.setId(idToken);
    token.addPhrase(phrase);
    sentence.getToken(idToken+2).addPhrase(phrase);
}
```

Annexe 8 Ontologies de MACAU

Voici la taxonomie relative à la complexité calculatoire. Celle de la forme est au IV.1.2.2 (p. 98).

- owl:Thing
  - Notions
    - Calculabilite
      - Fonction_Turing_calculable
    - Codage
    - Complexite_calculatoire
      - Classe_de_complexite
        - Classes_de_complexite
          - Co-NP
          - EXPTIME
          - NP
            - NP_Complet
              - Problemes_NP_complets
                - Problemes_de_graphes
                  - Circuit_Hamiltonien
                  - Couverture_de_graphe
                  - Voyageur_du_commerce
                - SAT
                  - 3_SAT
            - P
              - LogLineaire
              - Puissance_k
                - Cubique
                - Lineaire
                - Quadratique
              - Sublineaire
                - Constant
                - DLOGTIME
          - NP_Difficile
      - Complexite_d_un_programme_pseudo_Pascal
      - Complexite_d_un_programme_RAM
      - Equivalence_polynomiale
        - Equivalence_MT_a_k_bandes_a_MT_monobande
        - Equivalence_MT_a_quintuplets_et_MT_a_quadruplets
        - Equivalence_MT_a_RAM
        - Equivalence_pseudo_Pascal_a_RAM
        - Transitivite_de_la_reduction_polynomiale
      - Fonctions
        - Ordre_de_grandeur
          - Fonction_de_reference
            - 2_puissance_n
            - constante
            - factorielle_n
            - log_n
            - n
            - n_log_n
            - n_puissance_k
            - n_puissance_n
          - Notations_de_Knuth
            - GrandO
            - Omega
            - PetitO
            - Theta
      - Reduction_polynomiale
    - Langage_formel
    - Modele_de_calcul
      - Automate_a_pile
      - Automate_d_etats_finis
      - Lambda_calcul
      - Machine_a_registres
        - Machine_RAM
          - Bande_RAM
          - Programme_RAM

- Complexite_d_un_programme_RAM
- Instruction
- Registre
- Simulation_par_RAM
- Machine_RASP
- Machine_de_Turing
- Calcul_d_une_MT
- Complexite_d_un_calcul_de_MT
- Fonction_Calculee
- Pas_de_calcul
- Resultat_d_un_calcul
- Configuration_d_une_MT
- Configuration_Initiale
- Configuration_Terminale
- Determinisme
- Elements_d_une_MT
- Bande
- Alphabet_de_bande
- Expression_de_bande
- Etat
- Etat_Final
- Etat_Initial
- N_uplets
- Quadruplets
- Quintuplets
- Tete_de_lecture
- Transition
- Non_determinisme
- Simulation_par_une_MT
- Types_de_MT
- Machine_a_oracle
- Machine_monobande
- Machine_a_demi_bande
- Machine_lineairement_bornee
- Machine_multibande
- Modele_pseudo_Pascal
- Programme_pseudo_Pascal
- Complexite_d_un_programme_pseudo_Pascal
- Instruction_pseudo_Pascal
- Simulation
- Simulation_par_RAM
- Simulation_par_une_MT
- Turing_completude
- Turing_equivalence
- Problemes_de_decision
- Problemes_decidables
- Problemes_NP_complets
- Problemes_de_graphes
- Circuit_Hamiltonien
- Couverture_de_graphe
- Voyageur_du_commerce
- SAT
- 3_SAT
- Problemes_P
- Horn_SAT
- Problemes_indecidables
- Probleme_de_l_arret
- Probleme_de_Post
- Probleme_du_castor_affaire
- Theoremes
- Complexite
- Equivalence_polynomiale
- Equivalence_MT_a_k_bandes_a_MT_monobande
- Equivalence_MT_a_quintuplets_et_MT_a_quadruplets
- Equivalence_MT_a_RAM
- Equivalence_pseudo_Pascal_a_RAM
- Transitivite_de_la_reduction_polynomiale